Oracle® Business Intelligence Applications Fusion Edition

Installation and Configuration Guide

Release 7.9.5

E12083-01

May 2008

ORACLE®

Oracle Business Intelligence Applications Fusion Edition Installation and Configuration Guide, Release 7.9.5

E12083-01

Copyright © 2008, 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

Preface xxv

Audience xxv
Documentation Accessibility xxv
Related Documents xxvi
Conventions xxvi

1 What's New in This Release

1.1 What's New In Oracle Business Intelligence Applications Version 7.9.5? 1-1

2 Overview of Oracle Business Intelligence Applications

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 Business Intelligence Applications 2-5
2.5 Using Repository Documentation 2-6

3 Pre-installation and Predeployment Requirements for Oracle Business Intelligence Applications

3.1 Operating System, Driver, and Connectivity Requirements for Oracle Business Analytics Warehouse 3-2
3.2 General Guidelines for Setting Up Oracle Business Analytics Warehouse 3-2
3.3 IBM DB2 UDB-Specific Database Guidelines for Oracle Business Analytics Warehouse 3-4
3.4 IBM DB2 UDB zOS and OS/390 and Z/os-Specific Database Guidelines for Oracle Business Analytics Warehouse 3-6
3.5 Additional Suggestions for Optimizing Oracle Performance in Oracle Business Analytics Warehouse 3-6
3.6 SQL Server-Specific Database Guidelines for Oracle Business Analytics Warehouse 3-7
3.6.1 Setting the ANSI NULL Option 3-7
3.6.2 Modifying the DB Library Options Setting 3-8
3.6.3 Recommended SQL Server Database Parameters 3-8
3.7 Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse 3-9
3.7.1 General Guidelines for Teradata Deployments 3-10
3.7.2 Best Practices For Teradata Deployments 3-10

3.7.2.1 LEFT OUTER JOIN Issue ... 3-11
3.7.2.2 Database Statistics ... 3-12
3.7.2.3 Additional Statistics Required ... 3-12
3.7.2.3.1 Collect Statistics As A Pre-requisite ... 3-12
3.7.2.3.2 Group By Vs Distinct ... 3-12
3.7.2.3.3 Pruning Of Mapping ... 3-12
3.7.2.3.4 Loader Configurations ... 3-12
3.7.2.3.5 Fastload ... 3-13
3.7.2.3.6 Mload ... 3-13
3.7.2.3.7 Tpump ... 3-13
3.8 Miscellaneous Information About Oracle Business Intelligence Applications Deployments ... 3-14
3.8.1 Preconfigured Mapping for the SA System Subject Area ... 3-14
3.8.2 Using Initialization Blocks ... 3-15
3.8.3 Performance Improvements for Change Capture for Incremental ETL Run ... 3-16
3.8.3.1 Implementing Transactional Database (OLTP) Indexes ... 3-16
3.8.3.2 An Example Of Change Capture SQL And Required Indexes ... 3-16
3.8.4 Running an ETL with a Non-English OLTP Data Source ... 3-17
3.9 Partitioning Guidelines For Large Fact Tables ... 3-18
3.9.1 Introduction to Partitioning Large Fact Tables ... 3-18
3.9.2 How to Partition Large Fact Tables ... 3-18
3.9.3 Configure DAC To Support ETL For Partitioned Tables ... 3-21
3.9.4 Example XML Files ... 3-24
3.9.4.1 Pre_W_REVN_F_ini.xml - Pre SQL xml file name for Full Load ... 3-25
3.9.4.2 Post_W_REVN_F_ini.xml - Post SQL xml file name for Full Load ... 3-25
3.9.4.3 Pre_W_REVN_F_incr.xml- Pre SQL xml file name for Incremental Load ... 3-26
3.9.4.4 Post_W_REVN_F_incr.xml- Post SQL xml file name for Incremental Load ... 3-26

4 Installing and Setting Up Oracle Business Intelligence Applications

4.1 About Oracle Business Intelligence Applications Topologies ... 4-1
4.2 Installation and Configuration Process Task List ... 4-3
4.3 Mandatory Requirements and Pre-installation Tasks ... 4-5
4.3.1 Oracle Business Intelligence Infrastructure Requirements ... 4-5
4.3.2 Informatica PowerCenter Requirements ... 4-6
4.3.3 Code Page and Data Movement Requirements ... 4-7
4.3.4 Pre-installation Tasks ... 4-8
4.3.4.1 Create Databases ... 4-9
4.3.4.1.1 How to Create the SSE Role ... 4-9
4.3.4.2 Install Database Connectivity Software ... 4-10
4.3.4.2.1 Setting the NLS_LANG Environment Variable for Oracle Databases ... 4-10
4.3.4.2.2 Setting the DB2CODEPAGE Environment Variable for DB2 Databases ... 4-11
4.3.4.3 Create ODBC Database Connections ... 4-12
4.3.4.4 Set Environment Variables ... 4-13
4.3.4.5 How to Install the Java JDK ... 4-14
4.3.4.6 Download Hibernate Libraries ... 4-14
4.4 Installing Oracle Business Intelligence Applications Software ... 4-15
4.4.1 How to Run the Oracle Business Intelligence Applications Installer (Windows) ... 4-15

4.5 Installing Informatica PowerCenter Services Software ... 4-18
4.5.1 How to Install Informatica PowerCenter Client Tools ... 4-19
4.5.2 How to Install Informatica PowerCenter Services ... 4-20
4.6 Restoring the Oracle Business Intelligence Pre-Built Repository ... 4-23
4.7 Configuring Informatica PowerCenter Components ... 4-24
4.7.1 How to Copy Source Files and Lookup Files ... 4-24
4.7.2 Setting PowerCenter Integration Services Relaxed Code Page Validation ... 4-25
4.7.3 Setting PowerCenter Integration Services Custom Properties ... 4-25
4.8 Installing and Configuring the DAC Client ... 4-27
4.8.1 How to Install and Set Up the DAC Client ... 4-27
4.8.2 Enabling DAC Client Communication with Informatica PowerCenter ... 4-29
4.8.2.1 How to Install Informatica pmcmd and pmrep Command Line Programs ... 4-29
4.8.2.2 How to Set Environment Variables for a DAC Client ... 4-30
4.8.2.3 How to Verify that the DAC Client is able to use pmrep and pmcmd ... 4-31
4.8.3 How to Create a DAC Connection ... 4-31
4.8.4 Creating the DAC Repository Schema ... 4-32
4.9 Importing the DAC Metadata ... 4-33
4.10 Creating the Oracle Business Analytics Warehouse Database Tables ... 4-34
4.10.1 How to Create Data Warehouse Tables ... 4-35
4.10.2 How to Create Data Warehouse Tables on a Teradata Database ... 4-38
4.11 How to Install and Configure the DAC Server ... 4-39
4.11.1 Overview to Installing and Configuring the DAC Server ... 4-39
4.11.2 Installing the DAC Server ... 4-40
4.11.2.1 How to install a DAC Server on Windows ... 4-40
4.11.2.2 How to install a DAC Server on UNIX or Linux ... 4-41
4.11.3 Configuring the Connection Between the DAC Server and DAC Repository ... 4-42
4.11.3.1 How to Configure the DAC Server Repository Connection Using the DAC Client (Windows) ... 4-43
4.11.3.2 How to Configure the DAC Server Repository Connection Using serverSetupPrompt Scripts (Windows, UNIX or Linux) ... 4-44
4.11.4 Enabling DAC Server Communication with Informatica PowerCenter ... 4-45
4.11.4.1 How to Set Environment Variables on Windows ... 4-45
4.11.4.2 How to Set Environment Variables on UNIX or Linux ... 4-46
4.11.4.3 How to Verify Java JDK Availability and Version ... 4-48
4.11.5 Starting the DAC Server ... 4-48
4.12 Configuring the DAC Settings ... 4-48
4.12.1 How to Set DAC System Properties ... 4-48
4.12.2 How to Register the Informatica Services in the DAC ... 4-49
4.12.3 How to Set Physical Data Sources ... 4-51
4.13 Configuring Relational Connections In Informatica Workflow Manager ... 4-53
4.13.1 How to Log Into Informatica PowerCenter Workflow Manager ... 4-54
4.13.2 Configuring Relational Connections in Informatica PowerCenter Workflow Manager ... 4-54
4.13.2.1 How to configure the relational connections ... 4-54
4.13.2.2 (Teradata specific) How to configure the Informatica Repository for Teradata external loader connections ... 4-56
4.14 Configuring the SiebelUnicodeDB Custom Property ... 4-56
4.15 Miscellaneous and Additional Configuration Tasks ... 4-57

4.15.1 How to Set the Code Page File Property for Unicode Environments on Teradata Databases ... 4-58
4.15.2 How to Set Up the HOSTS File for Teradata Installations on Windows ... 4-58
4.15.3 How to Update Siebel Transactional Database Schema Definitions ... 4-59
4.15.3.1 How to Run the DDLIMP Tool From The Command Line ... 4-59
4.15.3.2 How to Apply Siebel (CRM) Schema Changes For Oracle, DB2/UDB, MSSQL ... 4-59
4.15.3.3 How to Apply CRM Schema Changes For DB2 on 390 or z/os ... 4-60
4.15.3.4 How to Verify the Siebel (CRM) Schema Changes ... 4-61
4.15.3.5 About Delete Triggers ... 4-61
4.15.4 How to Create Stored Procedures for DB2-UDB ... 4-62
4.15.4.1 How to Verify the existence of stored procedures ... 4-62
4.15.4.2 How to Create DB2 stored procedures ... 4-62
4.15.5 Teradata-Specific Installation and Configuration Checklist ... 4-63
4.15.6 How to Configure Email Recipients in the DAC Client ... 4-63
4.15.7 How to Configure Email in the DAC Server ... 4-64
4.15.8 How to Back Up and Restore the Informatica Repository (Optional) ... 4-65
4.15.9 How to Set DAC Source System Parameters ... 4-65
4.15.10 Migrating Seed Data from a Siebel Source System to the Oracle BI Repository ... 4-66
4.16 About Running A Full Load ETL ... 4-67
4.16.1 An Example of Running a Full Load ETL ... 4-67

5 Customizing the Oracle Business Analytics Warehouse

5.1 Oracle Business Analytics Warehouse Customization Scenarios ... 5-1
5.1.1 Types of Customizations ... 5-2
5.1.2 Upgrade Consideration ... 5-2
5.2 Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables ... 5-2
5.2.1 About Extending Mappings ... 5-3
5.2.2 Impact of Customization on Upgrade ... 5-4
5.2.2.1 Points to Remember ... 5-5
5.2.3 Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse ... 5-6
5.2.4 Scenario for Extending the Oracle Business Analytics Warehouse ... 5-6
5.2.4.1 Tips for Modifying the SQL Override in a Source Qualifier ... 5-7
5.2.4.2 Example of Extracting Data from a Siebel Source ... 5-7
5.2.4.3 Including a Source Table for the Change Capture Process ... 5-9
5.2.4.4 Example of Extracting Data from a Non-Siebel Packaged Source ... 5-9
5.2.4.5 Example of Extracting Data from a Universal Source ... 5-11
5.2.4.6 Example of Loading Data into an Existing Target Table ... 5-12
5.2.4.7 Updating the DAC ... 5-13
5.3 Other Types of Customizations Requiring Special Handling ... 5-13
5.3.1 Modifying Category 2 SCD Triggers ... 5-13
5.3.2 Adding Dimensions to Existing Facts ... 5-14
5.3.3 Adding Date Dimensions to Existing Facts ... 5-16
5.3.4 Adding Currencies to an Existing Table ... 5-17
5.4 Category 2 Customizations: Adding Additional Tables ... 5-17
5.4.1 About Creating New Dimension or Fact Tables ... 5-17
5.4.1.1 Required Columns ... 5-18

5.4.1.2 About the Oracle Business Analytics Warehouse DATASRC_NUM_ID Column ... 5-18
5.4.2 Using Custom Folders in the Oracle Business Analytics Warehouse ... 5-19
5.4.3 Creating Custom Informatica Workflows ... 5-19
5.4.4 Important Notes About Customizing the Oracle Business Analytics Warehouse .. 5-20
5.4.4.1 Additional Customization Instructions ... 5-20
5.4.5 Including a Source Table for Change Capture Process ... 5-22
5.4.6 Adding a New Dimension in the Oracle Business Analytics Warehouse ... 5-23
5.4.7 Adding a New Fact Table in the Oracle Business Analytics Warehouse ... 5-23
5.4.8 Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse ... 5-24
5.5 Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table ... 5-24
5.6 Configuring Extracts ... 5-25
5.6.1 Extracting Additional Data ... 5-25
5.6.1.1 Extracting New Data Using an Existing Source Table ... 5-26
5.6.1.2 Extracting Data from a New Source Table ... 5-26
5.6.2 Setting Up the Delimiter for Source Files ... 5-27
5.6.3 Configuring a Source Adapter Mapplet ... 5-27
5.7 Configuring Loads ... 5-28
5.7.1 Filtering and Deleting Records ... 5-28
5.7.2 About Primary Extract and Delete Mappings Process ... 5-29
5.7.3 About Working with Primary Extract and Delete Mappings ... 5-30
5.7.3.1 Deleting the Configuration for Source-Archived Records ... 5-30
5.7.3.2 Enabling Delete and Primary Extract Sessions ... 5-31
5.8 Configuring Slowly Changing Dimensions ... 5-31
5.8.1 About Identifying Historically Significant Attributes ... 5-32
5.8.1.1 About the Extract View ... 5-32
5.8.2 Type I and Type II Slowly Changing Dimensions ... 5-33
5.8.2.1 Type I Slowly Changing Dimension ... 5-33
5.8.2.2 Type II Slowly Changing Dimension ... 5-33
5.8.2.3 Effective Dates ... 5-34
5.9 About Stored Lookups ... 5-35
5.10 Codes Lookup ... 5-35
5.10.1 W_CODES Table ... 5-36
5.10.2 Codes Mappings ... 5-36
5.10.3 Codes Mapplets ... 5-37
5.10.4 Configuring Extension Column Code Description Lookups ... 5-37
5.11 About Resolving Dimension Keys ... 5-37
5.11.1 Resolving the Dimension Key Using Lookup ... 5-38
5.12 About Domain Values ... 5-39
5.12.1 About the Domain Value Conversion Process ... 5-39
5.12.1.1 Preparing to Define the Rules ... 5-39
5.12.2 About the Importance of Domain Values ... 5-40
5.12.2.1 Hire Count ... 5-41
5.12.2.2 Re-hires Ratio ... 5-41
5.12.2.3 New Hire Count ... 5-41

5.12.2.4 Newly Separated Veterans - New Hires ... 5-41
5.12.2.5 Other Protected Veterans - New Hires ... 5-41
5.12.2.6 Special Disabled Veteran Head count - New Hires ... 5-41
5.12.2.7 Vietnam Era Veteran Head count - New Hires ... 5-42
5.12.3 About Extending the Domain Value Set ... 5-42
5.13 Configuring the Domain Value Set with CSV Worksheet Files ... 5-42
5.14 Configuring the Domain Value Set Using Informatica PowerCenter Designer ... 5-43
5.15 Configuring Conformed Dimensions ... 5-44
5.15.1 Configuring Conformed Dimensions for Universal Adapter ... 5-44
5.15.1.1 Product Effective Dates in the Products Dimension ... 5-44

6 Integrated Security for Oracle Business Intelligence Applications

6.1 Types of Security in Oracle Business Intelligence Applications ... 6-1
6.1.1 Checking Oracle Business Intelligence Applications User Responsibilities ... 6-2
6.1.2 Registering a New User Responsibility in Oracle Business Intelligence ... 6-3
6.1.3 Default Security Settings in Oracle Business Intelligence ... 6-3
6.2 Integrating Data Security with Oracle E-Business Suite ... 6-3
6.2.1 Authorization ... 6-4
6.2.2 Operating Unit-Based Security With Oracle EBS ... 6-4
6.2.2.1 About Operating Unit-Based Security With Oracle EBS ... 6-4
6.2.2.2 Implementation Steps For Operating Unit-Based Security With Oracle EBS ... 6-5
6.2.3 Inventory Organization-Based Security With Oracle EBS ... 6-7
6.2.3.1 About Inventory Organization-Based Security With Oracle EBS ... 6-7
6.2.3.2 Implementation Steps For Inventory Organization-Based Security With Oracle EBS ... 6-8
6.2.4 Company Org-based Security Against Oracle EBS ... 6-9
6.2.4.1 About Company Org-based Security Against Oracle EBS ... 6-9
6.2.4.2 Implementation Steps For Company Org-based Security Against Oracle EBS 11i ... 6-9
6.2.5 Ledger Security Against Oracle EBS ... 6-10
6.2.6 Business Group Org-based Security Against Oracle EBS ... 6-13
6.2.6.1 About Business Group Org-based Security Against Oracle EBS ... 6-13
6.2.6.2 Implementation Steps For Business Group Org-based Security Against Oracle EBS ... 6-13
6.2.7 HR Org-based Security Against Oracle EBS ... 6-14
6.2.8 Human Resource Personnel Data Analyst Security Against Oracle EBS ... 6-16
6.2.9 Employee Based Security with Oracle EBS ... 6-18
6.3 Integrating Data Security with Oracle's PeopleSoft Enterprise Applications ... 6-19
6.3.1 Authorization ... 6-19
6.3.2 About Operating Unit-based Security Against PeopleSoft Financials ... 6-20
6.3.3 About Company Org-based Security Against PeopleSoft Financials or PeopleSoft HR ... 6-21
6.3.4 About Ledger-based Security against Oracle's PeopleSoft Enterprise Applications ... 6-22
6.3.5 About HR Org-based Security Against PeopleSoft HR ... 6-23
6.3.6 About Payables Org-based security Against PeopleSoft Financials ... 6-24
6.3.7 About Receivables Org-based Security Against PeopleSoft Financials ... 6-25
6.3.8 About SET ID-based Security Against PeopleSoft Financials ... 6-26

6.3.9 About Human Resource Personnel Data Analyst Security Against PeopleSoft HR ... 6-27
6.3.10 Employee Based Security Against PeopleSoft Application ... 6-29
6.4 Metadata Object-Level Security in Oracle Business Intelligence Applications ... 6-29
6.4.1 Metadata Object-Level Security (Repository Groups) ... 6-29
6.4.1.1 Where is Repository Groups Security Configured? ... 6-30
6.4.1.2 Metadata Object-Level (Presentation Services) Security in Oracle Business Intelligence Applications ... 6-30
6.5 Integrating Data Security with the Siebel CRM Application ... 6-31
6.5.1 Primary Position based Security ... 6-31
6.5.1.1 Introduction ... 6-31
6.5.1.2 Primary Position based Security Group ... 6-32
6.5.1.3 Configuring Oracle BI Repository Table Joins for Position/Employee Based Security ... 6-33
6.5.2 Primary Position Based Security for CRM Vertical applications ... 6-34
6.5.2.1 Oracle Partner Analytics Security Settings ... 6-35
6.5.2.1.1 Partner Manager Role-Based Interactive Dashboards Mapping ... 6-35
6.5.2.1.2 Oracle's Siebel Partner Portal Role-Based Interactive Dashboards Mapping ... 6-37
6.5.2.1.3 PRM Analytics Subject Area Mapping ... 6-38
6.5.2.1.4 PRM Analytics Subject Area Visibility ... 6-39
6.5.2.1.5 PRM Analytics Data-Level Visibility ... 6-39
6.5.2.2 Consumer Sector Analytics Security Settings ... 6-40
6.5.2.3 Communications, Media, and Energy (CME) Analytics Security Settings ... 6-40
6.5.2.4 Financial Services Analytics Security Settings ... 6-41
6.5.2.4.1 Parent and Child Group Behavior ... 6-42
6.5.2.5 Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics Security Settings ... 6-43
6.5.2.6 Usage Accelerator Analytics Security Settings ... 6-45
6.5.3 Team Based Visibility ... 6-47
6.5.3.1 Summary ... 6-47
6.5.3.2 Business Requirements Overview ... 6-47
6.5.3.3 Design Details ... 6-48
6.5.3.4 Data Model ... 6-48
6.5.3.5 Fact Security ... 6-50
6.5.3.6 ETL Design ... 6-50
6.5.3.6.1 Available Mappings ... 6-50
6.5.3.6.2 Creating Delete Triggers ... 6-52
6.5.3.6.3 DAC Metadata ... 6-52
6.5.3.7 DAC Metadata ... 6-52
6.5.3.8 Oracle Business Intelligence Applications Repository Metadata ... 6-53
6.5.4 Embedded Oracle Business Intelligence support for Changing Positions ... 6-54
6.5.5 Business Unit based Security ... 6-55
6.5.6 Primary Owner Security ... 6-56
6.6 Data-Level Security In Oracle Business Intelligence Applications ... 6-56
6.6.1 Data-Level Security in Oracle Business Intelligence Applications ... 6-56
6.6.1.1 Roadmap of Data Security in Oracle Business Intelligence Applications releases ... 6-57

6.6.2 Implementing Data-Level Security in the Oracle Business Intelligence Applications Repository 6-58
6.6.3 Initialization Blocks Used For Security Oracle Business Intelligence Applications 6-59
6.6.3.1 Data Security groups in Oracle Business Intelligence Applications 6-60

7 Configuring Common Areas and Dimensions

7.1 Source Independent Configuration Steps 7-1
7.1.1 Configuration Required Before A Full Load 7-1
7.1.1.1 How to Configure Initial Extract Date 7-2
7.1.1.2 How to Configure Global Currencies 7-2
7.1.1.3 How to Configure Exchange Rate Types 7-3
7.1.1.4 How to Configure Fiscal Calendars 7-4
7.1.1.4.1 Overview To Setting Up The Time Dimension 7-4
7.1.1.4.2 Setting Up Fiscal Calendar 7-5
7.1.1.4.3 Setting Up Fiscal Calendar By Fiscal Week 7-5
7.1.1.4.4 Setting Up Fiscal Calendar By Fiscal Month 7-5
7.1.1.4.5 Process of Setting Up The Fiscal Calendar 7-5
7.1.1.4.6 Reloading the Time Dimension Tables After Your Data Warehouse is Populated 7-6
7.1.1.4.7 Notes 7-6
7.1.1.5 How to Configure the General Ledger Account Hierarchies 7-7
7.1.1.6 Configuring General Ledger Account Hierarchies Using General Ledger Accounting Flexfield value sets definitions 7-8
7.1.1.7 How to Configure General Ledger Account Hierarchies Using Financial Statement Generator (FSG) Report Definition 7-13
7.1.2 Configuration Steps for Controlling Your Data Set for All Source Systems 7-17
7.1.2.1 How to Configure Data Source Num IDs 7-17
7.2 Oracle EBS-Specific Configuration Steps 7-18
7.2.1 Configuration Required Before A Full Load for Oracle EBS 7-19
7.2.1.1 Configuration of Product hierarchy (except for GL, HR modules) 7-19
7.2.1.1.1 How to identify Category Sets from EBS 7-19
7.2.1.1.2 Configure the DAC Source System Parameters for Product Hierarchy 7-19
7.2.1.1.3 Configure the Hierarchy Mapping With the Segments 7-20
7.2.1.2 Configure the Master Inventory Organization in Product dimension Extract for Oracle 11i adapter (Except for GL & HR Modules) 7-21
7.2.1.3 How to Map Oracle General Ledger Natural Accounts to Group Account Numbers 7-21
7.2.1.4 How to do Group Account correction for Oracle Application 7-24
7.2.2 Configuration Steps for Controlling Your Data Set for Oracle EBS 7-25
7.2.2.1 Configuring the Country Region and State Region Name 7-25
7.2.2.2 Configuring the State Name 7-26
7.2.2.3 Configuring the Country Name 7-27
7.2.2.4 Configuring the Configuring the Make-Buy Indicator 7-28
7.3 PeopleSoft-Specific Configuration Steps 7-28
7.3.1 Configuration Required Before A Full Load for PeopleSoft 7-28
7.3.2 Configuration Steps for Controlling Your Data Set for PeopleSoft 7-28
7.3.2.1 Internal Organization Hierarchy Extraction Customization 7-28
7.4 Siebel-Specific Configuration Steps 7-30

7.4.1 Configuration Required Before A Full Load for Siebel Source Systems 7-30
7.4.2 Configuration Steps for Controlling Your Data Set for Siebel Source Systems..... 7-31

8 Configuring Oracle Procurement and Spend Analytics

8.1 Overview of Oracle Procurement and Spend Analytics 8-1
8.1.1 Oracle Procurement and Spend Analytics Module 8-1
8.1.2 Supplier Performance Analytics Module 8-2
8.2 Configuration Required Before A Full Load for Oracle Procurement and Spend Analytics 8-3
8.2.1 Configuration Steps for Oracle Procurement and Spend Analytics For All Source Systems 8-3
8.2.1.1 How to Deploy Stored Procedures 8-3
8.2.2 Configuration Steps for Oracle Procurement and Spend Analytics For Oracle EBS. 8-4
8.2.2.1 How to Configure the DAC parameter for Purchase Cycle Line 8-4
8.2.2.2 Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics 8-4
8.2.2.2.1 List of Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics 8-5
8.2.2.2.2 To configure domainValues_PaymentType_ora12.csv 8-6
8.2.2.2.3 To configure domainValues_Status_Purch_Approve_ora11i.csv 8-6
8.2.2.2.4 To configure domainValues_Status_Purch_Cycle_ora11i.csv 8-6
8.2.2.2.5 To configure domainValues_Xact_Types_Purch_Orders_ora11i.csv 8-7
8.2.2.2.6 To configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv .. 8-7
8.2.2.2.7 To configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv 8-8
8.2.2.2.8 To configure domainValues_Xact_Types_PO_Line_Type_CODE1_ora11i.csv 8-8
8.2.2.2.9 To configure domainValues_ShipmentType_ora12.csv 8-8
8.2.3 Configuration Steps for Oracle Procurement and Spend Analytics For PeopleSoft.. 8-9
8.2.4 Configuration Steps for Oracle Procurement and Spend Analytics For Universal ... 8-9
8.2.5 Configuration Steps for Controlling Your Data Set 8-9
8.2.5.1 Configuration Steps for Oracle Procurement and Spend Analytics For All Source Systems 8-9
8.2.5.1.1 About Configuring the Purchase Receipts Aggregate Table 8-10
8.2.5.1.2 How to Configure the Purchase Receipts Aggregate Table 8-11
8.2.5.1.3 About Configuring the Purchase Cycle Lines Aggregate Table 8-11
8.2.5.1.4 How to Configure the Purchase Cycle Lines Aggregate Table 8-12
8.2.5.2 Configuration Steps for Oracle Procurement and Spend Analytics For Oracle EBS 8-13
8.2.5.2.1 How to Extract Particular Purchase Order Records 8-13
8.2.5.3 Configuration Steps for Oracle Procurement and Spend Analytics For PeopleSoft 8-13
8.2.5.4 Configuration Steps for Oracle Procurement and Spend Analytics For Universal 8-13
8.2.5.4.1 About Configuring Oracle Procurement and Spend Analytics for Universal Source 8-14
8.2.5.4.2 How to Configure the Preferred Merchant Flag 8-15
8.2.5.4.3 How to Configure the Customer Billable Indicator 8-15
8.2.5.4.4 How to Configure the Receipts Indicator 8-15

8.2.5.4.5 How to Configure Expense Payment Types 8-16
8.2.5.4.6 How to Configure the Default Expense Distribution Percentage 8-16
8.2.5.4.7 How to Configure Lookup Dates for Currency Conversion 8-16

9 Configuring Oracle Financial Analytics

9.1 Overview of Oracle Financial Analytics 9-1
9.2 Configuration Required Before A Full Load for Financial Analytics 9-2
9.2.1 Configuration Steps for Financial Analytics For All Source Systems 9-2
9.2.2 Configuration Steps for Financial Analytics For Oracle EBS 9-2
9.2.2.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics 9-3
9.2.2.2 How to configure domainValues_Xact_Types_DocTypes_ora12.csv for Oracle General Ledger and Profitability Analytics 9-3
9.2.2.3 How to configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics 9-4
9.2.2.4 How to Specify the Ledger or Set of Books for which GL Data is Extracted 9-5
9.2.3 Configuration Steps for Financial Analytics For PeopleSoft 9-8
9.2.3.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics 9-8
9.2.3.2 How to configure Domain Values for Oracle Receivables Analytics 9-8
9.2.3.3 How to insert beginning GL Balances into Oracle Business Intelligence Applications 9-9
9.2.3.4 GL Account Dimension, Chartfields and Hierarchy for PeopleSoft 9-10
9.2.3.5 Group Account Configuration 9-12
9.2.4 Configuration Steps for Financial Analytics For Universal 9-12
9.2.5 Configuration Steps for Controlling Your Data Set 9-13
9.2.5.1 Configuration Steps for Financial Analytics For All Source Systems 9-13
9.2.5.1.1 How to include UnApplied Payments in AR Aging Calculations 9-13
9.2.5.1.2 How to Configure the AP/AR Aging Tables 9-13
9.2.5.2 Configuration Steps for Financial Analytics For Oracle EBS 9-14
9.2.5.2.1 How to Configure How GL Balances Is Populated in Oracle EBS 9-14
9.2.5.2.2 How to Configure Oracle Profitability Analytics Transaction Extracts 9-15
9.2.5.2.3 How to Configure Cost Of Goods Extract (Oracle EBS 11i-specific) 9-15
9.2.5.2.4 How to Configure AP Balance ID for Oracle Payables Analytics 9-16
9.2.5.2.5 How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics 9-17
9.2.5.2.6 How to Configure the AR Adjustments Extract for Oracle Receivables Analytics 9-17
9.2.5.2.7 How to Configure the AR Schedules Extract 9-18
9.2.5.2.8 How to Configure the AR Cash Receipt Application Extract for Oracle Receivables Analytics 9-18
9.2.5.2.9 How to Configure the AR Credit-Memo Application Extract for Oracle Receivables Analytics 9-19
9.2.5.2.10 How to Setup Drill Down in Oracle BI Answers from General Ledger to Subledger 9-19
9.2.5.3 Configuration Steps for Financial Analytics For PeopleSoft 9-20
9.2.5.3.1 About PeopleSoft Trees in Oracle Financial Analytics 9-20
9.2.5.3.2 GL ChartField Hierarchy and Internal Organization Hierarchy Customization Process 9-21

9.2.5.3.3 General Ledger Chartfield Hierarchy Extraction Customization 9-25
9.2.5.4 Configuration Steps for Financial Analytics Chain For Universal..... 9-27
9.2.5.4.1 About the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics..... 9-28
9.2.5.4.2 How to Configure the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics 9-29
9.2.5.4.3 How to Setup Drill Down in Oracle BI Answers from General Ledger to Subledger 9-29
9.2.5.4.4 How to configure Budgets for Oracle Profitability Analytics..... 9-29

10 Configuring Oracle Supply Chain and Order Management Analytics

10.1 Overview of Oracle Supply Chain and Order Management Analytics..... 10-1
10.2 Configuration Required Before A Full Load for Oracle Supply Chain and Order Management Analytics 10-2
10.2.1 Configuration Steps for Oracle Supply Chain and Order Management Analytics For All Source Systems 10-2
10.2.2 Configuration Steps for Oracle Supply Chain and Order Management Analytics For Oracle EBS..... 10-2
10.2.2.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics 10-3
10.2.2.2 How to Configure Invoice Type Domain Values 10-4
10.2.2.3 How to Configure Pick Types Domain Values 10-4
10.2.2.4 How to Configure Order Types Domain Values 10-5
10.2.2.5 How to Configure Pick Status Domain Values 10-5
10.2.2.6 How to Configure Invoice Status Domain Values..... 10-6
10.2.2.7 How to Configure Order Overall Status Domain Values..... 10-6
10.2.2.8 How to Configure Pay Method Domain Values..... 10-7
10.2.2.9 To Configure Movement Types Domain Values 10-8
10.2.2.10 How to Configure the Bill of Materials Explosion..... 10-8
10.2.2.11 How to Configure Quantity Types for Product Transactions 10-9
10.2.3 Configuration Steps for Oracle Supply Chain and Order Management Analytics For PeopleSoft 10-9
10.2.4 Configuration Steps for Oracle Supply Chain and Order Management Analytics For Universal 10-10
10.2.5 Configuration Steps for Controlling Your Data Set..... 10-10
10.2.5.1 Configuration Steps for Oracle Supply Chain and Order Management Analytics For All Source Systems 10-10
10.2.5.1.1 Tracking Multiple Attribute Changes in Bookings 10-11
10.2.5.1.2 Process of Aggregating Oracle Supply Chain and Order Management Analytics Tables..... 10-11
10.2.5.1.3 About Tracking Multiple Products for Oracle Supply Chain and Order Management Analytics..... 10-14
10.2.5.1.4 Adding Dates to the Order Cycle Time Table for Post-Load Processing. 10-15
10.2.5.1.5 About Configuring the Backlog Period Date for Oracle Supply Chain and Order Management Analytics 10-16
10.2.5.1.6 Example of How Backlog Data Is Stored in the Backlog History Table... 10-16
10.2.5.1.7 About Configuring the Customer Status History Fact for Post-Load Processing In Oracle Supply Chain and Order Management Analytics.. 10-18
10.2.5.1.8 Configuring the Customer Status History Fact table..... 10-18

10.2.5.1.9 How to Configure the Customer Status History Fact table 10-19
10.2.5.1.10 About Configuring the Inventory Monthly Balance Table 10-20
10.2.5.1.11 How to Configure the Inventory Monthly Balance..... 10-20
10.2.5.1.12 About Configuring the Product Transaction Aggregate Table 10-21
10.2.5.1.13 How to Configure the Product Transaction Aggregate Table 10-22
10.2.5.2 Configuration Steps for Oracle Supply Chain and Order Management Analytics For Oracle EBS..... 10-23
10.2.5.2.1 How to Configure Sales Order Lines Data Storage..... 10-23
10.2.5.2.2 About the Handling of Booked and Nonbooked Orders in the Order Lines and Bookings Table 10-24
10.2.5.2.3 How to Configure Sales Schedule Lines Data Storage 10-25
10.2.5.2.4 About the Handling of Booked and Nonbooked Orders in the Sales Schedule Lines Table..... 10-25
10.2.5.2.5 About Loading Bookings at the Schedule Line Level..... 10-26
10.2.5.2.6 How to Configure Early and Late Tolerances for Shipments..... 10-27
10.2.5.2.7 How to Configure Sales Invoice Lines Data Storage 10-28
10.2.5.2.8 How to Configure the Sales Invoice Extract..... 10-28
10.2.5.2.9 How to Configure the Order Lines Execution Plan and OTB ETL 10-28
10.2.5.2.10 How to Configure Procurement and Spend Analytics for Oracle Supply Chain and Order Management Analytics..... 10-29
10.2.5.2.11 How to Configure Oracle Financial Analytics for Oracle Supply Chain and Order Management Analytics 10-30
10.2.5.2.12 About Tracking Attribute Changes in Bookings 10-30
10.2.5.2.13 About Viewing the Data Warehouse Changes by Salesperson ID 10-30
10.2.5.2.14 How to Configure Different Types of Backlog Calculations 10-31
10.2.5.2.15 Adding Closed Orders to Backlog Calculations..... 10-32
10.2.5.2.16 How to Configure the Bill of Materials Explosion Options..... 10-33
10.2.5.2.17 How to Configure the Left Bound and Right Bound Calculation Option 10-37
10.2.5.3 Configuration Steps for Oracle Supply Chain and Order Management Analytics For PeopleSoft 10-38
10.2.5.4 Configuration Steps for Oracle Supply Chain and Order Management Analytics For Universal..... 10-38

11 Configuring Oracle Human Resources Analytics

11.1 Overview of Oracle Human Resources Analytics..... 11-1
11.2 Configuration Required Before A Full Load for Oracle HR Analytics 11-2
11.2.1 Configuration Steps for Oracle HR Analytics For All Source Systems..... 11-2
11.2.2 Configuration Steps for Oracle HR Analytics For Oracle EBS..... 11-2
11.2.2.1 About Domain Values and CSV Worksheet Files for Oracle HR Analytics 11-3
11.2.2.2 How to Configure the domainValues_Employee_Ethnic_Group_Code_ora<ver>.csv 11-8
11.2.2.3 How to Configure the domainValues_Employee_Sex_MF_ora<ver>.csv..... 11-8
11.2.2.4 How to Configure the domainValues_Employee_Veteran_Status_Code_ora<ver>.csv 11-9
11.2.2.5 How to Configure the domainValues_Employment_Cat_ora<ver>.csv 11-9
11.2.2.6 How to Configure the domainValues_Employment_Exempt_Flg_ora<ver>.csv 11-10

11.2.2.7 How to Configure the domainValues_Employment_Full_Time_Flg_ora<ver>.csv 11-10
11.2.2.8 How to Configure the domainValues_Employment_Status_ora<ver>.csv.... 11-10
11.2.2.9 How to Configure the domainValues_EventTypes_ora<ver>.csv 11-11
11.2.2.10 How to Configure the domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv 11-11
11.2.2.11 How to Configure the domainValues_HRPosition_Exempt_Flg_ora<ver>.csv..... 11-11
11.2.2.12 How to Configure the domainValues_Job_Eeo_Cat_Code_ora<ver>.csv..... 11-12
11.2.2.13 How to Configure the domainValues_Job_Flsa_Stat_Code_ora<ver>.csv 11-12
11.2.2.14 How to Configure the domainValues_Pay_Type_Grp_Code_ora<ver>.csv . 11-12
11.2.2.15 How to Configure the domainValues_Pay_Type_Flg_ora<ver>.csv 11-13
11.2.2.16 How to Configure Address Types for HR Profile 11-13
11.2.2.17 How to Configure Phone Types for HR Profile 11-14
11.2.2.18 How to Configure Education Degree Codes for Employee Dimension..... 11-15
11.2.2.19 About Configuring Key Flexfields 11-16
11.2.2.20 How to Configure the Key Flexfields for the Job Dimension 11-16
11.2.2.21 How to Configure the Key Flexfields for the HR Position Dimension..... 11-17
11.2.2.22 How to Configure the Key Flexfields for the Pay Grade Dimension 11-17
11.2.2.23 How to Configure multi-segmented Flexfields 11-18
11.2.2.24 How to Configure Flags for the Pay Type Dimension..... 11-19
11.2.2.25 How to Configure Classification Names for Payroll..... 11-19
11.2.3 Configuration Steps for Oracle HR Analytics For PeopleSoft 11-21
11.2.3.1 Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM 11-22
11.2.3.2 How to configure the domainValues_Employee_Ethnic_Group_Code_psft.csv..... 11-24
11.2.3.3 How to configure the domainValues_Employee_Sex_MF_psft.csv 11-25
11.2.3.4 How to configure the domainValues_Employee_Veteran_Status_Code_psft.csv 11-25
11.2.3.5 How to configure the domainValues_Employment_Category_psft.csv..... 11-25
11.2.3.6 How to configure the domainValues_Employment_Exempt_Flg_psft.csv.... 11-26
11.2.3.7 How to configure the domainValues_Employment_Full_Time_Flg_psft.csv 11-26
11.2.3.8 How to configure the domainValues_Employment_Status_psft.csv 11-26
11.2.3.9 How to configure the domainValues_Emp_EventType_Grp_Code_psft.csv 11-27
11.2.3.10 How to configure the domainValues_Emp_EventType_SubG_Code_psft.csv 11-27
11.2.3.11 How to configure the domainValues_HRPosition_Active_Pos_Flg_psft.csv 11-28
11.2.3.12 How to configure the domainValues_HRPosition_Exempt_Flg_psft.csv 11-28
11.2.3.13 How to configure the domainValues_Job_Eeo_Cat_Code_psft.csv 11-28
11.2.3.14 How to configure the domainValues_Job_Flsa_Stat_Code_psft.csv 11-29
11.2.3.15 How to configure the domainValues_Pay_Type_Grp_Code_psft.csv 11-29
11.2.3.16 How to configure the domainValues_Emp_EventType_EventReason_Code_psft.csv..... 11-30
11.2.3.17 How to configure the domainValues_Emp_EventType_EventClass_Code_psft.csv 11-30
11.2.3.18 How to configure the file_pension_comp_flg.csv 11-31

11.2.3.19 How to Configure Compensation Flag and Pension Compensation Flag for Pay Type Dimension 11-31
11.2.3.20 How to Configure the Performance Review Type ETL runtime parameter... 11-31
11.2.3.21 How to configure Event Types using domainValues_EventTypes_psft.csv . 11-32
11.2.4 Configuration Steps for Oracle HR Analytics For Universal 11-32
11.2.5 Configuration Steps for Controlling Your Data Set..... 11-32
11.2.5.1 Configuration Steps for Oracle HR Analytics For All Source Systems 11-33
11.2.5.1.1 How to Aggregate the Payroll Table for Oracle HR Analytics..... 11-33
11.2.5.1.2 About Configuring the Employee Snapshot Fact tables..... 11-34
11.2.5.1.3 How to control the rolling period for storing Employee Daily Snapshot data..... 11-35
11.2.5.1.4 How to control the granularity of the Employee Monthly Snapshot data..... 11-35
11.2.5.2 Configuration Steps for Oracle HR Analytics For Oracle EBS..... 11-35
11.2.5.3 Configuration Steps for Oracle HR Analytics For PeopleSoft 11-36
11.2.5.3.1 How to Tune Performance for the Payroll table for PeopleSoft HCM HR Analytics 11-36
11.2.5.4 Configuration Steps for Oracle HR Analytics For Universal..... 11-36

12 Configuring Oracle Sales Analytics

12.1 Overview of Oracle CRM..... 12-1
12.2 Configuration Required Before A Full Load..... 12-1
12.2.1 How to Deploy Stored Procedures 12-1
12.2.2 Configuration Steps for Controlling Your Data Set 12-2
12.2.2.1 How to configure the snapshot frequency in Pipeline fact 12-2
12.3 Configuring Usage Accelerator 12-2
12.3.1 What is Usage Accelerator?..... 12-2
12.3.1.1 Configuring Usage Accelerator..... 12-3
12.3.1.1.1 Tracking of Targeted Users..... 12-3
12.3.1.1.2 Other Siebel Required Responsibilities..... 12-3
12.3.1.2 About Deployment Date 12-6
12.3.1.3 About Metric Definitions and Variables 12-6
12.3.1.3.1 User Adoption Metrics 12-6
12.3.1.3.2 Opportunity Updates Metrics 12-7
12.3.1.3.3 Completeness Metrics..... 12-7
12.3.1.3.4 Coverage Metrics..... 12-7
12.3.2 About the Usage Accelerator Execution Plan..... 12-8
12.3.3 Using Usage Accelerator with Siebel 7.5.3 Applications 12-8
12.3.3.1 Usage Accelerator Application Integration with Siebel 7.5.3 CRM Application..... 12-8
12.3.3.2 Usage Accelerator-Analytics Adapter Files..... 12-10

13 Configuring Oracle Contact Center Telephony Analytics

13.1 Overview of Oracle Contact Center Telephony Analytics..... 13-1
13.2 Configuration Required Before A Full Load for Contact Center Telephony Analytics 13-1
13.2.1 About the Dimension Key Resolution Process for Universal Source 13-2
13.2.2 How to Load the Dimension Table 13-2

13.2.3 How to Load the Fact Table ... 13-3
13.2.4 Setting Up The DATASOURCE_NUM_ID Parameter For The Universal Adapter 13-3
13.2.5 About Configuring the Event Type Columns ... 13-3
13.2.5.1 The Activity Type Column ... 13-3
13.2.5.2 Call Type (or Contact Type) Column ... 13-4
13.2.5.3 Call Event Type Column ... 13-4
13.2.5.4 After-Call Work Activity (ACW) Type Column ... 13-5
13.2.6 About Configuring The Contact Reason Column ... 13-5
13.2.7 About Configuring Contact Status Type Column ... 13-6
13.2.8 Setting Up the Representative Activities Table ... 13-6
13.2.9 Setting Up the Contact Center Performance Table ... 13-7
13.2.10 Setting Up the Benchmarks and Targets Table ... 13-8
13.2.11 Logical Delete of Records From Fact Tables ... 13-10
13.2.12 How to Configure Dates and Times ... 13-11
13.2.13 Configuring Flags for Oracle Contact Center Telephony Analytics ... 13-12
13.2.14 How to Exclude Representative Data for Post-Load Processing ... 13-13
13.2.15 How to Configure Alternate Load Plan to Enable Direct Load to the Entry-Level Rep Activity Aggregates ... 13-13
13.2.16 How to Configure Alternate Load Plan to Enable Direct Load to the Entry-Level Contact Performance Aggregates ... 13-14
13.2.17 How to Set Up the Integration_Id of the Contact Center Telephony Facts in the Warehouse ... 13-15
13.2.18 Setting Up the IVR Menu and IVR Navigation Profile Tables ... 13-16
13.2.18.1 Determining Navigation Profile and Transfer Flag for a Call ... 13-16
13.2.18.2 Slowly Changing Dimension Behavior of IVR Menu and IVR Navigation Profile Dimensions ... 13-17
13.2.19 How to View Duration Metrics in hh:mm:ss Format in Oracle BI Answers ... 13-17
13.2.20 Configuration Steps for Controlling Your Data Set ... 13-18

14 Configuring the Oracle Business Intelligence Applications Repository

14.1 Modifying the Oracle BI Repository for Siebel version 6.3 and version 7.5 applications ... 14-1
14.2 Configuring the Oracle BI Repository Connections ... 14-2
14.2.1 About The Predefined Connection Pools In The Oracle Business Analytics Warehouse ... 14-2
14.2.1.1 How to Configure Dynamic Data Source Names ... 14-3
14.2.2 How to Configure Oracle Business Intelligence Applications Repository Connections ... 14-4
14.2.3 How to Configure Oracle Business Intelligence Applications Repository Variables ... 14-5
14.3 Setting up Date Specific Metrics ... 14-5
14.4 Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse 14-6
14.5 Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse... 14-7
14.6 About the Period Ago Keys for Oracle Business Analytics Warehouse ... 14-7
14.7 About Oracle BI Time Repository Variables ... 14-8
14.8 About User Authentication ... 14-9
14.9 About the Security or Visibility Configuration ... 14-9
14.10 About the Group Variable ... 14-12

14.11 About Configuring Usage Tracking for Oracle Business Analytics Warehouse 14-13
14.12 About the Incremental Deployment of the Oracle BI Applications Repository 14-13

A Supporting Tasks for DAC and Informatica PowerCenter

A.1 How to Log Into the DAC Client A-1
A.2 How to Start and Stop the DAC Server A-2
A.3 How to Log Into Informatica PowerCenter Administration Console A-3
A.4 How to Start and Stop Informatica Services A-4
A.5 How to Stop and Start the Informatica Repository Service (Optional) A-4
A.6 About the DAC Server Scripts for Bash and C Shells A-5
A.7 How to Configure the DAC Metadata Repository A-7
A.7.1 How to Set Up DAC System Properties A-7
A.7.2 How to Activate Join Indices for Teradata Databases A-8

B Configuring Universal Adapter for CRM Analytics

B.1 What is Universal Adapter for CRM Analytics? B-1
B.2 Data Format Requirements B-1
B.3 About The Sample Source Files Provided B-2
B.4 How to Create Data Files for Universal Adapter for CRM Analytics B-2
B.5 List of Sample Source Files for Universal Adapter for CRM Analytics B-2
B.5.1 List of Source Files for Seed Data and Common Dimensions B-3
B.5.2 List of Source Files for Specific Subject Areas B-3

C Localizing Oracle Business Intelligence Deployments

C.1 Process of Maintaining Translation Tables for Oracle BI C-1
C.1.1 Upgrading Oracle Business Intelligence Seed Data for Non-English Locales C-2
C.1.1.1 Importing Locale Seed Data Into The Translation Table (W_LOCALIZED_STRING_G) C-2
C.1.2 Externalizing Customer Metadata Strings C-3
C.1.3 Adding Custom Translations to the W_LOCALIZED_STRING_G Table C-4
C.1.3.1 Adding String Translations for Analytics Metadata C-4
C.2 About Translating Presentation Services Strings C-5
C.3 Changing the Default Currency in Analytics Applications C-5

D Integrating Interactive Dashboards and Operational Applications Data

D.1 Importing Oracle's Siebel Industry Applications Seed Data D-1
D.2 Completing the Initialization in the Siebel Operational Application D-2
D.2.1 Updating the Siebel Operational Application D-2
D.2.2 Reapplying Customized Style Sheets D-3
D.2.3 How to Configure Oracle Business Intelligence with Oracle's Siebel Web Extension (SWE) D-4
D.2.4 Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services D-5
D.2.5 Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE D-5
D.2.6 How to Configure Oracle BI Action Links D-6
D.2.7 Testing Action Links D-6

D.2.8 Process of Integrating Analytics and Oracle's Siebel Operational Applications D-6
D.2.9 Accessing Optional Analytics Applications D-7
D.2.10 Configuring the Browser Settings for Operational Applications D-7
D.2.11 Process of Customizing Oracle BI Application Home Page and Dashboards D-8
D.2.12 Customizing Oracle BI Content on the Siebel Operational Application Home Page D-8
D.2.13 Determining the Oracle BI Report Path Argument D-10
D.2.14 Adding Views for Custom Oracle BI Interactive Dashboards D-11
D.2.15 Verifying NQHOST Setup for Pharma Disconnected Analytics Client D-11
D.2.16 Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client D-12
D.2.17 Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client D-13
D.3 Viewing Information About Preconfigured Dashboards D-14
D.3.1 Exposing Dashboards and Requests D-14
D.3.2 Locating Dashboard and Page Names D-14

E Configuring Metadata for Analytics Applications

E.1 What Are Oracle's Siebel Operational Applications? E-1
E.2 Updating Financial Analytics Logical Table Sources E-2
E.3 Externalizing Financial Services Metadata Translation Strings E-3
E.4 Disabling the Financial Analytics Logical Table Sources E-3
E.5 Developing and Deploying Predictive Scores E-4
E.6 Analytics Metadata Requirements for Oracle's Siebel Industry Applications E-5
E.6.1 Oracle's Siebel Consumer Sector Dashboards and Pages E-5
E.6.2 Oracle's Siebel Consumer Sector Data Requirements E-7
E.6.3 Oracle's CME family of products (Oracle Communications, Media and Energy Sales Analytics E-7
E.6.4 Oracle's Siebel CME Dashboards and Pages E-9
E.6.5 Oracle's Siebel CME Data Requirements E-10
E.6.6 Oracle Pharma Sales Analytics Dimensions E-10
E.6.6.1 Positions Dimension E-10
E.6.6.2 Alignments Dimension E-10
E.6.6.3 Products Dimension E-11
E.6.6.4 Product Costs Dimension E-11
E.6.7 Dimensions Specific to Subject Areas in Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics E-12
E.6.7.1 Pharma Sales Effectiveness E-12
E.6.7.2 Pharma Product Categories E-12
E.6.7.3 Pharma Promotional Effectiveness E-12
E.6.7.4 Pharma Medical Education Effectiveness E-13
E.6.7.5 Pharma Objectives Achievement` E-13

F Oracle Business Analytics Warehouse for Life Sciences Data Considerations

F.1 What's New for Pharma Analytics in Version 7.9 ..... F-1
F.2 Oracle Business Analytics Warehouse for Life Sciences Overview F-1
F.3 Importing Syndicated Data into Oracle Business Analytics Warehouse F-2

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

G Pharma Disconnected Analytics Administration - Supplemental Information

G.1 About Pharma Disconnected Analytics G-1
G.2 Sourcing Reports for Oracle's Pharma Disconnected Analytics G-2

H Using Oracle Business Analytics Warehouse Exception Reports

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

I Troubleshooting Oracle Business Intelligence Applications

I.1 Oracle Business Analytics Warehouse Installation and Initial Configuration Issues I-2
I.2 Informatica and Loading Issues I-2
I.3 Unicode Issues with Informatica and the Oracle Business Analytics Warehouse I-6
I.4 UNIX Installation Issues with the Oracle Business Analytics Warehouse I-7
I.5 Oracle Business Analytics Warehouse Upgrade Issues I-8
I.6 Oracle BI Interactive Dashboard Issues I-9
I.7 ETL Issues I-10

J About the Versioned Informatica Repository

J.1 Summary of Versioning in the Informatica Repository J-1
J.2 Customization of Repository Objects Using Check Out and Check In J-1

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.

Oracle Business Intelligence Applications Fusion Edition Installation and Configuration Guide contains instructions on installing, configuring, and customizing Oracle Business Intelligence Applications Version 7.9.5. For a high level road map of the steps required to install, configure, and customize Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications".

Oracle recommends reading the *Oracle Business Intelligence Applications Fusion Edition Release Notes* before installing, using, or upgrading Oracle Business Intelligence Applications. The *Oracle Business Intelligence Applications Fusion Edition 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` (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 Release 7.9.5 documentation set (available at `http://www.oracle.com/technology/documentation/bi_apps.html`):

- *Oracle Business Intelligence Applications Fusion Edition Release Notes*
- *System Requirements and Supported Platforms for Oracle Business Intelligence Applications Fusion Edition*
- *Oracle Business Intelligence Data Warehouse Administration Console Guide*
- *Oracle Business Intelligence Applications Fusion Edition Upgrade Guide*

Conventions

The following text conventions are used in this document:

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

Part I

Getting Started

Part 1 helps you get started with Oracle Business Intelligence Applications, and contains the following chapters:

- Chapter 1, "What's New in This Release"
- Chapter 2, "Overview of Oracle Business Intelligence Applications"

Note: For a high level road map for installation, configuration, and customization steps for Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications".

1

What's New in This Release

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

1.1 What's New In Oracle Business Intelligence Applications Version 7.9.5?

The following list shows the main changes in Oracle Business Intelligence Applications Version 7.9.5:

- New Adapters:
  - Siebel CRM 8.1.
  - Financial Analytics for PeopleSoft 8.9 and 9.0.
  - Human Resources for PeopleSoft 8.9 and 9.0.
- Documentation. The Oracle Business Applications Installation and Configuration Guide has the following improvements:
  - Streamlined installation steps for Oracle Business Intelligence Applications and Informatica PowerCenter 8.1.1 SP4.
  - An example topology for Oracle Business Intelligence Applications components (for more information, see Section 4.1, "About Oracle Business Intelligence Applications Topologies").
- Informatica:
  - Oracle Business Intelligence Applications Version 7.9.5 requires Informatica PowerCenter Version 8.1.1 SP4.

2

Overview of Oracle Business Intelligence Applications

This chapter provides an overview of Oracle Business Intelligence Applications, and 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 Business Intelligence Applications"
- Section 2.5, "Using Repository Documentation"

2.1 What is Oracle Business Intelligence Applications?

Oracle Business Intelligence Applications is a prebuilt business intelligence solution.

Oracle Business Intelligence Applications supports 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.

Table 2–1 (Cont.) Oracle Business Intelligence Applications Components

Component	Description
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 (EnterpriseBusinessAnalytics.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 (for more information, see Section 2.2, "Oracle Business Analytics Warehouse Overview").

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.

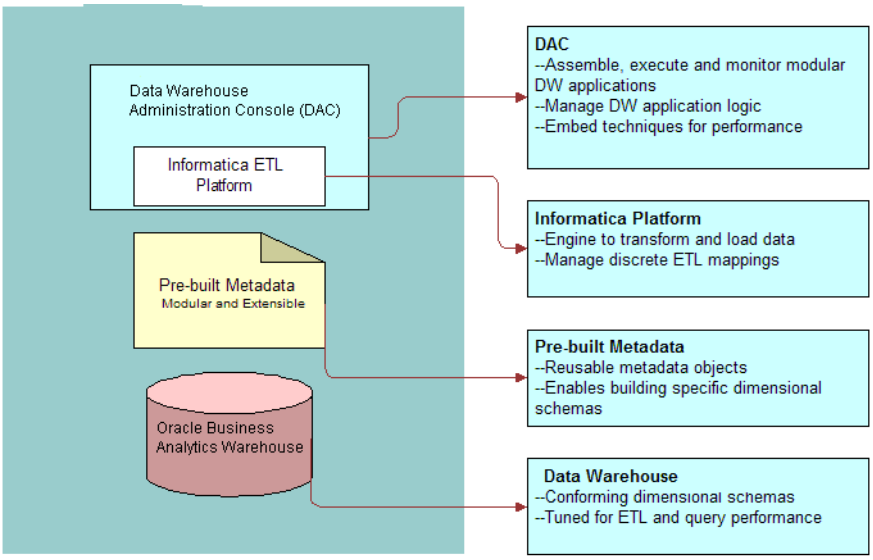
The Oracle Business Analytics Warehouse includes the following (see figure below):

- 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 Fusion Edition 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.

Figure 2–1 An Overview Of the Oracle Business Analytics Warehouse



Tip: Once you have installed Oracle Business Intelligence Applications, use the Repository Documentation option in Oracle BI Administration Tool to list the repository objects in a text file (for more information, see Section 2.5, "Using Repository Documentation".

2.3 Oracle Business Analytics Warehouse Architecture

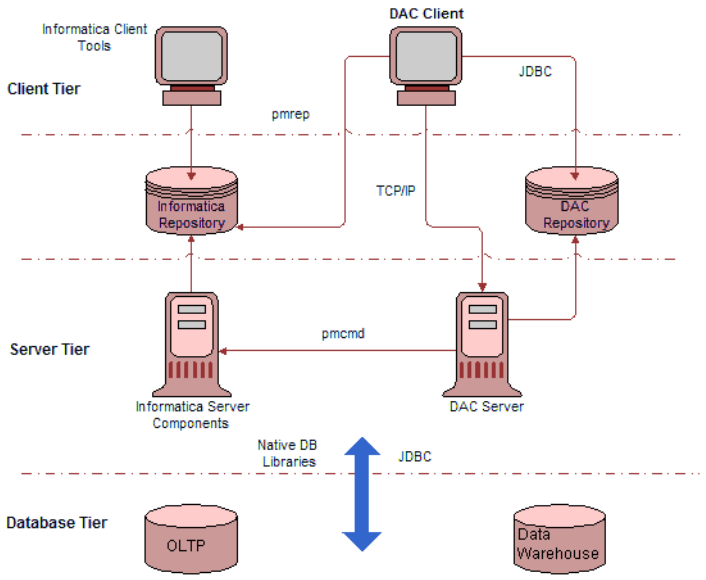
High-level analytical queries, like those commonly used in Oracle Business Intelligence, scan and analyze large volumes of data using complex formulas. This process can take a long time when querying a transactional database, which impacts overall system performance.

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 PowerCenter Services:**
  - Integration Services - The Integration Service reads workflow information from the repository. The Integration Service connects to the repository through the Repository Service to fetch metadata from the repository.
  - Repository Services - The Repository Service manages connections to the PowerCenter repository from client applications. The Repository Service is a separate, multi-threaded process that retrieves, inserts, and updates metadata in the repository database tables.
- **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 Business Intelligence Applications

To install, configure, and customize Oracle Business Intelligence 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 Business Intelligence Applications".
- Install and set up the Oracle Business Intelligence Applications components and Informatica PowerCenter components as described in Chapter 4, "Installing and Setting Up Oracle Business Intelligence Applications".

  Note: For an example that shows a typical deployment topology for Oracle Business Intelligence Applications, see Section 4.1, "About Oracle Business Intelligence Applications Topologies".
- (Mandatory) Perform the source-independent steps that are required before a full data load, described in Section 7.1.1, "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 7.2.1, "Configuration Required Before A Full Load for Oracle EBS"
  - Section 7.3.1, "Configuration Required Before A Full Load for PeopleSoft"
  - Section 7.4.1, "Configuration Required Before A Full Load for Siebel Source Systems"
- (Optional) Perform any required additional source-independent steps described in Section 7.1.2, "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 7.2.2, "Configuration Steps for Controlling Your Data Set for Oracle EBS"
  - Section 7.3.2, "Configuration Steps for Controlling Your Data Set for PeopleSoft"
  - Section 7.4.2, "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 8, "Configuring Oracle Procurement and Spend Analytics"
  - Chapter 9, "Configuring Oracle Financial Analytics"
  - Chapter 10, "Configuring Oracle Supply Chain and Order Management Analytics"
  - Chapter 11, "Configuring Oracle Human Resources Analytics"
  - Chapter 12, "Configuring Oracle Sales Analytics"
  - Chapter 13, "Configuring Oracle Contact Center Telephony Analytics"
- (Optional) Perform any required additional steps for all applications that you deploy, from the appropriate chapters below:
  - Chapter 8, "Configuring Oracle Procurement and Spend Analytics"
  - Chapter 9, "Configuring Oracle Financial Analytics"

  - Chapter 10, "Configuring Oracle Supply Chain and Order Management Analytics"
  - Chapter 11, "Configuring Oracle Human Resources Analytics"
  - Chapter 12, "Configuring Oracle Sales Analytics"
  - Chapter 13, "Configuring Oracle Contact Center Telephony Analytics"
- (Optional) If you want to customize the out-of-the-box Oracle Business Intelligence Applications functionality, follow the steps described in Chapter 5, "Customizing the Oracle Business Analytics Warehouse".
- (Optional) If you want to modify the out-of-the-box Oracle Business Intelligence Applications security, follow the steps described in Chapter 6, "Integrated Security for Oracle Business Intelligence Applications".

Once you have installed and configured Oracle Business Intelligence Applications components, configured the modules (optional), and customized Oracle Business Intelligence Applications (optional), you are ready to start running ETLs.

For a detailed example of how to run an ETL for Oracle Financials with an Oracle EBS OLTP data source, see Section 4.16, "About Running A Full Load ETL".

For information about running ETLs, see *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

2.5 Using Repository Documentation

When you deploy Oracle Business Intelligence Applications, you can use the following documentation and tools to manage your metadata:

- Repository Documentation

  Using Oracle BI Administration Tool, you can generate repository documentation that lists the mapping from the presentation columns to the corresponding logical and physical columns. You might use this information for gap-analysis, or to create a record of your repository that you can use to compare with other repositories.

  To generate Repository Documentation into a text or comma-separated file, log into Oracle BI Administration Tool and choose Tools, then Utilities, then Repository Documentation.

  For more information about generating repository documentation, see *Oracle Business Intelligence Server Administration Guide*.

- Presentation Services Catalog

  Using the Catalog Manager, you can view the names of the prebuilt dashboards and requests in the Presentation Services Catalog.

  To view the Presentation Catalog, select Catalog Manager from the Windows\ Start\Programs\Oracle Business Intelligence menu.

Part II

Installation and Post-Installation Steps For BI Applications

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

- Chapter 3, "Pre-installation and Predeployment Requirements for Oracle Business Intelligence Applications"
- Chapter 4, "Installing and Setting Up Oracle Business Intelligence Applications"

Note: For a high level road map for installation, configuration, and customization steps for Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications".

3

Pre-installation and Predeployment Requirements for Oracle Business Intelligence 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.

You also need to satisfy the database and Informatica PowerCenter requirements that are specified in Chapter 4 (for more information, see Section 4.3, "Mandatory Requirements and Pre-installation Tasks").

Notes

- For information about Oracle-specific database settings, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications Fusion Edition*.
- For information about code page settings, refer to the Informatica documentation.

This chapter contains the following topics:

- Section 3.1, "Operating System, Driver, and Connectivity Requirements for Oracle Business Analytics Warehouse"
- Section 3.2, "General Guidelines for Setting Up Oracle Business Analytics Warehouse"
- Section 3.3, "IBM DB2 UDB-Specific Database Guidelines for Oracle Business Analytics Warehouse"
- Section 3.4, "IBM DB2 UDB zOS and OS/390 and Z/os-Specific Database Guidelines for Oracle Business Analytics Warehouse"
- Section 3.5, "Additional Suggestions for Optimizing Oracle Performance in Oracle Business Analytics Warehouse"
- Section 3.6, "SQL Server-Specific Database Guidelines for Oracle Business Analytics Warehouse"
- Section 3.7, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse"
- Section 3.8, "Miscellaneous Information About Oracle Business Intelligence Applications Deployments"
- Section 3.9, "Partitioning Guidelines For Large Fact Tables"

3.1 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 Fusion Edition*.

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

Component	Operating System	Software	Connectivity and Driver
ETL Servers	▪ Solaris ▪ AIX ▪ HP ▪ Windows ▪ Linux	▪ Informatica PowerCenter Integration Service ▪ DAC Server	▪ Java SDK ▪ JDBC drivers ▪ Javamail
ETL Clients	Windows	▪ Informatica client tools ▪ DAC client	▪ Java SDK ▪ JDBC drivers ▪ ODBC drivers
Oracle Business Analytics Warehouse	▪ Solaris ▪ AIX ▪ HP ▪ Windows ▪ OS/390, Z/os	Database software with Oracle Business Analytics Warehouse	Not applicable
ETL Repositories	▪ Solaris ▪ AIX ▪ HP ▪ Windows	Database software with ETL repositories	Not applicable

3.2 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 4.10.1, "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.

3.3 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	

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

Parameter	DB2 UDB V7	DB2 UDB V8	Notes
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	
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.

3.4 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	
CDSSRDEF	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.

3.5 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))
```

3.6 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 3.6.1, "Setting the ANSI NULL Option"
- Section 3.6.2, "Modifying the DB Library Options Setting"
- Section 3.6.3, "Recommended SQL Server Database Parameters"

3.6.1 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.

3.6.2 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.

3.6.3 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	
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	

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

Parameter	Recommended Setting	Notes
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.

3.7 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 3.7.1, "General Guidelines for Teradata Deployments"
- Section 3.7.2, "Best Practices For Teradata Deployments"

3.7.1 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 Business Intelligence Applications in a Teradata environment, you must load the pre-built Informatica repository file \dwrep\Oracle_BI_DW_Teradata.rep.
- 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 Client, refer to Section 4.15.9, "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.

3.7.2 Best Practices For Teradata Deployments

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

- Section 3.7.2.1, "LEFT OUTER JOIN Issue"
- Section 3.7.2.2, "Database Statistics"
- Section 3.7.2.3.2, "Group By Vs Distinct"
- Section 3.7.2.3.3, "Pruning Of Mapping"
- Section 3.7.2.3.4, "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.

3.7.2.1 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.

3.7.2.2 Database Statistics

This section describes database statistics.

3.7.2.3 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.

3.7.2.3.1 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.

3.7.2.3.2 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.

3.7.2.3.3 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.

3.7.2.3.4 Loader Configurations This section explains the loaders that are available in Teradata, and how they are used in Oracle Business Intelligence Applications.

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

- Fastload
- Mload
- Tpump

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).

3.7.2.3.5 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 Fastlload 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.

3.7.2.3.6 Mload The Mload process is slower than Fastload but quicker than Tpump. 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.

3.7.2.3.7 Tpump The Tpump process is slower than Mload but faster than ODBC. The Tpump 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, Tpump starts loading data from the last committed data.

Tpump can be used in the following modes:

- Tpump_Insert: Use to do inserts.
- Tpump_Update: Use to do updates (this mode requires you to define the primary key in the Informatica target table definition).
- Tpump_Upsert: Use to do update otherwise insert (this mode requires you to define the primary key in the Informatica target table definition).

- Tpump_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 Tpump 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 Tpump 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.

3.8 Miscellaneous Information About Oracle Business Intelligence Applications Deployments

This section contains the following topics:

- Section 3.8.1, "Preconfigured Mapping for the SA System Subject Area"
- Section 3.8.2, "Using Initialization Blocks"
- Section 3.8.3, "Performance Improvements for Change Capture for Incremental ETL Run"
- Section 3.8.4, "Running an ETL with a Non-English OLTP Data Source"

3.8.1 Preconfigured Mapping for the SA System Subject Area

For Oracle's Siebel Applications customers, Table 3–6 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 3–6 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
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	

3.8.2 Using Initialization Blocks

Table 3–7 lists some of the initialization blocks common to all Oracle Business Intelligence applications and their purposes. Initialization blocks that are specific to each Oracle Business Intelligence Applications area are not listed here.

Table 3–7 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.

Table 3–7 (Cont.) Initialization Blocks and Their Purposes

Initialization Block	Purpose
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.
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.

3.8.3 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.

3.8.3.1 Implementing Transactional Database (OLTP) Indexes

To implement indexes, you use SQL files that are available in the \OracleBI\dwrep directory. Table 3–8 describes the SQL files that are appropriate for the specified applications.

Table 3–8 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.

3.8.3.2 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 3–9 shows the indexes that, based on the SQL above, are created on the S_ CONTACT table by the SQL generation scripts.

Table 3–9 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

3.8.4 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).

3.9 Partitioning Guidelines For Large Fact Tables

This section explains how to use partitioning to maximize performance in your Oracle Business Intelligence Applications deployment, and contains the following topics:

- Section 3.9.1, "Introduction to Partitioning Large Fact Tables"
- Section 3.9.2, "How to Partition Large Fact Tables"
- Section 3.9.3, "Configure DAC To Support ETL For Partitioned Tables"
- Section 3.9.4, "Example XML Files"

3.9.1 Introduction to Partitioning Large Fact Tables

Taking advantage of range partitioning for fact tables will not only reduce index and statistics maintenance time during ETL, but also improve web queries performance. Since the majority of inserts and updates impact the last partition(s), you would need to disable only local indexes on a few impacted partitions, and then rebuild disabled indexes after the load and compute statistics on updated partitions only. Online reports and dashboards should also render results faster, since the optimizer would build more efficient execution plans using partitions elimination logic.

Large fact tables, with more than 10 million rows, are good candidates for partitioning. To build an optimal partitioned table with reasonable data distribution, you can consider partitioning by month, quarter, year, etc. You can either identify and partition target fact tables before initial run, or convert the populated tables into partitioned objects after the full load.

To implement the support for partitioned tables in Oracle Business Analytics Data Warehouse, you need to update DAC metadata and manually convert the candidates into partitioned tables in the target database.

To Deploy Partitioned Fact Tables:

1. Partition the Large Fact Tables (for more information, see Section 3.9.2, "How to Partition Large Fact Tables").
2. Configure the DAC to support ETL for partitioned tables (for more information, see Section 3.9.3, "Configure DAC To Support ETL For Partitioned Tables").

3.9.2 How to Partition Large Fact Tables

If you have large fact tables that are affecting performance, you can maximize performance by partitioning the fact tables as described in this section.

1. Identify a partitioning key and decide on a partitioning interval.

   Choosing the correct partitioning key is the most important factor for effective partitioning, since it defines how many partitions will be involved in web queries or ETL updates. Review the following guidelines for selecting a column for a partitioning key:

- Identify eligible columns of type DATE for implementing range partitioning.
- Connect to the Oracle BI Server repository and check the usage or dependencies on each column in the logical and presentation layers.
- Analyze the summarized data distribution in the target table by each potential partitioning key candidate and data volumes per time range, month, quarter or year.
- Based on the compiled data, decide on the appropriate partitioning key and partitioning range for your future partitioned table.
- The recommended partitioning range for most implementations is a month.

The following table summarizes the recommended partitioning keys for some large Oracle Business Intelligence Applications Fact tables:

Table 3–10 Recommended partitioning keys for large fact tables

Area	Table Name	Partitioning Key
Financials	W_AP_XACT_F	POSTED_ON_DT_WID
Financials	W_AR_XACT_F	POSTED_ON_DT_WID
Financials	W_GL_REVN_F	POSTED_ON_DT_WID
Financials	W_GL_COGS_F	POSTED_ON_DT_WID
Financials	W_TAX_XACT_F	ACCT_PERIOD_END_DT_WID
Financials	W_GL_OTHER_F	ACCT_PERIOD_END_DT_WID
Sales	W_SALES_INVOICE_LINE_F	INVOICED_ON_DT_WID
Sales	W_SALES_SCHEDULE_LINE_F	ORDERED_ON_DT_WID
Sales	W_REVN_F	CLOSE_DT_WID

2. Identify future global and local indexes.

   Determine all active indexes on a partitioned table candidate (W_REVN_F in the example) using the following query in the DAC schema:

```
select I.NAME, C.NAME, C.POSTN, I.ISUNIQUE, I.ISBITMAP
from W_ETL_TABLE T, W_ETL_INDEX I, W_ETL_INDEX_COL C
where T.ROW_WID = I.TABLE_WID
and T.NAME='W_REVN_F'
and I.ROW_WID=C.INDEX_WID
and I.INACTIVE_FLG = 'N'
order by I.NAME, C.POSTN;
```

   Review the query output and choose an existing unique index as a global unique index for your partitioned table. All bitmap indexes must be re-created as local partitioned indexes. If the query returns B-Tree indexes other than the chosen candidate for a global index, consider creating them as local indexes as well.

3. Create a partitioned table.

You can pre-create a partitioned table prior to the initial load, or load data into the regular table and then create its partitioned copy and migrate the summarized data. If you have already completed the initial load into a regular table and then decided to partition it, you do **not** need to re-run the initial load.

You can consider two options to convert a table into a partitioned one:

- Create table as select.

  This method is simpler and faster.

- Create table exchange partition syntax and then split partitions.

  This method is suitable in high availability data warehouses when you have to carry out partitioning with end users accessing the data.

Note: It is important to make sure that your partition names include the values from the DAC system variables CURRENT_MONTH_WID and PREVIOUS_MONTH_WID, which are defined in the next section. These variables will be used in custom XML files to handle local and global indexes using syntax: 'partition PART_@DAC_CURRENT_MONTH_WID' and 'partition PART_@DAC_PREVIOUS_MONTH_WID'. Note that the DDL syntax must have the prefix 'DAC_' in the custom XML files.

You should create enough partitions to cover the time range for the summarized data. You should create additional partitions for future ranges as well. You may also consider distributing the partitions across different tablespaces.

The syntax in the following SQL commands assumes everything in one tablespace USERS:

```
SQL> rename W_REVN_F to W_REVN_F_ORIG;
SQL> create table W_REVN_F
partition by range (CLOSE_DT_WID) (
partition PART_MIN    values less than (200601),
partition PART_200601 values less than (200602),
partition PART_200602 values less than (200603),
...
partition PART_200711 values less than (200712),
partition PART_200712 values less than (200801),
...
partition PART_MAX    values less than (maxvalue)
)
tablespace USERS
nologging parallel
enable row movement
as
select * from W_REVN_F_ORIG
;
```

4. Drop all indexes on the renamed table.

   To ensure index naming standards in the target schema, drop all indexes on the renamed table (W_REVN_F_ORIG table in the example).

5. Create a global unique index.

   Create a global unique B-Tree index using the information from item Step 2, for example:

```
create unique index W_REVN_F_P1
on W_REVN_F (ROW_WID)
tablespace USERS_IDX global nologging parallel;
```

6. Create local indexes.

   You can generate the set of required 'Create index' DDL commands for Bitmap indexes using the DAC repository, for example:

   ```
   select 'CREATE BITMAP INDEX '|| i.name || ' ON W_REVN_F ('|| c.name || ' ASC)
   tablespace USERS_IDX LOCAL NOLOGGING PARALLEL;'
   from W_ETL_TABLE T, W_ETL_INDEX I, W_ETL_INDEX_COL C
   where T.ROW_WID = I.TABLE_WID
   and T.NAME='W_REVN_F'
   and I.ROW_WID = C.INDEX_WID
   and I.ISBITMAP = 'Y'
   and I.INACTIVE_FLG = 'N';
   ```

   If there are any local B-Tree indexes identified in Step 2, create them as local partitioned indexes as well. You can modify the query above and add I.ISBITMAP = 'N'. Connect to the Target schema and execute the generated DDLs to create the indexes.

7. Compute statistics.

   Compute statistics on the newly partitioned table and its indexes using the DBMS_STATS API, for example:

   ```
   SQL> BEGIN
   DBMS_STATS.GATHER_TABLE_STATS( NULL
   , tabname=>'W_REVN_F'
   , cascade => TRUE
   , estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE
   , method_opt => 'FOR ALL INDEXED COLUMNS SIZE AUTO'
   );
   END;
   ```

8. Enable row movement.

   Since the custom XML scripts will make local indexes unusable during incremental runs, you need to set the parameter skip_unusable_indexes to true in the session running ETL. Connect to your Informatica Workflow Manager, navigate to Connections, then Relational, and edit Data Warehouse connection. Add the following SQL statement in the field Environment SQL:

   ```
   alter session set skip_unusable_indexes=true;
   ```

9. Drop the original table.

   After you confirm performance gains with the partitioned table and indexes in your target database, you can consider dropping the original renamed table (_ORIG).

You now need to configure the DAC to support ETL for partitioned tables (for more information, see Section 3.9.3, "Configure DAC To Support ETL For Partitioned Tables").

3.9.3 Configure DAC To Support ETL For Partitioned Tables

Once you have partitioned your fact tables as described in Section 3.9.2, "How to Partition Large Fact Tables", you need to configure the DAC console to support the partitioned tables, as follows:

1. Inactivate all indexes on a partitioned target table, as follows:
   a. In the DAC Client, display the Design View.

      For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
   b. Display the Indices tab in the upper right pane.
   c. Click the Query button, then enter the table name, then click Go.
   d. Select the **Inactive** check box for all indexes on the target table.

      These indexes will be handled outside DAC through pre-SQL and post-SQL at a task level.

      Note: The DAC version released with Oracle Business Intelligence Applications Version 7.9.5 does not disable computing statistics at a table level. Since computing statistics for a partitioned table will be handled by a post-SQL XML file, Analyze Table Task in DAC for the table will be redundant. To workaround this limitation, you can abort the execution plan in DAC, mark the task Analyze Table for your staging table as Completed and restart the Execution Plan.

      ---

2. Create two new source system parameters In the DAC Client, as follows:
   a. In the DAC Client, display the Design View.

      For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
   b. Select your custom container from the pull-down menu (next to the Execute Button).

      Note: You cannot create any custom parameters in the pre-packaged containers. You must first create a new container. Refer to the *Oracle Business Intelligence Applications Database Administration Console Guide* for information to create a Custom Container as a copy of a packaged one.

      ---

   c. Display the Source System Parameters tab in the upper right pane.
   d. Use the New button to create two new source system parameters with the following values:

Table 3–11 Source system parameters for partitioned tables

Name	Data Type	Value
CURRENT_ MONTH_WID	SQL	select to_char(row_wid) from w_month_d where w_current_cal_month_code = 'Current' When you click on the Value field to display the Enter Parameter Value dialog, select DBConnection_OLAP from the **Logical Data Source** drop down list.

Table 3–11 (Cont.) Source system parameters for partitioned tables

Name	Data Type	Value
PREVIOUS_ MONTH_WID	SQL	select to_char(row_wid) from w_month_d where w_current_cal_month_code = 'Previous' When you click on the Value field to display the Enter Parameter Value dialog, select DBConnection_OLAP from the **Logical Data Source** drop down list.

These source system parameters will be used in pre-SQL and post-SQL xml files during incremental loads.

3. Prepare pre-SQL xml files and post-SQL xml files for full and incremental loads.

   You can either obtain the template file samplesql.xml, located under <DAC_HOME>\CustomSQLs directory and generate the required pre- and post- SQL xml files, or use the XML files examples in the next section for your custom XML files. DAC will execute the SQL from pre-SQL XMLs before Informatica tasks, and then run the SQL from post-SQL XML files.

   Create the following set of custom xml files, one set per partitioned table:

   - pre_<TABLE_NAME>_ini.xml - drops all local and global indexes.
   - post_<TABLE_NAME>_ini.xml - creates all local and global indexes.
   - pre_<TABLE_NAME>incr.xml - makes local indexes unusable for last two partitions 'PART@DAC_CURRENT_MONTH_WID' and 'PART_@DAC_PREVIOUS_MONTH_WID'.
   - post_<TABLE_NAME>incr.xml - rebuilds the unusable local indexes and computes statistics on the two partitions, the rebuilt local and global indexes.

     Important: Make sure that your partition names include the values from the DAC system variables CURRENT_MONTH_WID and PREVIOUS_MONTH_WID. These variables must be used in the custom XML files to handle local and global indexes using syntax: 'partition PART_@DAC_CURRENT_MONTH_WID' and 'partition PART_@DAC_PREVIOUS_MONTH_WID'. Note that the DDL syntax must have the prefix 'DAC_' in the custom XML files.

4. Copy all pre- and post-SQL xml files under <DAC_HOME>\CustomSQLs location.

5. Associate the four generated custom XML files with a task, which loads data into the target partitioned table.

   a. In the DAC Client, display the Design View.

      For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

   b. Select your custom container from the pull-down menu (next to the Execute Button).

   c. Click the Tables menu in the upper right pane, and find or query the target partitioned Fact table.

   d. Click the menu Target for Tasks (RO) in the lower right pane.

e. Find the SIL primary mapping name. For example the primary mapping for W_GL_REVN_F fact table is SIL_GLRevenueFact. Copy or save the mapping name.

f. Click the menu Tasks in the upper right pane.

g. Find or query the identified mapping name from the preceding step.

h. Fill in the custom XML file names in the following fields on the bottom right pane and save the change:

Table 3–12 Field Values on Tasks Tab

Field Name on Tasks Tab	Value	Description
Pre-SQL for Full Load	pre_<TABLE_ NAME>_ini.xml	Pre SQL xml file name for Full Load. For an example XML file, see Section 3.9.4.1, "Pre_W_ REVN_F_ini.xml - Pre SQL xml file name for Full Load".
Post-SQL for Full Load	post_<TABLE_ NAME>_ini.xml	Post SQL xml file name for Full Load. For an example XML file, see Section 3.9.4.2, "Post_W_ REVN_F_ini.xml - Post SQL xml file name for Full Load".
Pre-SQL for Incremental Load	pre_<TABLE_ NAME>incr.xml	Pre SQL xml file name for Incremental Load. For an example XML file, see Section 3.9.4.3, "Pre_W_REVN_F_incr.xml- Pre SQL xml file name for Incremental Load".
Post-SQL for Incremental Load	post_<TABLE_ NAME>incr.xml	Post SQL xml file name for Incremental Load. For an example XML file, see Section 3.9.4.4, "Post_W_REVN_F_incr.xml- Post SQL xml file name for Incremental Load".

6. Save the changes in DAC.

7. Select the checkbox Analyze Truncate Table Only under the Execution Plan tab in the DAC.

   a. Display the Execute view.

   b. Click the tab Execution Plans in the upper right pane.

   c. Find your Execution Plan.

   d. Check the checkbox Analyze Truncated Tables only and save the changes.

      The DAC will compute statistics on tables that were truncated prior to an ETL and reloaded during the plan execution.

   e. Rebuild the Execution Plan in DAC.

      Click the tab Build in the upper right pane to rebuild the execution plan. When the Build process is complete, the ETL is ready to run.

3.9.4 Example XML Files

This section contains example XML files for pre-SQL and post-SQL for use in Step 5 of the task Section 3.9.3, "Configure DAC To Support ETL For Partitioned Tables", and contains the following topics:

- Section 3.9.4.1, "Pre_W_REVN_F_ini.xml - Pre SQL xml file name for Full Load"
- Section 3.9.4.2, "Post_W_REVN_F_ini.xml - Post SQL xml file name for Full Load"

- Section 3.9.4.3, "Pre_W_REVN_F_incr.xml- Pre SQL xml file name for Incremental Load"
- Section 3.9.4.4, "Post_W_REVN_F_incr.xml- Post SQL xml file name for Incremental Load"

3.9.4.1 Pre_W_REVN_F_ini.xml - Pre SQL xml file name for Full Load

```
<?xml version="1.0" ?>
<CustomSQLs>
   <!--  Start Drop Index -->
   <!--  Drop W_REVN_F_F1 -->
   <sql name="Drop Index W_REVN_F_F1" type="SQL" continueOnFail="true"
validDBPlatforms="Oracle">
      <![CDATA[ drop index W_REVN_F_F1 ]]>
   </sql>
   <!--  Drop W_REVN_F_F2 -->
      <sql name="Drop Index W_REVN_F_F2" type="SQL" continueOnFail="true"
validDBPlatforms="Oracle">
      <![CDATA[ drop index W_REVN_F_F2 ]]>
   </sql>
...
...
...
   <!--  Drop W_REVN_F_U1 -->
      <sql name="Drop Index W_REVN_F_U1" type="SQL" continueOnFail="true"
validDBPlatforms="Oracle">
      <![CDATA[ drop index W_REVN_F_U1 ]]>
   </sql>
</CustomSQLs>
```

3.9.4.2 Post_W_REVN_F_ini.xml - Post SQL xml file name for Full Load

```
 <?xml version="1.0" ?>
<CustomSQLs>
   <!--  Start Create Index -->
   <!--  Create global index W_REVN_F_U1 -->
      <sql name="Create Index W_REVN_F_P1" type="SQL" continueOnFail="false"
validDBPlatforms="Oracle">
      <![CDATA[  create unique index W_REVN_F_P1 on W_REVN_F(ROW_WID) tablespace
USERS_IDX GLOBAL NOLOGGING PARALLEL ]]>
   </sql>
   <!--  Create local index W_REVN_F_F1 -->
      <sql name="Create Index W_REVN_F_F1" type="SQL" continueOnFail="false"
validDBPlatforms="Oracle">
      <![CDATA[ CREATE BITMAP INDEX W_REVN_F_F1 ON W_REVN_F(COLUMN_NAME1 ASC)
TABLEPSACE USERS_IDX LOCAL NOLOGGING PARALLEL ]]>
   </sql>
   <!--  Create local index W_REVN_F_F2 -->
      <sql name="Create Index W_REVN_F_F2" type="SQL" continueOnFail="false"
validDBPlatforms="Oracle">
      <![CDATA[ CREATE BITMAP INDEX W_REVN_F_F2 ON W_REVN_F(COLUMN_NAME2 ASC)
TABLEPSACE USERS_IDX LOCAL NOLOGGING PARALLEL ]]>
   </sql>
...
...
...
</CustomSQLs>
```

3.9.4.3 Pre_W_REVN_F_incr.xml- Pre SQL xml file name for Incremental Load

```
<?xml version="1.0" ?>
<CustomSQLs>
   <!--  Start W_REVN_F_F1 -->
   <sql name="W_REVN_F_F1 Disable Previous" type="SQL" continueOnFail="true"
validDBPlatforms="Oracle">
      <![CDATA[ alter index W_REVN_F_F1 modify partition PART_@DAC_PREVIOUS_MONTH_
WID unusable ]]>
   </sql>
   <sql name="W_REVN_F_F1 Disable Current" type="SQL" continueOnFail="true"
validDBPlatforms="Oracle">
      <![CDATA[ alter index W_REVN_F_F1 modify partition PART_@DAC_CURRENT_MONTH_
WID unusable ]]>
   </sql>
   <!--  Start W_REVN_F_F2 -->
   <sql name="W_REVN_F_F2 Disable Previous" type="SQL" continueOnFail="true"
validDBPlatforms="Oracle">
      <![CDATA[ alter index W_REVN_F_F2 modify partition PART_@DAC_PREVIOUS_MONTH_
WID unusable ]]>
   </sql>
   <sql name="W_REVN_F_F2 Disable Current" type="SQL" continueOnFail="true"
validDBPlatforms="Oracle">
      <![CDATA[ alter index W_REVN_F_F2 modify partition PART_@DAC_CURRENT_MONTH_
WID unusable ]]>
   </sql>
...
...
...
</CustomSQLs>
```

3.9.4.4 Post_W_REVN_F_incr.xml- Post SQL xml file name for Incremental Load

```
<?xml version="1.0" ?>
<CustomSQLs>
   <!--  Start W_REVN_F PART_PREVIOUS_MONTH_WID -->
   <sql name="W_REVN_F Rebuild Previous" type="SQL" continueOnFail="false"
validDBPlatforms="Oracle">
      <![CDATA[ alter table W_REVN_F modify partition PART_@DAC_PREVIOUS_MONTH_WID
rebuild unusable local indexes]]>
   </sql>
   <!--  Start W_REVN_F PART_CURRENT_MONTH_WID -->
   <sql name=" W_REVN_F Rebuild Current " type="SQL" continueOnFail="false"
validDBPlatforms="Oracle">
      <![CDATA[alter table W_REVN_F modify partition PART_@DAC_CURRENT_MONTH_WID
rebuild unusable local indexes]]>
   </sql>
   <!--  Gather Stats on global indexes -->
   <sql name="Gather Stats on W_REVN_F_U1" type="Stored Procedure"
continueOnFail="false" validDBPlatforms="Oracle">
      <![CDATA[
               DBMS_STATS.GATHER_INDEX_STATS( NULL
               , indname => 'W_REVN_F_P1    '
               , estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE
               , degree => DBMS_STATS.DEFAULT_DEGREE
               )
      ]]>
   </sql>
   <!--  Gather Stats on W_REVN_F table -->
   <sql name="W_REVN_F Gather Stats Previous" type="Stored Procedure"
continueOnFail="false" validDBPlatforms="Oracle">
```

```
      <![CDATA[
               DBMS_STATS.GATHER_TABLE_STATS( NULL
               , tabname=>'W_REVN_F'
               , cascade => TRUE
               , partname => 'PART_@DAC_PREVIOUS_MONTH_WID'
               , estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE
               , granularity => 'PARTITION'
               , method_opt => 'FOR ALL INDEXED COLUMNS SIZE AUTO'
               , degree => DBMS_STATS.DEFAULT_DEGREE
               )
      ]]>
   </sql>
   <sql name="W_REVN_F Gather Stats Current" type="Stored Procedure"
continueOnFail="false" validDBPlatforms="Oracle">
      <![CDATA[
               DBMS_STATS.GATHER_TABLE_STATS( NULL
               , tabname=>'W_REVN_F'
               , cascade => TRUE
               , partname => 'PART_@DAC_CURRENT_MONTH_WID'
               , estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE
               , granularity => 'PARTITION'
               , method_opt => 'FOR ALL INDEXED COLUMNS SIZE AUTO'
               , degree => DBMS_STATS.DEFAULT_DEGREE
               )
      ]]>
   </sql>
</CustomSQLs>
```

4

Installing and Setting Up Oracle Business Intelligence Applications

This chapter explains how to install and set up the Oracle Business Intelligence Applications components. It contains the following main topics:

- Section 4.1, "About Oracle Business Intelligence Applications Topologies"
- Section 4.2, "Installation and Configuration Process Task List"
- Section 4.3, "Mandatory Requirements and Pre-installation Tasks"
- Section 4.4, "Installing Oracle Business Intelligence Applications Software"
- Section 4.5, "Installing Informatica PowerCenter Services Software"
- Section 4.7, "Configuring Informatica PowerCenter Components"
- Section 4.8, "Installing and Configuring the DAC Client"
- Section 4.9, "Importing the DAC Metadata"
- Section 4.10, "Creating the Oracle Business Analytics Warehouse Database Tables"
- Section 4.11, "How to Install and Configure the DAC Server"
- Section 4.12, "Configuring the DAC Settings"
- Section 4.13, "Configuring Relational Connections In Informatica Workflow Manager"
- Section 4.15, "Miscellaneous and Additional Configuration Tasks"
- Section 4.16, "About Running A Full Load ETL"

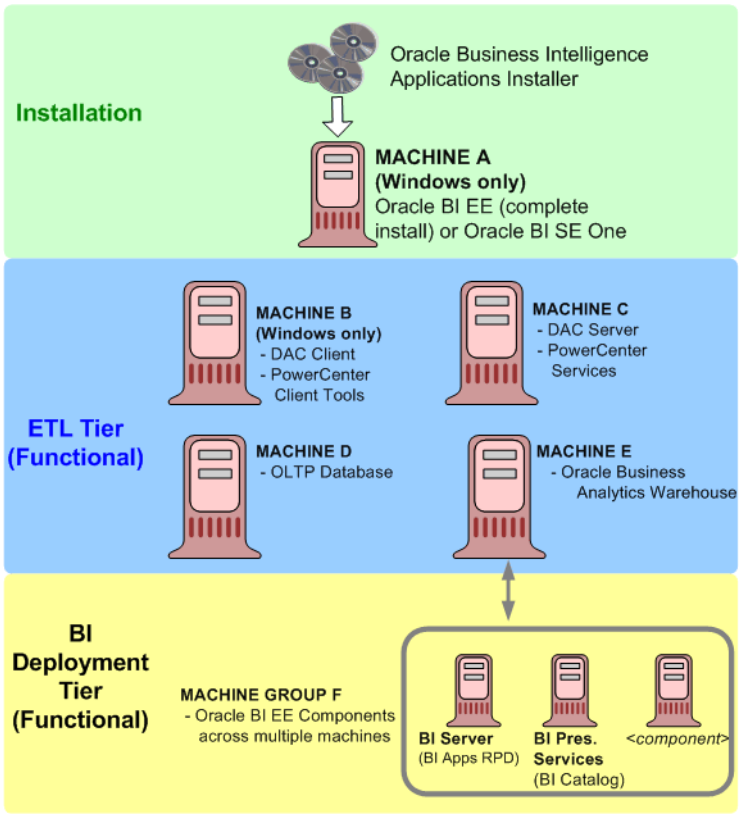
For information about supported Windows versions, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications Fusion Edition*.

4.1 About Oracle Business Intelligence Applications Topologies

Oracle Business Intelligence Applications and Informatica PowerCenter can be deployed flexibly across a wide range of topologies on different platforms and combinations of platform.

This section uses an example to explain a typical topology for an Oracle Business Intelligence Applications deployment. The figure below shows a typical deployment topology.

Figure 4–1 A typical topology for an Oracle Business Intelligence Applications deployment



In the figure above, note the following:

- **Installation**
  - Machine A (Windows-only)

    Machine A is a machine that has installed Oracle Business Intelligence Enterprise Edition or Oracle Business Intelligence Standard Edition One, on which you run the Oracle Business Intelligence Applications installer to install the Oracle Business Intelligence Applications files.

    Note: The instance of Oracle Business Intelligence Enterprise Edition or Oracle Business Intelligence Standard Edition One does not need to be the functional version of Oracle Business Intelligence Enterprise Edition that you will use to deploy dashboards in your live system. This instance is only required to enable the Oracle Business Intelligence Applications installer to install the Oracle Business Intelligence Applications files on a machine.

    After the Oracle Business Intelligence Applications files have been installed on Machine A, the DAC Client is deployed to Machine B, and the DAC Server is deployed to Machine C.

    In addition, the following files are copied from the installation machine (Machine A) to the Business Intelligence Deployment Tier (Machine Group G) as follows:

    - The OracleBI\Server\Repository\EnterpriseBusinessAnalytics.rpd file is copied from Machine A to the machine that runs the BI Server in Machine Group F, or to Machine G.

- The OracleBIData\Web\Catalog\EnterpriseBusinessAnalytics*.* files are copied from Machine A to the machine that runs the BI Presentation Services Catalog in Machine Group F, or to Machine G.

  You typically develop the RPD and Presentation Catalog and perform customization changes to fit your business requirements.

- **ETL Tier (Functional)**
  - Machine B (Windows-only)

    Machine B is a machine that runs the DAC Client and Informatica PowerCenter Client Tools.
  - Machine C (Windows, Unix, Linux)

    Machine C is a machine that runs the DAC Server and Informatica PowerCenter Services.
  - Machine D (Windows, Unix, Linux)

    Machine D is a machine that hosts the transactional (OLTP) database.
  - Machine E (Windows, Unix, Linux)

    Machine E is a machine that hosts the Oracle Business Analytics Warehouse database.
- **BI Deployment Tier (Functional)**

  The BI Deployment Tier (Functional) tier is used to deploy the business intelligence dashboards, can have either of the following:
  - Machine Group F (Windows, Unix, Linux)

    Machine Group F is a group of machines that runs the Oracle Business Intelligence Enterprise Edition components. For example, one machine might rung the BI Server and BI Applications RPD, and another machine might run the BI Presentation Services and the Oracle Business Analytics Warehouse.

    Note: You can also run Oracle Business Intelligence Applications on a single machine installation of Oracle Business Intelligence Standard Edition One.

4.2 Installation and Configuration Process Task List

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

1. Before you start to install Oracle Business Intelligence Applications and Informatica PowerCenter Services, make sure that you have satisfied the requirements, as follows:
   a. Decide on the Data Movement Type that you will need (for more information, see Section 4.3.3, "Code Page and Data Movement Requirements").
   b. Create the required databases to store the Informatica repository, DAC repository, and Oracle Business Analytics Warehouse. In addition, install database connectivity software and create of ODBC connections. For more information, see Section 4.3.4.1, "Create Databases".
   c. Make sure that a complete installation of Oracle Business Intelligence Enterprise Edition or Oracle Business Intelligence Standard Edition One is installed on the machine on which you want to install Oracle Business

Intelligence Applications (for more information, see Section 4.3.1, "Oracle Business Intelligence Infrastructure Requirements").

Note: This instance does not need to be the functional version of Oracle Business Intelligence Enterprise Edition or Oracle Business Intelligence Standard Edition One that you will use to deploy dashboards in your live system. This instance is only required to enable the Oracle Business Intelligence Applications installer to install the Oracle Business Intelligence Applications files on a machine.

d. Read the Informatica PowerCenter overview section and plan your Informatica topology before you install Informatica PowerCenter (for more information, see Section 4.3.2, "Informatica PowerCenter Requirements").

e. Make sure that you have installed Java JDK on the DAC Client and DAC Server machine (for more information, see Section 4.3.4.5, "How to Install the Java JDK").

2. Install the Oracle Business Intelligence Applications Version 7.9.5 software (for more information, see Section 4.4, "Installing Oracle Business Intelligence Applications Software"), as follows:

   a. If you do not have Java SDK installed, install Java SDK on the DAC Client and DAC Server machines. For more information, see Section 4.3.4.5, "How to Install the Java JDK".

   b. Run the Oracle installer to install the Oracle Business Intelligence Applications software. For more information, see Section 4.4.1, "How to Run the Oracle Business Intelligence Applications Installer (Windows)".

3. Install Informatica PowerCenter Server and Client Tools Version 8.1.1 SP4 software (for more information, see Section 4.5, "Installing Informatica PowerCenter Services Software"), as follows:

   a. Run the Informatica PowerCenter installer to install Informatica PowerCenter Client Tools (including PowerCenter Workflow Manager, PowerCenter Designer, PowerCenter). For more information, see Section 4.5.1, "How to Install Informatica PowerCenter Client Tools".

   b. Run the Informatica PowerCenter installer to install Informatica PowerCenter Services. For more information, see Section 4.5.2, "How to Install Informatica PowerCenter Services".

4. Restore the pre-built repository file (for more information, see Section 4.6, "Restoring the Oracle Business Intelligence Pre-Built Repository").

5. Configure the Informatica PowerCenter components (for more information, see Section 4.7, "Configuring Informatica PowerCenter Components"), as follows:

   a. Use the Informatica PowerCenter Administration Console to specify the server port number. For more information, see Section 4.7.3, "Setting PowerCenter Integration Services Custom Properties".

   b. Configure the required relational connections. For more information, see Section 4.13.2, "Configuring Relational Connections in Informatica PowerCenter Workflow Manager".

6. Install and Configure the DAC Client. For more information, see Section 4.8, "Installing and Configuring the DAC Client".

7. Use the DAC client to create the DAC metadata repository (for more information, see Section 4.9, "Importing the DAC Metadata").

8. Create the Oracle Business Analytics Warehouse tables. For more information, see Section 4.10, "Creating the Oracle Business Analytics Warehouse Database Tables".
9. Install and configure the DAC server. For more information, see Section 4.11, "How to Install and Configure the DAC Server".
10. Configure DAC Integration Settings. For more information, see Section 4.12, "Configuring the DAC Settings".
11. Configure Informatica Workflow Manager. For more information, see Section 4.13, "Configuring Relational Connections In Informatica Workflow Manager".
12. Perform required post-installation tasks (for more information, see Section 4.15, "Miscellaneous and Additional Configuration Tasks").

    For additional and supporting configuration tasks, see Section 4.15, "Miscellaneous and Additional Configuration Tasks".

> **Note:** After you have completed the above steps and completed any customization that you need to make, you are ready to perform a full load of your OLTP data. For an example of performing a full load of OLTP data, see Section 4.16, "About Running A Full Load ETL". For detailed information about using the DAC to perform ETL, see *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

4.3 Mandatory Requirements and Pre-installation Tasks

This section includes mandatory requirements that you must satisfy and pre-installation tasks that you must perform before you can deploy Oracle Business Intelligence Applications, and contains the following topics:

- Section 4.3.1, "Oracle Business Intelligence Infrastructure Requirements"
- Section 4.3.2, "Informatica PowerCenter Requirements"
- Section 4.3.3, "Code Page and Data Movement Requirements"
- Section 4.3.4, "Pre-installation Tasks"

4.3.1 Oracle Business Intelligence Infrastructure Requirements

Oracle Business Intelligence Applications has the following infrastructure requirements:

- For installing Oracle Business Intelligence Applications, you must have installed Oracle Business Intelligence infrastructure on a Windows machine where you will run the Oracle Business Intelligence Applications installer. The Oracle Business Intelligence Infrastructure instance must be a complete installation of Oracle Business Intelligence Enterprise Edition or Oracle Business Intelligence Standard Edition One. If you install Oracle Business Intelligence Enterprise Edition, you must install it using use the Complete setup type option.

  Note: This instance does not need to be the functional version of Oracle Business Intelligence Enterprise Edition that you will use to deploy reports and dashboards in your live system. This instance is only required to enable the Oracle Business Intelligence Applications installer to install the Oracle Business Intelligence Applications files on a machine.
- For running Oracle Business Intelligence Applications, you must have installed an Oracle Business Intelligence infrastructure. The Oracle Business Intelligence

infrastructure required to run Oracle Business Intelligence Applications may be either Oracle Business Intelligence Enterprise Edition or Oracle Business Intelligence Standard Edition One. The Oracle Business Intelligence infrastructure uses the Oracle Business Analytics Warehouse as a datasource, the prebuilt EnterpriseBusinessAnalytics.rpd as the repository for the Oracle BI Server, and provides users with the Oracle BI Applications reports and dashboards.

To determine the minimum version of Oracle Business Intelligence infrastructure that is supported for this release of Oracle Business Intelligence Applications, refer to the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications Fusion Edition* for Oracle Business Intelligence Applications.

4.3.2 Informatica PowerCenter Requirements

This section contains requirements information for Informatica PowerCenter.

Oracle Business Intelligence Applications Version 7.9.5 requires Informatica PowerCenter Version 8.1.1. SP4.

If you license Informatica PowerCenter with Oracle Business Intelligence Applications, you install Informatica PowerCenter Version 8.1.1 SP4 from the Oracle Business Intelligence Applications installation disk (for more information, see Section 4.5, "Installing Informatica PowerCenter Services Software").

If you license Informatica PowerCenter separately and you do not have Informatica PowerCenter Version 8.1.1 SP4, you must upgrade your Informatica PowerCenter license to Version 8.1.1 SP4 before deploying Oracle Business Intelligence Applications.

Informatica PowerCenter 8.1.1 has significant architecture changes from previous versions. Before installing Oracle Business Intelligence Applications, Oracle recommends that you do the following:

- Read the Informatica PowerCenter documentation to familiarize yourself with the new architecture, components and features. Informatica PowerCenter 8.1.1 SP 4 documentation is included on the Informatica PowerCenter DVD provided with Oracle Business Intelligence Applications.
- Plan your topology carefully before installing Informatica PowerCenter Services.

You must install Informatica PowerCenter Client Tools on the same machine as the DAC Client, and to install Informatica PowerCenter Services on the same machine as the DAC Server.

Informatica PowerCenter Services runs on 32-bit or 64-bit platforms. For more information see the topic titled 'Working with 32-bit and 64-bit Platforms' in the chapter on PowerCenter Pre-requisites in the *PowerCenter Installation and Configuration Guide*.

About Changes In Informatica Terminology

This section lists changes in terminology in Informatica PowerCenter 8.1.1 SP4.

Table 4–1 Changes in Informatica Terminology

Term used in 7.1.x versions of Informatica PowerCenter	Term used in 8.1.1 version of Informatica PowerCenter
Informatica Repository	Informatica Repository The Informatica Repository is managed by the Repository Service.

Table 4–1 (Cont.) Changes in Informatica Terminology

Term used in 7.1.x versions of Informatica PowerCenter	Term used in 8.1.1 version of Informatica PowerCenter
Informatica Repository Server	Not applicable. This component does not exist in Informatica PowerCenter 8.1.1. The Informatica Repository is managed by the Repository Service.
Informatica Server	Integration Services
Not applicable.	Repository Service

4.3.3 Code Page and Data Movement Requirements

The Oracle Business Analytics Warehouse can be deployed in various code page environments and supports global deployments. Data movement in the following source database and data warehouse configuration modes are supported:

- Unicode to Unicode
- Code page (multi- or single-byte) to Unicode
- Code page to Code page (where the code pages are the same)

Oracle Business Intelligence Applications uses Informatica PowerCenter to perform extract-transform and load routines to move data from source database(s) to the Oracle Business Analytics Warehouse.

During the installation and configuration procedures described in this chapter, you will make various settings to enable accurate data movement. Use the guidelines and references noted below to determine values for these settings that are appropriate for your environment:

- Determining the source to target configuration mode. Consult your database administrator to determine the code page your source OLTP database uses. Based on the type of data that will be moved from one or more source databases to the Oracle Business Analytics Warehouse, determine what code page you will need to use for the Oracle Business Analytics Warehouse database. Consider future requirements for storing data when determining what code page to use for the Oracle Business Analytics Warehouse.

  For accurate data movement from source database to target, the code page of the Oracle Business Analytics Warehouse (target) must be superset of the code page of the source database. Informatica considers a code page to be a superset of another code page when the code page contains all the characters encoded in the other code page and additional characters not encoded in the other code page.

  If your source to target configuration mode for data movement is Unicode to Unicode, you will set a custom property called SiebelUnicodeDB on the Integration Services. Configuration modes of code page to Unicode or code page to code page do not require this property to be set.

  Note: To enable data movement from source(s) to the Oracle Business Analytics Warehouse, you will set relaxed Code Page Validation for the Integration Services. You must ensure that the target code page is a superset of the source code page for accurate data movement.

- Determining the Data Movement Mode. Before you deploy Oracle Business Intelligence Applications, you must determine what data movement mode to use (ASCII or Unicode) for the PowerCenter Integration Service. The Character Data

Movement Mode is an Informatica PowerCenter Integration Service option that you choose based on whether you want to move single-byte or multi-byte data. Choose Unicode if non-ASCII characters have to be moved. Otherwise, choose ASCII. The Data Movement Mode option is configurable and can be reset after installation.

- Determining code pages for Informatica PowerCenter components. In addition to source and target code pages, Informatica PowerCenter uses code pages for PowerCenter Client, the Integration Service, the Informatica Repository, and PowerCenter command line programs (pmcmd and pmrep, which are used by DAC to communicate with PowerCenter Services). Carefully review, particularly if your environment requires Data Mode Movement mode to be set to UNICODE, the Informatica documentation on 'Understanding Globalization' in the PowerCenter Administration Guide. The section discusses code page compatibility and code page requirements for the Informatica components.
- Setting environment variables. The Informatica installer requires the appropriate locale to be set on UNIX machines. Use LANG, LC_CTYPE or LC_ALL to set the UNIX code page. The installer uses the locale environment variable to generate the list of compatible code pages. For more information, see the topic titled 'Locale Environment Variables' in the PowerCenter Installation and Configuration Guide.

  If your environment uses Oracle or DB2 database, you need to set environment variables NLS_LANG or DB2CODEPAGE. For information on how to set these environment variables see Section 4.3.4.2.1, "Setting the NLS_LANG Environment Variable for Oracle Databases" and Section 4.3.4.2.2, "Setting the DB2CODEPAGE Environment Variable for DB2 Databases".
- Configuring Relational Connections. When you configure relational connections in the Workflow Manager, choose a code page that is compatible with the code page of the database client. If you set a database environment variable to specify the language for the database, ensure the code page for the connection is compatible with the language set for the variable. For example, if you set the NLS_LANG environment variable for an Oracle database, ensure that the code page of the Oracle connection is identical to the value set in the NLS_LANG variable.

For more information about data movement modes, refer to the Informatica PowerCenter documentation. Informatica PowerCenter 8.1.1 SP 4 documentation is provided in the Third_Party_Software\Informatica folder of the Oracle Business Intelligence Applications Release 7.9.5 installation disk.

4.3.4 Pre-installation Tasks

This section explains the mandatory tasks that you must perform for an Oracle Business Intelligence Applications deployment, and contains the following topics:

- Section 4.3.4.1, "Create Databases"
- Section 4.3.4.2, "Install Database Connectivity Software"
- Section 4.3.4.3, "Create ODBC Database Connections"
- Section 4.3.4.4, "Set Environment Variables"
- Section 4.3.4.5, "How to Install the Java JDK"
- Section 4.3.4.6, "Download Hibernate Libraries"

4.3.4.1 Create Databases

This section provides information on creating databases required by Oracle Business Intelligence Applications components and Informatica PowerCenter.

Before you install Oracle Business Intelligence Applications and Informatica PowerCenter, use your target database tool to create database instances to hold the following:

- the DAC metadata repository.
- the Informatica repository.
- the PowerCenter domain configuration.
- the Oracle Business Analytics Warehouse.

  Note: Make sure that the Oracle Business Analytics Warehouse instance is granted the SSE_ROLE (for more information, see Section 4.3.4.1.1, "How to Create the SSE Role").

You can store both the DAC metadata repository and the Oracle Business Analytics Warehouse in separate databases, or in the same database.

For information about supported database platforms, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications Fusion Edition*. The Informatica and DAC repositories are not supported on DB2-390 or Teradata databases. If your source or Oracle Business Analytics Warehouse database is DB2-390, or your Oracle Business Analytics Warehouse database is Teradata, you need to use a DB2-UDB, MS SQL Server, or Oracle database for both the DAC repository and the Informatica repository.

For details on creating the Informatica repository and PowerCenter domain configuration databases, see the section on creating databases in the PowerCenter Installation and Configuration Guide. Informatica PowerCenter 8.1.1 SP 4 documentation is included on the Informatica PowerCenter DVD provided with Oracle Business Intelligence Applications.

4.3.4.1.1 How to Create the SSE Role

Follow this procedure to create the SSE role for the Oracle Business Analytics Warehouse database.

To create the SSE Role

1. 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\createwtables.bat file.

4.3.4.2 Install Database Connectivity Software

You must install and configure the appropriate database native connectivity client software on the machines that will host the Informatica PowerCenter Services and DAC Server, and DAC Client.

The machines that will host these components require connectivity to the Oracle Business Analytics Warehouse (target) database, transactional (source) database(s), and the DAC and Informatica repository databases.

Notes

- For instructions on how to install and configure the native database client connectivity software, refer to the PowerCenter Installation and Configuration Guide:
- To install and configure on the Windows machine that will host the DAC Client and PowerCenter Tools, see section 'Connecting to Databases from Windows' in the Informatica Installation and Configuration Guide. Also refer to this section if the DAC Server and PowerCenter Services will be installed on Windows.
- To install and configure on a UNIX or Linux machine that will host the DAC Server and PowerCenter Services, see section 'Connecting to Databases from UNIX'.
- Informatica PowerCenter 8.1.1 SP 4 documentation is included on the Informatica PowerCenter DVD provided with Oracle Business Intelligence Applications.
- For Oracle databases, set the NLS_LANG environment variable on the machines where the Oracle client was installed, as described in Section 4.3.4.2.1, "Setting the NLS_LANG Environment Variable for Oracle Databases".
- For DB2 databases, set the DB2CODEPAGE environment variable on the machines where the DB2 client was installed, as described in Section 4.3.4.2.2, "Setting the DB2CODEPAGE Environment Variable for DB2 Databases".

4.3.4.2.1 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 each machine that has the Oracle client installed.

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 and Linux, 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 Integration Service is running on UNIX or Linux, 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.

4.3.4.2.2 Setting the DB2CODEPAGE Environment Variable for DB2 Databases

Follow this procedure to set the DB2CODEPAGE environment variable for DB2 databases on machines that have the DB2 client installed.

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 or Linux, set the variable as shown below:

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

   For example: `setenv 1208`.

4.3.4.3 Create ODBC Database Connections

The DAC Client uses an ODBC connection to create and alter tables in the Oracle Business Analytics Warehouse. Create an ODBC connection to the Oracle Business Analytics Warehouse database on the machine that hosts the DAC Client, as described below.

Additionally, if your source system is Siebel CRM, you must create an ODBC connection to the OLTP (source) database on the machine where you will install Oracle Business Intelligence Applications software. This ODBC connection will be used by the ddlimp utility.

For the ODBC connections to - Oracle databases, you must use the Oracle Merant ODBC Driver that is installed with the Oracle Business Intelligence Applications installer. For all other databases, you should use ODBC drivers supplied by your database vendor.

To create ODBC connections for DB2 databases:

1. Using the DB2 Client Configuration Assistant, create a database connection to the Oracle Business Analytics Warehouse database on the machine that will host the DAC Client. If your source system is Siebel CRM, also create an ODBC connection to the transactional database on the machine where you will run the Oracle Business Intelligence Applications installer.

   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, on Windows, in the System DSN tab of the ODBC Data Source Administrator, create an ODBC connection to the Oracle Business Analytics Warehouse (and, if your source system is Siebel CRM, to the transactional database) using an ODBC driver.
3. Test the connections to make sure they work.

To create ODBC connections for Oracle databases:

Note: You must use the Oracle Merant ODBC driver to create the ODBC connections. The Oracle Merant ODBC driver is installed by the Oracle Business Intelligence Applications installer. Therefore, you will need to create the ODBC connections after you have run the Oracle Business Intelligence Applications installer and have installed the DAC Client.

1. On the machine that will host the DAC Client, use the System DSN tab of the ODBC Data Source Administrator to create an ODBC connection to the Oracle Business Analytics Warehouse database using the Oracle Merant ODBC driver that is supplied with Oracle Business Intelligence Applications.

   For example, you might create a database connection called Connect_to_OLAP.

2. (If your source system is Siebel CRM) In Windows, in the System DSN tab of the ODBC Data Source Administrator, create an ODBC connection to the Siebel transactional database using the Oracle Merant Closed driver.

3. Test the connections to make sure they work.

To create ODBC connections for SQL Server databases:

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 database on the machine that will host the DAC Client.
   - (If your source system is Siebel CRM) an ODBC connection to the transactional database on the machine where you will run the Oracle Business Intelligence Applications installer

   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 ODBC connections for Teradata databases:

1. On the Windows machine that will host the DAC Client, in the System DSN tab of the ODBC Data Source Administrator, create an ODBC connection to the Oracle Business Analytics Warehouse Teradata database.

2. Set the following parameters:

   Field=Enter

   DateFormat=AAA

   SessionMode=ANSI

   NoScan=Yes

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

4.3.4.4 Set Environment Variables

Before you run the Informatica PowerCenter installer, you must set environment variables as described in 'Step 5. Configure Environment Variables' in the chapter 'Before You Install' in the PowerCenter Installation and Configuration Guide. For UNIX machines, you must set environment variables for database client, Library Path and Locale as described. Informatica PowerCenter 8.1.1 SP 4 documentation is

included on the Informatica PowerCenter DVD provided with Oracle Business Intelligence Applications.

4.3.4.5 How to Install the Java JDK

The DAC Client and DAC Server require Java JDK. Install Java JDK on the machines that will host the DAC Client and DAC Server. On Windows, you must install JDK with no spaces in the directory path.

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 the DAC server or 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 Fusion Edition.*

To install Java SDK

1. Download and install the Java JDK for your operating system(s) from a web site that provides Java JDK downloads (for example, the Sun or IBM web site). You must download and install the Java JDK for either 32 bit or 64 bit depending on your operating system.

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

4.3.4.6 Download 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 are not installed as part of Oracle Business Intelligence 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 download the 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).

4.4 Installing Oracle Business Intelligence Applications Software

This section explains how to install Oracle Business Intelligence Applications.

The Oracle Business Intelligence Applications installer runs on Windows, and requires the Oracle Business Intelligence infrastructure to be installed. For more information on Oracle Business Intelligence infrastructure requirements, including versions supported for this release of Oracle BI Applications, see Section 4.3.1, "Oracle Business Intelligence Infrastructure Requirements".

4.4.1 How to Run the Oracle Business Intelligence Applications Installer (Windows)

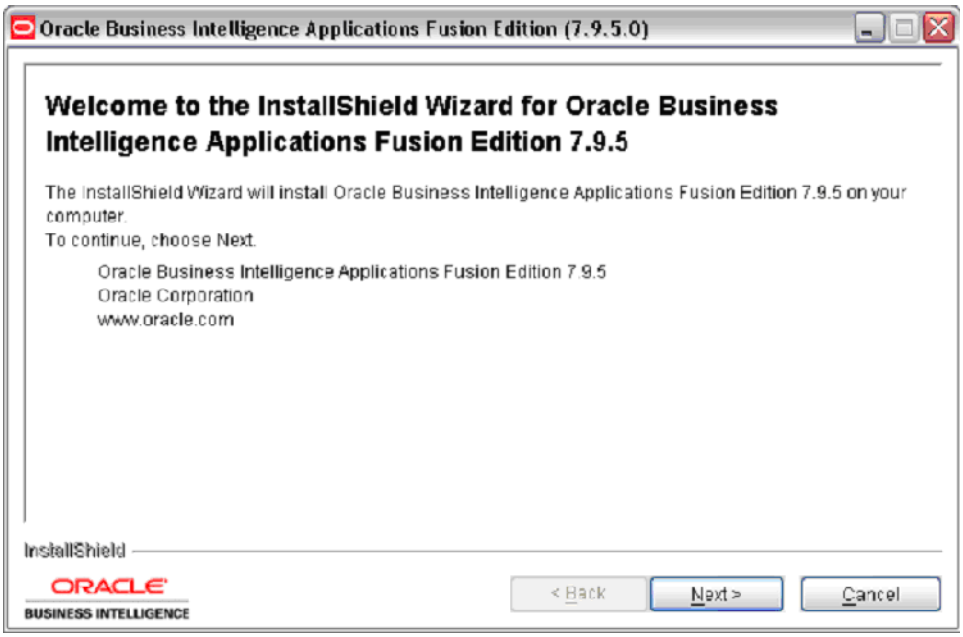
This section explains how to install the Oracle Business Intelligence Applications files using the Oracle Business Intelligence Applications Installation Wizard.

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

Note: If you have a previous version of Oracle Business Intelligence Applications installed, you must uninstall this version before you run the installer for Oracle Business Intelligence Applications Version 7.9.5. If you do not uninstall the old version, some Version 7.9.5 directories will not be correctly installed. (Make a back-up of your RPD, presentation catalog, and DAC and dwrep directories before you uninstall the old version.)

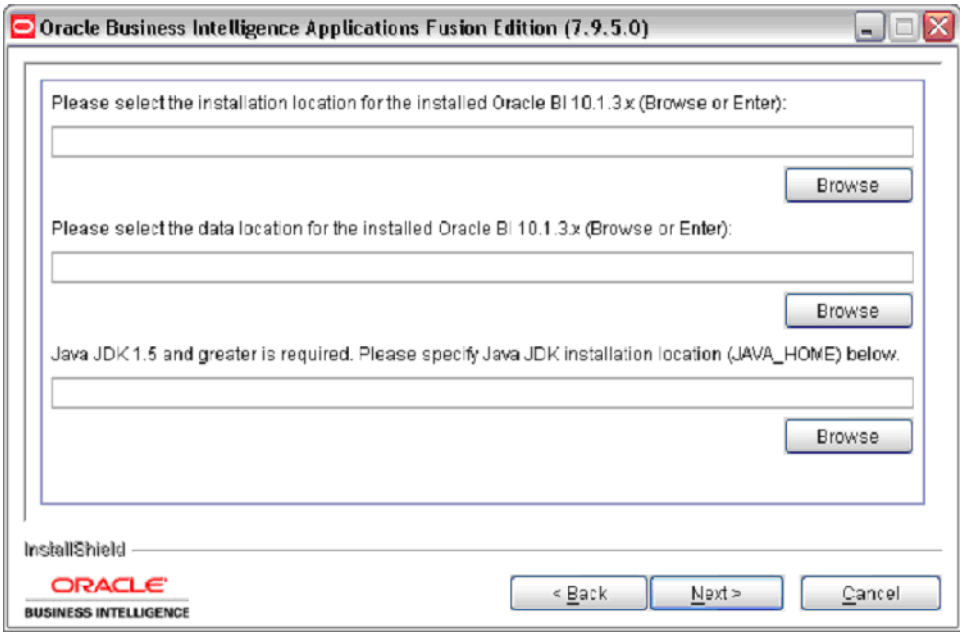
To install Oracle Business Intelligence Applications on Windows

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

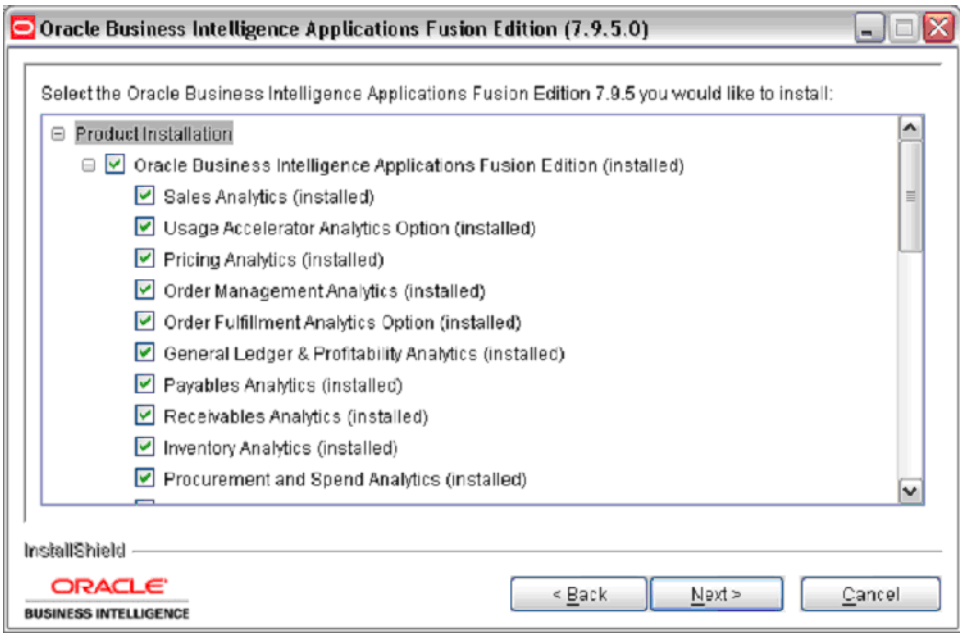


Note: To run the installer in console (or text) mode, run the command setup.exe -console. You do not see following dialogue screens in console installation mode. Instead, you enter input as plain text in the terminal window when prompted.

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



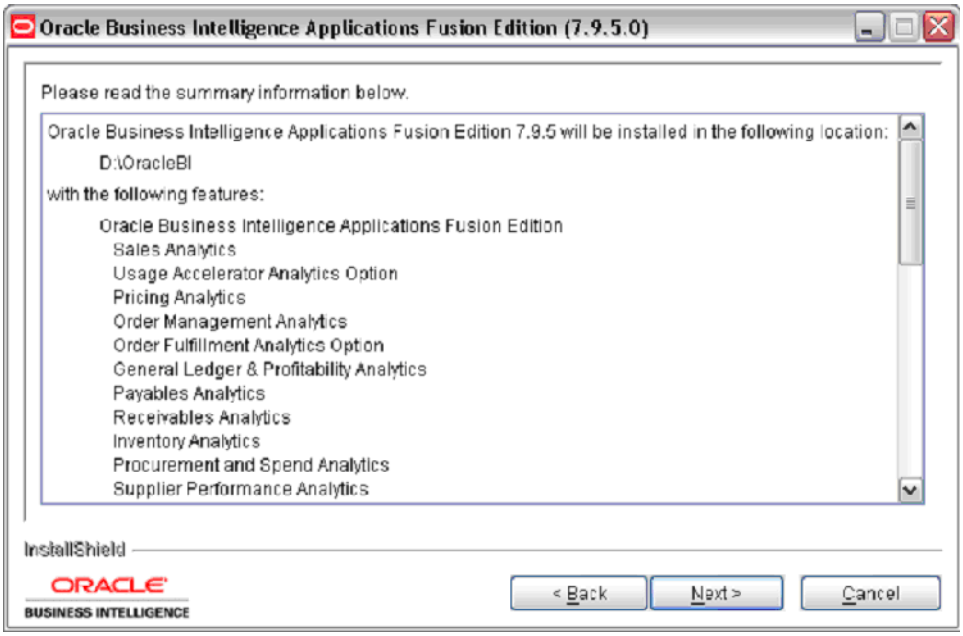
3. Enter or browse for the location for the Oracle Business Intelligence infrastructure (for example, <DRIVE>:\OracleBI\), the Oracle Business Intelligence infrastructure data (for example, <DRIVE>:\OracleBIData\), and the Java SDK (for example, <DRIVE>:\jdk1.5).
4. Click Next to display the Select Oracle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



To determine the applications you should install for the Oracle Business Intelligence Applications licenses you have purchased, see the spreadsheet entitled '2-Installer Options' in the Oracle Business Intelligence Applications Fusion Edition Licensing and Packaging Guide. This guide is part of the Oracle Business Intelligence Media Pack.

Once you have installed applications, you can deploy them for one or more source systems selectively when you create the DAC metadata repository in a later step (for more information, see Section 4.9, "Importing the DAC Metadata").

6. If you selected more than one application on the Select Oracle Business Intelligence Applications page, click Next at the informational page.
7. At the summary page, review the summary information and click Next to start the installation.



The installer installs the Oracle Business Intelligence Applications directories and files in the Oracle Business Intelligence infrastructure installation (for example, <DRIVE>:\OracleBI\), and installs the Oracle BI Repository file and Presentation Catalog.

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

8. Click Finish.

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

When the installation is complete, verify that the following directories or files are installed:

- The DAC folder.
- The dwrep folder.
- The Oracle Business Intelligence Applications repository file named EnterpriseBusinessAnalytics.rpd in the OracleBI\server\Repository folder.
- The Presentation Catalog in the OracleBIData\web\catalog folder. The Presentation Catalog is provided as a zipped file named EnterpriseBusinessAnalytics.zip and will need to be unzipped before it can be used.
  - Language files (for the languages available for Oracle BI Applications) in OracleBIData\web\res folder.
  - Seed Data folder containing localization files.
  - Upgrade folder containing repository files from previous versions of Oracle BI Application and Siebel Analytics Application releases.

Notes

- The credentials required to log into the EnterpriseBusinessAnalytics.rpd file using the Oracle Business Intelligence Administration Tool are:
  - Username: Administrator
  - Password: SADMIN
- Refer to the Oracle Business Intelligence Enterprise Edition or Oracle Business Intelligence Standard Edition One documentation for more information on working with the repository and Presentation Catalog.

4.5 Installing Informatica PowerCenter Services Software

This section explains how to install Informatica PowerCenter Services, and contains the following topics:

- Section 4.5.1, "How to Install Informatica PowerCenter Client Tools"
- Section 4.5.2, "How to Install Informatica PowerCenter Services"

Notes

- You need to determine the machines that will host Informatica PowerCenter Services and Informatica PowerCenter Client Tools. The PowerCenter Services can be installed on UNIX or on Windows. The PowerCenter Client Tools must be installed on Windows.

  Oracle recommends that you:

  - co-locate the DAC Client with the PowerCenter Client Tools.

  - co-locate the DAC Server with the PowerCenter Services.

  DAC produces parameter files that are used by Informatica. If an Execution plan fails in DAC and you want to debug the workflow by running it directly from Informatica, then the parameter file produced by DAC should be visible to Informatica.

- You must meet all requirements and complete pre-install tasks as described in Section 4.3, "Mandatory Requirements and Pre-installation Tasks" and in chapter 'Before You Install' of the PowerCenter Installation and Configuration Guide. Informatica PowerCenter 8.1.1 SP 4 documentation is included on the Informatica PowerCenter DVD provided with Oracle Business Intelligence Applications.

- When you install PowerCenter Services and Client Tools version 8.1.1 SP4, use the Informatica PowerCenter installation documentation in conjunction with the steps provided in the following sections. The Oracle Business Intelligence Applications Installation and Configuration Guide summarizes the steps and assumes a single-machine installation for the PowerCenter Services components. Note: For a new installation, you must select the 'Create new PowerCenter Repository' option on the PowerCenter Repository Configuration screen.

Note: If you have licensed Informatica PowerCenter separately from Oracle Business Intelligence Applications, skip this section and go to section Section 4.6, "Restoring the Oracle Business Intelligence Pre-Built Repository".

Make sure that you have updated to the required version of Informatica PowerCenter (for more information, see Section 4.3.2, "Informatica PowerCenter Requirements").

4.5.1 How to Install Informatica PowerCenter Client Tools

This section explains how to install Informatica PowerCenter 8.1.1 SP4 Client Tools for an Oracle Business Intelligence Applications deployment. For detailed generic information about installing Informatica 8.1.1, refer to the *Informatica PowerCenter Installation and Configuration Guide*, and related documentation.

Note: Oracle recommends that you co-locate the PowerCenter Client Tools with the the DAC Client.

Follow this procedure to install the Informatica PowerCenter Client Tools for an Oracle Business Intelligence Applications deployment.

To install Informatica PowerCenter Client tools on Windows

1. Run the program install.exe from the PowerCenter_8.1.1_SE_for_Windows folder on the Informatica PowerCenter DVD provided with Oracle Business Intelligence Applications.

Alternatively, run install.bat from the PowerCenter_8.1.1_SE_for_<OS>\Client\Disk1 folder.

2. If the installer displays the 'Please select an option to install' dialog, click Client.
3. Follow the instructions on the Informatica PowerCenter Client 8.1.1 installation wizard.
4. Install Informatica PowerCenter SP4 patch from the PowerCenter_8.1.1_SP4_for_<OS>\Client\Disk1 folder on the Informatica PowerCenter DVD provided with Oracle Business Intelligence Applications, as follows:
   - Install Client - accept all default installation settings.

4.5.2 How to Install Informatica PowerCenter Services

This section explains how to install Informatica PowerCenter Services 8.1.1 SP4 for an Oracle Business Intelligence Applications deployment. For detailed generic information about installing Informatica 8.1.1, refer to the *Informatica PowerCenter Installation and Configuration Guide*, and related documentation.

This section explains how to install Informatica PowerCenter Services for an Oracle Business Intelligence Applications deployment. During the Informatica PowerCenter Services installation, you must select the 'Create new PowerCenter Repository' option on the PowerCenter Repository Configuration screen.

Note: Oracle recommends that you co-locate the PowerCenter Services with the DAC Server.

To install Informatica PowerCenter Services

1. Start the Informatica PowerCenter Services installer.

   To start the installer on Windows, insert the Informatica PowerCenter DVD provided with Oracle Business Intelligence Applications into the DVD drive and run install.exe from the PowerCenter_8.1.1_SE_Windows folder. At the 'Please select an option to install dialog', click Server.

   To start the installer on UNIX or Linux, run install.sh at a shell command line. The PowerCenter 8.1.1 installation files are located in folder PowerCenter_8.1.1_SE_<OS> on the Informatica PowerCenter DVD provided with Oracle Business Intelligence Applications.

   On UNIX and Linux, the PowerCenter installer checks the operating system and verifies that the environment variables are correctly set. The environment variables must be specifically set to enable the PowerCenter installer to run processes and connect to databases and services. If the environment variables are not set, the PowerCenter installer gives you the option to stop the installation and modify the environment variables.

   On UNIX or Linux, select the option to use the GUI mode installation.
2. Follow the instructions on the Informatica PowerCenter Services installation wizard, as shown in the following table.

Note: The installation wizard pages that you see depend on the licensing options that you have purchased. For example, the Application Server Configuration page is only displayed if you have purchased the PowerCenter Data Analyzer option.

Page	Your Action	Notes
Welcome	Click Next.	
License Key	Click Browse, then locate and select the license key text file.	If you have purchased the 'Informatica OEM PowerCenter ETL Server and PowerConnect Adapters' license from Oracle, use the Oracle_All_OS_Prod.key license key file. This file is located in the DVD root directory of the Informatica PowerCenter DVD that is provided with Oracle Business Intelligence Applications.
Installation Prerequisites		
Installation Set	Select the **Install PowerCenter 8.1.1.** radio button.	For information about upgrading to Informatica PowerCenter 8.1.1, refer to Informatica PowerCenter Installation and Configuration Guide.
DA & MM License Key	Not applicable.	
Installation directory	Specify the installation folder.	On Windows, do not install Informatica in a directory that has a space in the directory name. For example, do not install in D:\Program Files\INFA. If you do install into a directory with a space in the name, you will cause errors in the DAC.
Application Server Configuration	Not applicable.	
Pre-Installation Summary	Click Install.	
Installing...	Not applicable.	
Domain Configuration	At the Create or Join Domain page, select the **Create New Domain** radio button.	
Domain Configuration	At the Configure Domain Database page, specify the database details.	Configure Domain Database page: ▪ Specify the database details for the database account that stores the PowerCenter domain configuration metadata. Use the database that you created earlier (for more information, see Section 4.3.4.1, "Create Databases"). You can use the same database that is used to store the Informatica repository.

Page	Your Action	Notes
Domain Configuration	At the Configure Domain page, specify the domain details.	Configure Domain page: ■ Informatica PowerCenter creates a default domain called Domain_<*install machine name*> and a default node called Node01_<*install machine name*>. ■ Specify a domain password (for example, admin), which you be use to log into PowerCenter Administration Console. ■ Make sure that you select the **Create Repository Service** check box.
Domain Configuration		
Node Configuration	At the PowerCenter Repository Configuration page, select the **Create new PowerCenter repository** radio button, and specify the Services details. You must specify the appropriate repository Code Page setting.	PowerCenter Repository Configuration page: ■ Specify the database details that you created for the Informatica repository. ■ Choose a code page that is compatible with the code page of the machine where you install PowerCenter. For more information, refer to the Informatica PowerCenter documentation, and Section 4.3.3, "Code Page and Data Movement Requirements".
Node Configuration	At the Configure Integration Service page, select the **Create Integration Service** check box. Select the data character movement mode, either ASCII or Unicode.	PowerCenter Services Configuration page: ■ The username here is the username for the Informatica repository (for example, Administrator). ■ The password here is the password for the Informatica repository (for example, Administrator). ■ Select Unicode if non-ASCII characters are to moved from source database to the Oracle Business Analytics Warehouse. Otherwise, select ASCII. For more information, see Section 4.3.3, "Code Page and Data Movement Requirements".
Authentication	Not applicable.	
Repository Configuration	Not applicable.	
Post-Installation Summary	Click Done.	

3. Stop any Informatica services that are running.
4. Install the Informatica PowerCenter SP4 patch from the PowerCenter_8.1.1_SP4_<OS> folder, as follows:

- Install Server - select the Custom installation mode and clear the check box for **PowerCenter Data Analyzer 8.1.1 SP4** and **PowerCenter Metadata Manager 8.1.1 SP4**. Accept the other default installation settings.

4.6 Restoring the Oracle Business Intelligence Pre-Built Repository

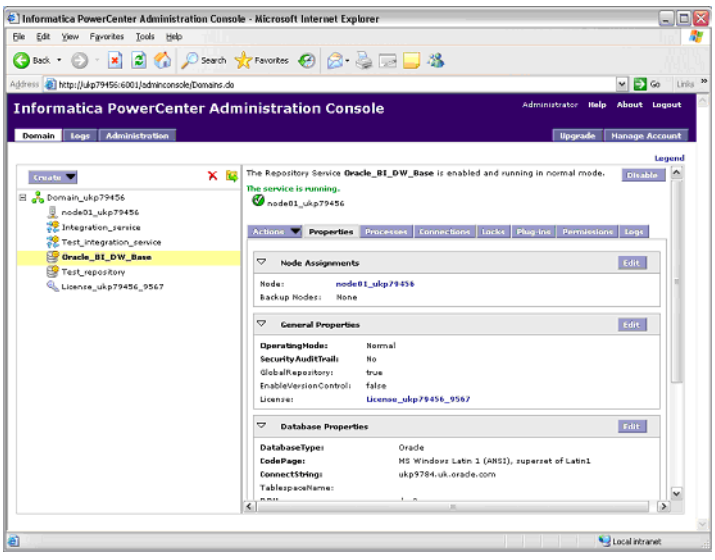
An Informatica repository file called Oracle_BI_DW_Base.rep is installed into the OracleBI\dwrep\Informatica\Repository directory during the Oracle Business Intelligence Applications installation. For use with deployments where the Oracle Business Analytics Warehouse is on Teradata, a repository file called Oracle_BI_DW_Teradata.rep is installed.

You use the Restore option in Informatica PowerCenter Administration Console to load the pre-built Oracle_BI_DW_Base repository or Oracle_BI_DW_Teradata (in a Teradata environment).

To load the pre-built Oracle_BI_DW_Base.rep or Oracle_BI_DW_Teradata.rep repository into Informatica

1. Copy the file Oracle_BI_DW_Base.rep (or Oracle_BI_DW_Teradata.rep) from the OracleBI\dwrep\Informatica\Repository directory on the machine where the Oracle Business Intelligence Applications installer was run to the following folder on the machine where Informatica PowerCenter Services has been installed:
   - On Windows, to the \Informatica PowerCenter8.1.1\server\infa_shared\Backup directory.
   - On UNIX or Linux, to $Informatica/PowerCenter8.1.1/server/infa_shared/Backup directory.
2. In Informatica PowerCenter Administration Console, select the select the Repository Service that was created during the installation process (for example, Oracle_BI_DW_Base or Oracle_BI_DW_Teradata).

   For information about how to connect to Informatica PowerCenter Administration Console, see Section A.3, "How to Log Into Informatica PowerCenter Administration Console".



3. In the Properties tab, click Edit in the General properties area.
4. Make sure that the **OperatingMode** value is Exclusive.

5. Click OK.
6. Choose Actions, then Delete Contents.
7. At the **Delete contents for <repository name>** dialog, enter the repository username and password (for example, Administrator\Administrator), then click OK.
8. Choose Actions > Restore Contents.
9. At the Restore Contents dialog, select Oracle_BI_DW_Base (or Oracle_BI_DW_Teradata for Teradata installations) from the **Select backup file** drop down list.
10. Select the **Restore as new** check box.
11. Click OK to start the restore process.
12. When the restore process is complete, click Close (or click Save first to save the logging information).

    When the restore is complete (the process typically takes approximately ten to twenty minutes), you will see a 'Success' message.

    When a repository is restored, the repository becomes a standalone repository. After restoring the repository, you need to promote it to a global repository.
13. In the Properties tab, click Edit in the General properties area.
14. Change the **OperatingMode** value to Normal.
15. If prompted, enter the repository username and password.

> **Note:** The pre-built Oracle_BI_DW_Base and Oracle_BI_DW_Teradata repository files are versioned from Oracle Business Intelligence Applications Version 7.9.3 and higher. For more information about the versioned repositories, see Appendix J, "About the Versioned Informatica Repository".

4.7 Configuring Informatica PowerCenter Components

This section explains how to install and configure Informatica PowerCenter Services for use with Oracle Business Intelligence Applications, and contains the following topics:

- Section 4.7.1, "How to Copy Source Files and Lookup Files"
- Section 4.7.2, "Setting PowerCenter Integration Services Relaxed Code Page Validation"
- Section 4.7.3, "Setting PowerCenter Integration Services Custom Properties"

4.7.1 How to Copy Source Files and Lookup Files

You need to copy source files and lookup files from the Oracle Business Intelligence Applications installation directory to the Informatica directory on the Informatica PowerCenter Services machine.

Note: The following instructions assume the Informatica default directory locations for source and look up files.

To copy source files and lookup files on Windows

1. Copy the source files in \OracleBI\dwrep\Informatica\SrcFiles to \Informatica PowerCenter8.1.1\server\infa_shared\SrcFiles on the Informatica PowerCenter Services machine.
2. Copy the lookup files in \OracleBI\dwrep\Informatica\LkpFiles to Informatica PowerCenter8.1.1\server\infa_shared\LkpFiles on the Informatica PowerCenter Services machine.

To copy source files and lookup files on UNIX or Linux

1. Copy the source files in \OracleBI\dwrep\Informatica\SrcFiles to the $Informatica/PowerCenter8.1.1/server/infa_shared/SrcFiles on the Informatica PowerCenter Services machine.
2. Copy the lookup files in \OracleBI\dwrep\Informatica\LkpFiles to the $Informatica/PowerCenter8.1.1/server/infa_shared/LkpFiles on the Informatica PowerCenter Services machine.

4.7.2 Setting PowerCenter Integration Services Relaxed Code Page Validation

PowerCenter Integration Services must be configured for relaxed code page validation.

To configure Informatica PowerCenter Integration Services for relaxed code page validation:

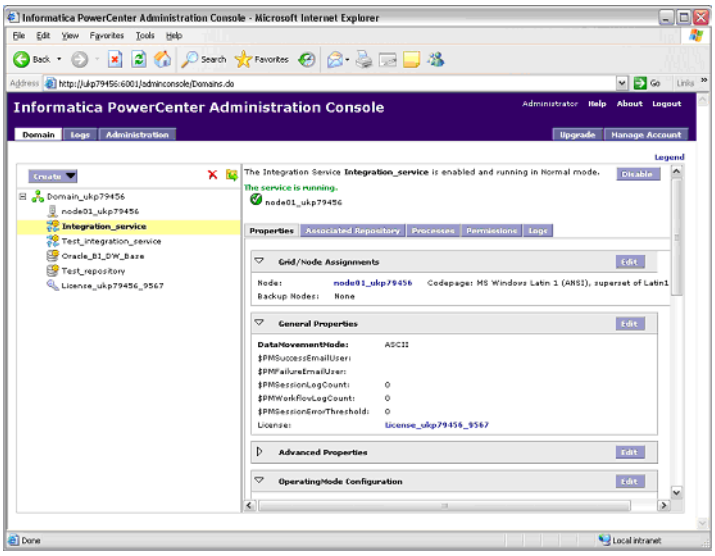
1. Log in to Informatica PowerCenter Administration Console.
2. Select the Integration Service.
3. Click the Properties tab > Configuration Properties > Edit.
4. Change the value for property 'ValidateDataCodePages' from 'Yes' to 'No'.

4.7.3 Setting PowerCenter Integration Services Custom Properties

To set Informatica PowerCenter Integration Services Custom Properties

1. In Informatica PowerCenter Administration Console, select the Integration Service.

   For example, on the machine where Informatica PowerCenter Services is installed, choose Programs, then Informatica PowerCenter 8.1.1, the Services, then Launch Admin Console.



2. Display the Properties tab.
3. In the Custom Properties area, click Edit.
4. Create a Custom Property by clicking Add to display new **Name** and **Value** fields.
5. For each of the custom properties in the table below, click Add and add a new custom property with an appropriate value:

Table 4–2 Custom Properties for Informatica PowerCenter Integration Services

Custom Properties Name	Custom Properties Value	Notes
ValidateDataCodePages	No	
ServerPort	<Server Port Number>	For <Server port number>, enter the number of an available port. For example, 4006.This custom property configures Informatica PowerCenter Services to listen on <Server port number>. The DAC communicates with the PowerCenter Integration Services service using this port.
overrideMpltVarWithMapVar	Yes	
DisableDB2BulkMode	Yes	Add this custom property and set value to Yes if your Oracle Business Analytics Warehouse is on DB2/390 database.
SiebelUnicodeDB	[user_OLTP]@[connectString_OLTP] [user_OLAP]@[ConnectString_OLAP] For example, oltp@db204007.host.com olap@db204008.host.com. **Note**: Always leave a space between the connection strings.	Set this value if your data movement is from Unicode to Unicode

6. Click OK to save the details.
7. Make sure that the Integration Service and Repository Service that you created during the Informatica PowerCenter installation are running.

The Integration Service and Repository Service must be registered in the DAC in a later step (for more information, see Section 4.12.2, "How to Register the Informatica Services in the DAC").

4.8 Installing and Configuring the DAC Client

This section explains how to install and configure the DAC Client, and contains the following topics:

- Section 4.8.1, "How to Install and Set Up the DAC Client"
- Section 4.8.2, "Enabling DAC Client Communication with Informatica PowerCenter"
- Section 4.8.3, "How to Create a DAC Connection"
- Section 4.8.4, "Creating the DAC Repository Schema"

4.8.1 How to Install and Set Up the DAC Client

This section explains how to install and set up the DAC Client.

The DAC Client is installed by the Oracle Business Intelligence Applications installer. The DAC Client only runs on a Windows platform. Oracle recommends that you run the DAC Client on the Windows machine that runs the Informatica PowerCenter Client Tools. This section explains how to install and configure a DAC Client.

Note: You do not need Oracle Business Intelligence Enterprise Edition installed on a machine to run a DAC Client.

Notes

- JDK 1.5 or higher should be installed. For more information, see Section 4.3.4.5, "How to Install the Java JDK".
- Download the Hibernate files from the Hibernate web site. For more information, see Section 4.3.4.6, "Download Hibernate Libraries".

To install and set up the DAC Client, do the following:

1. Copy the \OracleBI\DAC\ directory and its sub-folders from the Oracle Business Intelligence Applications installation (on Windows) to the target Windows machine (for example, to a directory called /DAC/).

   Alternatively, if the target machine already has an Oracle Business Intelligence infrastructure installed (for example, Oracle Business Intelligence Enterprise Edition), you can run the Oracle Business Intelligence Applications installer on the target machine to install the DAC Client. For information about how to run the Oracle Business Intelligence Applications installer, see Section 4.4.1, "How to Run the Oracle Business Intelligence Applications Installer (Windows)").

   Note: If you will run the DAC Client from the machine on which the Oracle Business Intelligence Applications software was run, ignore this step and proceed to step 2.

2. Copy the hibernate files to the appropriate DAC directories, as follows:

a. Copy the Hibernate files from the \hibernate-3.2 directory where you downloaded and unzipped the Hibernate files to the \DAC directory on the DAC Client machine, as described in the table below.

Table 4–3 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.

3. Install JDBC drivers for DAC database connectivity.

   You must install 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 the DAC Client is 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 4.15.4, "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 file sqljdbc.jar and copy it to the \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.-->.

Download he SQL Server JDBC drivers for SQL Server 2000 from the Microsoft web site. Copy the Copy the files msbase.jar, mssqlserver.jar, and msutil.jar to the \DAC\lib folder.

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. Configure DAC config.bat file and configure a connection for DAC repository, as follows:
   a. In the DAC directory, open the config.bat file.
   b. 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
      ```

5. 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 Enabling DAC Client Communication with Informatica PowerCenter

The DAC Client uses the Informatica pmrep and pmcmd command line programs to communicate with Informatica PowerCenter. The DAC Client uses pmrep to synchronize DAC tasks with Informatica workflows and to keep the DAC task source and target tables information up to date.

4.8.2.1 How to Install Informatica pmcmd and pmrep Command Line Programs

The pmrep program is installed in the Informatica PowerCenter Client and Informatica PowerCenter Services bin directories. Because of the requirement to co-locate the DAC Client with the PowerCenter Client, the pmrep program is available on the machine for the DAC Client to use.

The pmcmd program is installed in the PowerCenter Services bin directories. If PowerCenter Services 8.1.1 SP 4 has been installed on the same machine as the DAC Client and PowerCenter Client 8.1.1 SP4, copy the pmcmd program from the PowerCenter 8.1.1\server\bin directory to PowerCenter 8.1.1\client\bin directory.

If the DAC Client machine does not co-host the PowerCenter Services, follow the instructions below to install pmcmd on the DAC Client machine:

1. If Informatica PowerCenter Services 8.1.1 SP 4 is installed on a Windows machine other than the one that hosts the DAC Client, do the following:

a. Copy the pmcmd.exe file from the \PowerCenter 8.1.1\server\bin directory on the machine where PowerCenter Services is installed to the \PowerCenter 8.1.1\client\bin directory on the DAC Client machine.

2. If Informatica PowerCenter Services is installed on a UNIX or Linux machine, and an installation on Windows is not available, run the Informatica PowerCenter Services installer on the DAC Client machine or (any other Windows machine) to install only Integration Services as follows:

   a. Run the Informatica PowerCenter Services installer as described in section Section 4.5.2, "How to Install Informatica PowerCenter Services".

   b. In the Domain Configuration section, on the Create or Join Domain page, select the **Create New Domain** radio button

   c. At the Configure Domain Database page, specify the database details for the domain configuration database.

      You may use the same database that stores the domain configuration for your PowerCenter Services installation on UNIX. However, you will have to create a different database account.

   d. At the Configure Domain page, specify the domain details.

      Note: Do not check the **Create Repository Service** check box.

   e. Complete the installation.

   f. Apply Service Pack 4.

   g. Copy the pmcmd.exe file located in PowerCenter 8.1.1\server\bin to PowerCenter 8.1.1\client\bin.

4.8.2.2 How to Set Environment Variables for a DAC Client

In order for the DAC Client to be able to use the pmrep and pmcmd programs, you need to define the path of the Informatica Domain file 'domains.infa'.

To define the Informatica Domains File path:

1. Locate the file domains.infa in the root Informatica PowerCenter installation directory and note down the directory path of this file.

   For example, D:\Informatica\PowerCenter8.1.1.

2. Create an environment variable called INFA_DOMAINS_FILE with the value set to the directory path to the domans.infa file, as follows:

   - On Windows, display the Windows Environment Variables dialog (that is, from the Windows Control Panel, choose System, then Advanced, then Environment Variables), and create a System variable with the following values:
     - Variable name: INFA_DOMAINS_FILE
     - Variable value: <*directory path of domains file*>\domains.infa

       The path should include the name of the file. For example, 'D:\Informatica\PowerCenter8.1.1\domains.infa'.

3. Add the directory path to Informatica PowerCenter binaries to the PATH environment variable as follows:

   In the Windows System Properties > Environment Variables dialog, add the path of the Informatica \Client\bin directory to the PATH environment variable. For example: <drive>:\Informatica\PowerCenter8.1.1\client\bin.

4.8.2.3 How to Verify that the DAC Client is able to use pmrep and pmcmd

From a Windows command prompt, execute pmrep and then pmcmd. The test is successful if the pmrep and pmcmd prompts appear.

If pmrep and pmcmd are not recognized, then:

- Ensure Service Pack 4 has been applied to Informatica PowerCenter 8.1.1.
- Verify that the INFA_DOMAINS_FILE variable points to the domains.infa file located in the Informatica directory.
- Verify that the PATH variable includes the path to the Informatica binaries (\Informatica\PowerCenter\client\bin), and that pmrep.exe and pmcmd.exe exist in the \bin folder.

4.8.3 How to Create a DAC Connection

This section explains how to create a DAC Connection. A DAC connection is a stored set of login details that enable you to log into the DAC Client and connect to the DAC Repository.

Note: Before you start this procedure, you need to create a database for the DAC metadata repository. For more information, see Section 4.3.4.1, "Create Databases".

To create a DAC connection

1. Start the DAC client by navigating to the \DAC directory and double-clicking the startclient.bat file.

> **Note:** If you installed the DAC Client by running the Oracle Business Intelligence Applications installer, you can also start the DAC Client from the Start menu > Oracle Business Intelligence > Oracle DAC > DAC Client.

2. In the Login... dialog, select Configure.
3. In the Configuring... dialog, select Create Connection, and then click Next.
4. Enter the appropriate connection details as specified in the table below.

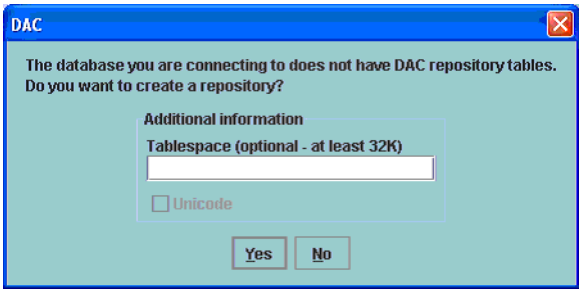
Field	Required Value
Name	Specify a unique name for the connection to the DAC repository (for example, DAC).
Connection type	Specify the type of database in which the DAC repository is stored.
Connection String, or Database name, or TNS Name, or Instance	Specify the database name or database account name of the DAC repository. If you are using: ■ Oracle (OCI8), use the tnsnames entry. ■ Oracle (Thin), use the instance name. ■ SQL Server, use the database name. ■ DB2-UDB, use the connect string as defined in the DB2 configuration.
Database Host	Specify the machine name of the machine where the DAC Repository database resides.
Database Port	Specify 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. To test the connection, select Test Connection and enter the Table owner name and password for the DAC repository.
6. Select Apply to save the connection details and return to the login dialog.
7. Select the new connection from the Connection drop down list, specify a table owner name and password for the DAC repository, and click Login.

The DAC Repository that you connect to does not contain the DAC repository tables. In the following section (Section 4.8.4, "Creating the DAC Repository Schema"), you will create the DAC repository schema.

4.8.4 Creating the DAC Repository Schema

When you log into DAC and connect to a DAC repository for the first time, the DAC detects that the DAC schema does not exist in the database and you are asked whether you want to create a repository.



In the dialog box, click Yes to create the repository tables.

For a repository on Oracle or DB2 database, optionally specify a tablespace.

The Unicode check box is available for a repository on MS SQL Server or DB2 database. Check the Unicode check box if your deployment requires a Unicode schema to be created. To determine if your deployment requires a Unicode schema for the DAC repository, see Section 4.3.3, "Code Page and Data Movement Requirements".

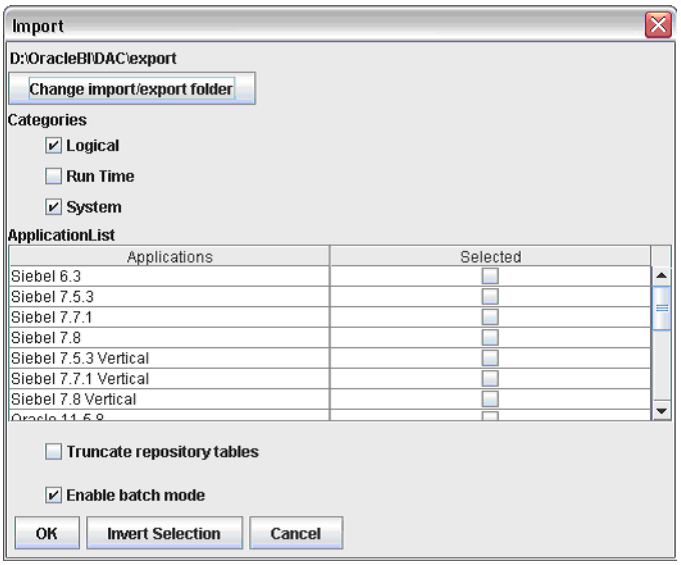
4.9 Importing the DAC Metadata

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

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

To import metadata into the DAC repository

1. In the DAC Client, choose Tools, then DAC Repository Management, then Import to display the Import dialog.



   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

2. Select the **Logical** check box, and the **System** check box.
3. In the Applications List table, use the check boxes in the **Selected** column to specify the source system applications for which you import the ETL metadata, as follows:

   a. Select the check box next to Universal.

      To avoid possible ETL errors in other applications (for example, missing tasks), you must import the Universal application.

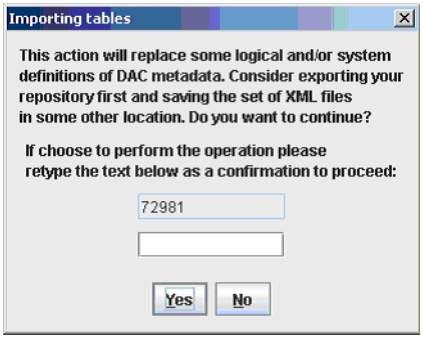
   b. Select the check box for any other business applications that you want to deploy.

      > **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 4.10.1, "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 \DAC\log\import.log file to diagnose errors.

   Note: The import process typically takes between one and five hours, depending on the location of the database that stores the DAC repository, the speed of the network, and the number of applications selected in the **Applications** list on the Import dialog.

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 \DAC\ directory.

4.10 Creating the Oracle Business Analytics Warehouse Database Tables

This section explains how to create tables in the Oracle Business Analytics Warehouse database.

Note: Before you start this procedure, you need to create a database for the data warehouse (for more information, see Section 4.3.4.1, "Create Databases").

The Oracle Business Analytics Warehouse tables are created by the DAC Client. The DAC Client uses ODBC connections to the Oracle Business Analytics Warehouse database for this procedure. Ensure that you have created an ODBC connection to the Oracle Business Analytics Warehouse database as described in Section 4.3.4.3, "Create ODBC Database Connections".

If your Oracle Business Analytics Warehouse database is Oracle, you must use the Oracle Merant ODBC driver to create the ODBC connection. This driver is installed by the Oracle Business Intelligence Applications. The DAC Client must be installed by running the Oracle Business Intelligence Applications installer so that the driver is installed on the machine. Create the ODBC DSN to the Oracle Business Analytics Warehouse as described in Section 4.3.4.3, "Create ODBC Database Connections".

Before you perform the procedures described in this section 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 4.3.4.1.1, "How to Create the SSE Role").

If your Oracle Business Analytics Warehouse database is not Teradata, refer to Section 4.10.1, "How to Create Data Warehouse Tables" to create the data warehouse tables.

If you are using a Teradata database as the Oracle Business Analytics database, the DAC creates a SQL file to create the schema tables, not the tables themselves. Follow the steps in Section 4.10.2, "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.

4.10.1 How to Create Data Warehouse Tables

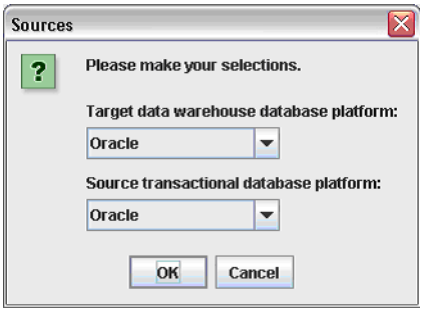
Use this procedure to create the data warehouse tables.

To create data warehouse tables

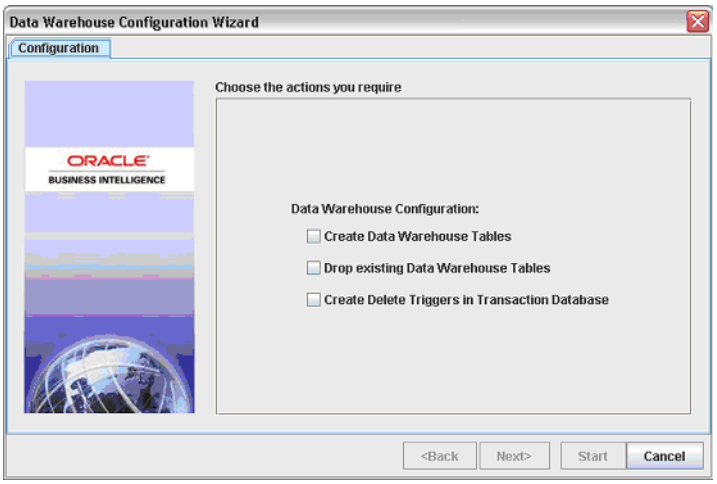
1. In the DAC Client, choose Tools, then ETL Management, then Configure.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client").

2. In the Sources dialog, select the database platform for the target data warehouse and source transactional database.

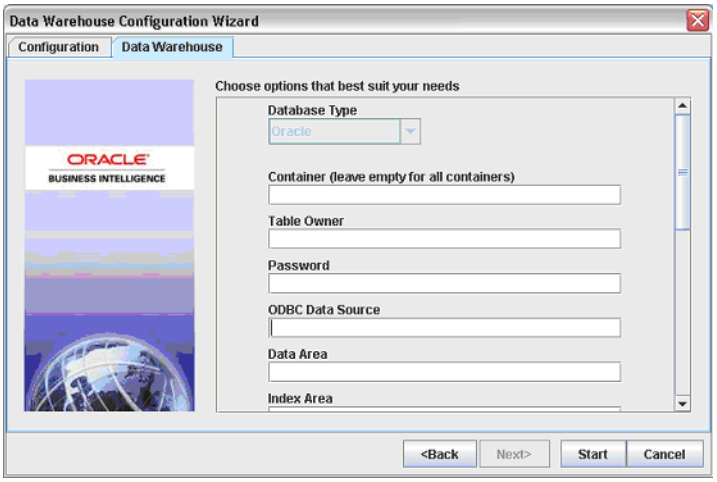


3. Click OK to display the Data Warehouse Configuration Wizard.



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	The name of the source business applications for which you want to create the data warehouse tables. **Note**: You must leave this field blank, unless told to specify a value by Oracle Support. If you leave the **Container** field blank, the DAC creates a container by default for all of the following: ■ The source business applications that you selected when you imported the seed data into the DAC metadata repository earlier (for more information, see Section 4.9, "Importing the DAC Metadata"). ■ Any copies of those source system applications. ■ Any additions made those source system applications. ■ Any new source system applications that you have custom built. For example, you might do the following: ■ Import the seed data for Oracle 11.5.9 and 11.5.10 ■ Make a copy of Oracle 11.5.10 and call it 'My Oracle 11.5.10'. ■ Add two data warehouse tables to 'My Oracle 11.5.10' ■ Create a new source system application called 'My new source' and add 10 data warehouse tables to 'my New Source'. If you leave the **Container** field blank, the The DAC will create the following containers: ■ Oracle 11.5.9 ■ Oracle 11.5.10 ■ My Oracle 11.5.10 ■ My new Source 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. 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. For example: ■ If you want to deploy Oracle 11.5.9, enter: 'Oracle 11.5.9'. ■ If you want to deploy Oracle 11.5.9 and 11.5.10, enter: 'Oracle 11.5.9,Oracle 11.5.10'. For more information about containers, see *Oracle Business Intelligence Data Warehouse Administration Console Guide*.
Table Owner	Valid database owner, username, or account that you set up to hold the data warehouse (for more information, see Section 4.3.4.1, "Create Databases").
Password	Valid database user password for the database owner, username, or account that you specified in the Table Owner field (for more information, see Section 4.3.4.1, "Create Databases").
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 Section 4.3.4.3, "Create ODBC Database Connections".

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. The database must be set to support a Unicode schema (for more information, see Section 4.3.3, "Code Page and Data Movement Requirements").

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.

4.10.2 How to Create Data Warehouse Tables on a Teradata Database

This section explains how to create and delete 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 3.7, "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. In the DAC Client, choose Tools, then ETL Management, then Configure.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client").

2. In the Sources dialog, 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.

4.11 How to Install and Configure the DAC Server

This section explains how to install and configure the DAC Server, and contains the following topics:

- Section 4.11.1, "Overview to Installing and Configuring the DAC Server"
- Section 4.11.2, "Installing the DAC Server"
- Section 4.11.3, "Configuring the Connection Between the DAC Server and DAC Repository"
- Section 4.11.4, "Enabling DAC Server Communication with Informatica PowerCenter"
- Section 4.11.5, "Starting the DAC Server"

4.11.1 Overview to Installing and Configuring the DAC Server

The DAC Server is installed by the Oracle Business Intelligence Applications installer. The DAC Server runs on Windows, Unix, and Linux platforms. Oracle recommends that you co-locate the DAC Server on the machine that runs the Informatica PowerCenter Services.

Note: You do not need Oracle Business Intelligence Enterprise Edition installed on a machine to run the DAC Server. If you want to use the Oracle Business Intelligence Applications installer to install the DAC Server on a Windows machine, you do need an Oracle Business Intelligence infrastructure installed (either with Oracle Business Intelligence Enterprise Edition or Oracle Business Intelligence Standard Edition One).

When installing a DAC Server, note the following pre-requisites:

- Make sure that JDK 1.5 or higher is installed. For more information, see Section 4.3.4.5, "How to Install the Java JDK".
- On UNIX or Linux, make sure that a zip and unzip utility is installed.
- Make sure that Informatica PowerCenter Services is co-located on the machine that will host the DAC Server. For more information, see Section 4.5, "Installing Informatica PowerCenter Services Software".

  The domain should be set properly and the server port set as a custom property.

- Make sure that database connectivity software has been installed for connection to source(s), to the Oracle Business Analytics Warehouse, and to the Informatica repository/DAC repository databases. For more information, see Section 4.3.4.2, "Install Database Connectivity Software".

4.11.2 Installing the DAC Server

On the machine that hosts Informatica PowerCenter Services, install a DAC Server by following the steps in this section.

You can install the DAC Server on Windows, UNIX, or Linux machines.

To install the DAC Server on Windows, see Section 4.11.2.1, "How to install a DAC Server on Windows".

To install the DAC Server on UNIX or Linux, see Section 4.11.2.2, "How to install a DAC Server on UNIX or Linux".

4.11.2.1 How to install a DAC Server on Windows

If you plan to run the DAC Server from the same machine that hosts the DAC Client, then no further steps are necessary to install of the DAC Server. Proceed to section Section 4.11.3, "Configuring the Connection Between the DAC Server and DAC Repository".

To install a DAC Server on a Windows machine that is different from the machine on which the DAC Client has been installed and configured, you can do either of the following:

- If the target machine already has an Oracle Business Intelligence infrastructure installed (either with Oracle Business Intelligence Enterprise Edition or Oracle Business Intelligence Standard Edition One), you can run the Oracle Business Intelligence Applications installer on the target machine to install the DAC Server. For information about how to run the Oracle Business Intelligence Applications installer, see Section 4.4.1, "How to Run the Oracle Business Intelligence Applications Installer (Windows)").
- (Recommended) You can install a DAC Server by copying across the \DAC\ directory from the Windows machine on which you installed and configured the DAC Client, as follows:

  1. Copy the \OracleBI\DAC\ directory from the DAC Client machine to the target machine.

     Note: The \DAC directory name on the target machine should not contain spaces.

     Note: If you have copied the DAC folder from the DAC Client machine and the DAC Client has been configured to use the Hibernate libraries, you do not have to install the Hibernate libraries again. Otherwise, install Hibernate files following instructions in described in step 2 of Section 4.8.1, "How to Install and Set Up the DAC Client".

     ---

  2. Copy the appropriate jdbc driver(s) to the \DAC\lib folder on the target machine. Follow instructions in described in step 3 of Section 4.8.1, "How to Install and Set Up the DAC Client".
  3. Configure DAC config.bat file, as follows:
     a. In the DAC directory, open the config.bat file.
     b. 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
```

4. Edit the DAC_HOME variable to point to the directory where you installed the DAC.

   For example:

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

4.11.2.2 How to install a DAC Server on UNIX or Linux

Oracle does not provide an installer for Oracle Business Intelligence Applications on Unix or Linux. Therefore, to install a DAC Server on Unix or Linux, you copy the \DAC folder from the Windows machine on which the DAC Client is installed to a Unix or Linux machine, as described below.

To copy over the DAC Server to a Unix or Linux machine:

1. On the Windows machine on which the DAC Client is installed, create a temporary directory (for example, a directory named \OracleBI_UNIX\).

   You will use this temporary directory to create a zip file for the Unix/Solaris or Linux deployment.

2. Copy the \DAC\ directory to the temporary directory (for example, \OracleBI_UNIX\.

3. From the \DAC\ directory in the temporary directory, remove the \export and \icons sub-folders.

4. Zip up the temporary directory (for example, \OracleBI_Solaris\).

5. Copy the ZIP file to the target Unix or Linux machine.

   Note: If you use FTP to copy the zip file, use binary mode.

6. On the target UNIX or Linux machine, place the zip file in a directory where you want to install the DAC server.

7. On the target machine, unzip the zip file.

   Shell scripts are provided in the /DAC directory. After copying these files to a UNIX or Linux machine and before using them, you might need to use a MS-DOS to UNIX conversion tool to convert the script files to UNIX format (that is, remove the carriage return and line feed characters). Alternatively, you can manually remove the carriage return and line feed characters from the script files. For more information, see Section A.6, "About the DAC Server Scripts for Bash and C Shells".

8. On the UNIX or Linux machine copy the appropriate jdbc driver(s) to the /DAC/lib directory as described below. You must copy all jdbc drivers for the database platforms that host source(s), Oracle Business Analytics Warehouse, and DAC repository databases:

   – Oracle. If you are using an Oracle database (other than 8.x), locate the directory where the Oracle client is installed on the machine. Copy the file named ojdbc14.jar in the jdbc/lib directory and paste it in the $DAC/lib folder, as described below.

     Note: Do not copy the ojdbc14.jar file from a different machine.

If you are using Oracle 8.x, copy the file named classes12.zip and paste it in the $DAC/lib folder. Also, edit the ORACLELIB setting in the config.sh or config.csh file as follows:

setenv ORACLELIB=$DAC_Home/lib/classes12.zip

The config.sh and config.csh files are located in the /DAC/ directory.

– DB2. If you are using a DB2 database, find the directory where the DB2 client is installed on the machine. In the Java sub-directory, copy the file named db2java.zip and paste it in the $DAC/lib folder.

Note: Do not copy the db2java.zip file from a different machine.

– MSSQL. The DAC 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 file sqljdbc.jar and copy it to the /DAC/lib directory.

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

You can use the Microsoft SQL Server 2000 JDBC Driver files if you edit the /DAC/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.--> Download he SQL Server JDBC drivers for SQL Server 2000 from the Microsoft web site. Copy the Copy the files msbase.jar, mssqlserver.jar, and msutil.jar to the /DAC/lib folder.

– 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 /DAC/lib directory. Depending on the Teradata JDBC version, you might not have some of the above files.

9. If required, edit the config.sh or config.csh file located in the /DAC directory to point to the correct version of the Java JDK by setting the JAVA_HOME environment variable.

4.11.3 Configuring the Connection Between the DAC Server and DAC Repository

You must configure the connection between the DAC Server and the DAC repository.

On Windows, you can use the DAC Client to configure a DAC Server that runs in the same \DAC folder. Optionally, or to configure a DAC Server installed in another folder or on another Windows machine, use the serverSetupPrompt.bat file to configure the repository connection.

On UNIX or Linux, use the serverSetupPrompt.sh or serverSetupPrompt.csh script to configure the connection between the DAC Server and the DAC repository, as follows:

- To configure the DAC Server repository connection using the DAC Client, see Section 4.11.3.1, "How to Configure the DAC Server Repository Connection Using the DAC Client (Windows)".
- To configure the DAC Server repository connection using the serverSetupPrompt scripts, see Section 4.11.3.2, "How to Configure the DAC Server Repository Connection Using serverSetupPrompt Scripts (Windows, UNIX or Linux)".

4.11.3.1 How to Configure the DAC Server Repository Connection Using the DAC Client (Windows)

If the DAC Server is co-located with a configured DAC Client in the same \DAC folder, you can set the connection between the DAC Server and DAC repository using the DAC Client, as described below.

To configure the connection between the DAC server and the DAC repository using the DAC Client

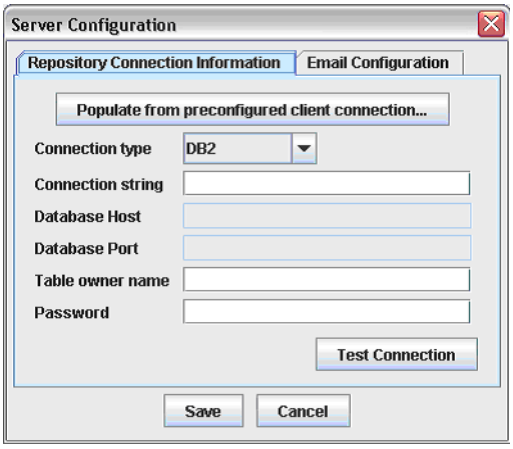
1. In the DAC Client, choose Tools, then DAC Server Management, then DAC Server Setup.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client").

   Note: The DAC repository that you connect to using the DAC Client is the one that will store the DAC Server repository connection information that you will specify in this procedure.

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

2. Click Yes to display the Server Configuration dialog.



3. In the Repository Connection Information tab, enter the appropriate information, as described in the table below.

   Tip: If the DAC Server is running on the same machine as the DAC Client, click **Populate from preconfigured client connection** to populate the fields with connection details from the DAC Client.

Field	Description
Connection type	Select the type of database that you are using to store the DAC metadata repository. Depending on what type you select, the connection details below change (see *Connection fields* below).

Field	Description
Connection fields (for example, Instance, TNS Name, Connection string/Database name.	Specify connection details for the database that stores the DAC metadata repository. ▪ If you select Oracle (Thin), you are prompted below for the following information: ▪ Instance (for example, mymachinename). ▪ Database Host (fully qualified, for example, mymachine.us.company.com). ▪ Database Port (for example, 1521). ▪ Table owner name, using the account that you created a database to store the DAC repository (for example, DAC). For more information, see Section 4.3.4.1, "Create Databases". ▪ Password (that is, for the above database account). ▪ If you select Oracle (OCI8), you are prompted below for a TNS name (for example, mymachinename@host.com). ▪ If you select DB2, you are prompted below for a Connection string. ▪ If you select MS SQL Server, you are prompted below for a Database name (for example, mydacdatabase).
Table owner name	The database table name or instance (for example, DAC) that you created to store the DAC repository (for more information, see Section 4.3.4.1, "Create Databases").
Password	The database or instance password (for example, DAC).

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 4.3.4.1, "Create Databases").

- When you created a DAC connection (for more information, see Section 4.8.3, "How to Create a DAC Connection").

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

4.11.3.2 How to Configure the DAC Server Repository Connection Using serverSetupPrompt Scripts (Windows, UNIX or Linux)

Use the serverSetupPrompt.sh or serverSetupPrompt.csh file to configure the connection between the DAC Server and the DAC repository when the DAC Server is installed on UNIX or Linux, as described below. For more information on the DAC Server scripts, see Section A.6, "About the DAC Server Scripts for Bash and C Shells".

Use the serverSetupPrompt.bat file to configure the connection between the DAC Server and the DAC repository when the DAC Server is installed on Windows, as described below.

To configure the connection between the DAC Server and the DAC repository using the serverSetupPrompt files:

1. Run the serverSetupPrompt script, as follows:
   - On Windows, double-click the serverSetupPrompt.bat located in the \DAC directory.
   - On UNIX or Linux, run serverSetupPrompt.sh or serverSetupPrompt.csh located in the /DAC directory.
2. Enter 1 in the 'Please make your selection' prompt to enter repository connection information.
3. Enter the number for the type of database storing the DAC repository from the list of connection type choices.
4. Enter the connection information as described in the Connection Information table in section Section 4.11.3.1, "How to Configure the DAC Server Repository Connection Using the DAC Client (Windows)".
5. Enter 2 to test the DAC repository connection.
6. Enter 5 to save changes.
7. Enter 6 to exit.

4.11.4 Enabling DAC Server Communication with Informatica PowerCenter

The DAC Server uses the following command line programs to communicate with Informatica PowerCenter:

- pmrep is used to communicate with PowerCenter Repository Services.
- pmcmd is used to communicate with PowerCenter Integration Services.

The DAC Server uses the pmcmd program to run the Informatica workflows using PowerCenter Integration Services. The pmrep and pmcmd programs are installed during the PowerCenter Services installation in the bin folder of the server directory. Oracle recommends that you co-locate the PowerCenter Services with the DAC Server so that the pmrep and pmcmd programs are available on the machine for the DAC Server to use. To enable the DAC Server to communicate with Informatica PowerCenter Services using the pmrep and pmcmd programs, you need to define the path of the Informatica Domain file 'domains.infa', and set appropriate environment variables as follows:

- If your DAC Server is installed on Windows, see Section 4.11.4.1, "How to Set Environment Variables on Windows".
- If your DAC Server is installed on UNIX or Linux, see Section 4.11.4.2, "How to Set Environment Variables on UNIX or Linux".

4.11.4.1 How to Set Environment Variables on Windows

To set the environment variables on Windows:

1. Locate the file domains.infa in the root Informatica PowerCenter installation directory and note down the directory path of this file.

   For example, D:\Informatica\PowerCenter8.1.1.

2. Create an environment variable called INFA_DOMAINS_FILE with the value set to the directory path to the domans.infa file, as follows:
   - On Windows, display the Windows Environment Variables dialog (that is, from the Windows Control Panel, choose System, then Advanced, then Environment Variables), and create a System variable with the following values:
     - Variable name: INFA_DOMAINS_FILE
     - Variable value: <<*directory path of domain file*>\domains.infa>

       The Variable value should include the domains.infa file name. For example, D:\Informatica\PowerCenter8.1.1\domains.infa.
3. Add the directory path to Informatica PowerCenter binaries to the PATH environment variable as follows:

   In the Windows System Properties > Environment Variables dialog, add the path of the Informatica \server\bin directory to the PATH environment variable. For example: <drive>:\Informatica\PowerCenter8.1.1\server\bin.

Verifying that the DAC Server is able to use pmrep and pmcmd

From a Windows command prompt, execute pmrep and then pmcmd. The test is successful if the pmrep and pmcmd prompts appear.

If pmrep and pmcmd are not recognized, then:

- Ensure Service Pack 4 has been applied to Informatica PowerCenter 8.1.1.
- Verify that the INFA_DOMAINS_FILE variable points to the domains.infa file located in the Informatica directory.
- Verify that the PATH variable includes the path to the Informatica binaries (\Informatica\PowerCenter\server\bin).

4.11.4.2 How to Set Environment Variables on UNIX or Linux

Use the dac_env.sh or dac_env.csh file to set the appropriate environment variables on a Unix or Linux machine. For more information on the DAC Server scripts, see Section A.6, "About the DAC Server Scripts for Bash and C Shells".

To set environment variables on a Unix or Linux machine:

1. Open the dac_env.sh file or dac_env.csh file for editing.

   The files are located in the /DAC directory.
2. Locate the OS section for the operating system on which your DAC Server is installed.
3. Uncomment out the line to set the library path environment variable, and replace %DAC_PMCMD_PATH% with the path to the PowerCenter Services binaries.

   For example, using the dac_env.sh file, if the DAC Server is on Solaris or Linux, in the SOLARIS\LINUX section, change the following line from:

   ```
   #export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:%DAC_PMCMD_PATH%
   ```

   To:

   ```
   export LD_LIBRARY_PATH=${LD_LIBRARY_
   PATH}:/home/Informatica/PowerCenter8.1.1/server/bin
   ```

4. In the Common section for Informatica Configuration, set the environment variables for PATH, PM_HOME, INFA_DOMAINS_FILE as follows:

   a. Replace %INFORMATICA_SERVER_LOCATION% with the path to Informatica PowerCenter Services installation.

   b. Replace %DAC_PMCMD_PATH% with the path to Informatica PowerCenter Services binaries.

   c. Replace %DOMAINS.INFA_FILE_LOCATION% with the path to the domains.infa file.

      Include the name of the file. For example, /home/Informatica/PowerCenter8.1.1/domains.infa.

   For example, using the dac_env.sh file, change the following lines in the common section from:

   ```
   export PATH=${PATH}:%DAC_PMCMD_PATH%:%INFORMATICA_SERVER_LOCATION%:.
   export PM_HOME=%INFORMATICA_SERVER_LOCATION%
   export INFA_DOMAINS_FILE=%DOMAINS.INFA_FILE_LOCATION%
   ```

   To:

   ```
   export PATH=${PATH}:/home/Informatica/PowerCenter8.1.1/server/bin
   export PM_HOME=/home/Informatica/PowerCenter8.1.1/server
   export PM_HOME=/home/Informatica/PowerCenter8.1.1/server
   export INFA_DOMAINS_FILE= /home/Informatica/PowerCenter8.1.1/domains.infa
   #export LANG=C (uncomment if required)
   #export LC_ALL=C (uncomment if required)
   ```

5. If required, edit the dac_env.sh file and change the source command from:

   ```
   source ./dac_env.sh
   ```

   To:

   ```
   . ./dac_env.sh
   ```

Verifying that the DAC Server on a Unix or Linux machine is able to use pmrep and pmcmd

Invoke config.sh or config.csh to set environment. For example, . ./config.sh. Then, invoke pmcmd.

Make sure that invoking pmcmd starts the pmcmd shell. If you get a 'command not found' error, then the location of the PowerCenter Services is not properly added to PATH in dac_env.sh or dac_env.csh. Review all environment variable settings to ensure they are correctly set.

Then, invoke pmrep, and make sure that invoking pmrep starts the pmrep shell. If you get a 'command not found' error, then the location of the PowerCenter Services is not properly added to PATH in dac_env.sh or dac_env.csh. Review all environment variable settings to ensure they are correctly set.

On some shells, export commands in dac_env.sh and config.sh might not work correctly. In this case, try breaking the commands in two. For example, from:

```
export JAVA_HOME=/opt/java1.5
```

To:

```
JAVA_HOME=/opt/java1.5
```

```
export JAVA_HOME
```

4.11.4.3 How to Verify Java JDK Availability and Version

To verify that the DAC Server on uses the correct Java JDK:

1. Invoke config.sh or config.csh to set environment.

   For example:

   ```
   . ./config.sh
   ```

2. Verify Java availability and version by typing the following command:

   ```
   $JAVA -version
   ```

   For example, the following information is returned:

   ```
   java version "1.5.0_06"
   Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_06-b05)
   Java HotSpot(TM) Server VM (build 1.5.0_06-b05, mixed mode)
   ```

   If you receive a 'command not found' error message, or the Java version is lower than 1.5, then the JAVA_HOME parameter in config.sh or config.csh is pointing to a non-existent or incorrect Java JDK location.

4.11.5 Starting the DAC Server

On Windows, start the DAC Server by double-clicking the \DAC\startserver.bat script.

On UNIX or Linux, start the DAC Server by executing either startserver.sh or startserver.csh.

For more information on starting and stopping the DAC Server on Windows, UNIX or Linux, see Section A.2, "How to Start and Stop the DAC Server".

4.12 Configuring the DAC Settings

This section explains how to configure DAC settings, and contains the following topics:

- Section 4.12.1, "How to Set DAC System Properties"
- Section 4.12.2, "How to Register the Informatica Services in the DAC"
- Section 4.12.3, "How to Set Physical Data Sources"

You must start the DAC Server to complete certain steps in the procedures described below. For information on how to start the DAC Server, see Section A.2, "How to Start and Stop the DAC Server".

4.12.1 How to Set DAC System Properties

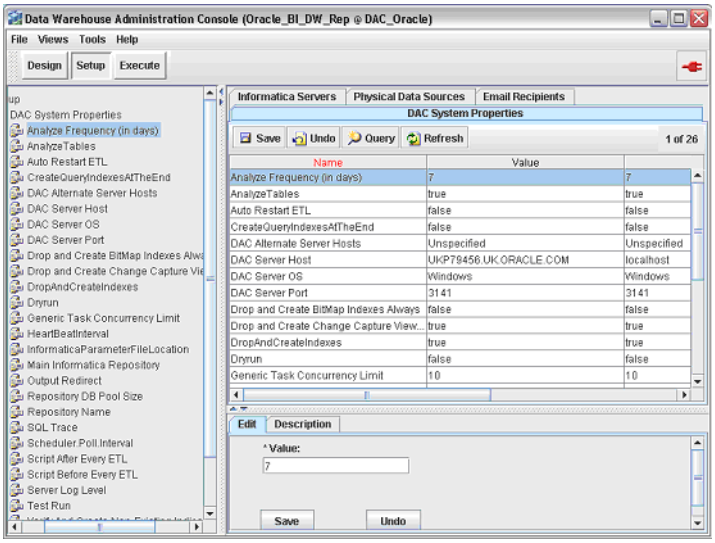
This section describes the DAC System Properties to set to ensure proper integration between the DAC Client, the DAC Server and Informatica.

To set DAC System Properties:

1. In the DAC Client, display the Setup view.

For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client").

2. Display the DAC System Properties tab.



3. Set values for the following properties:

Note: For the full list of DAC System Properties and their descriptions, see Section A.7.1, "How to Set Up DAC System Properties".

4.12.2 How to Register the Informatica Services in the DAC

This section explains how to register the Informatica PowerCenter Integration Services service and the Informatica PowerCenter Repository Service in the DAC Client.

When you register Informatica Services in the DAC Client, note the following:

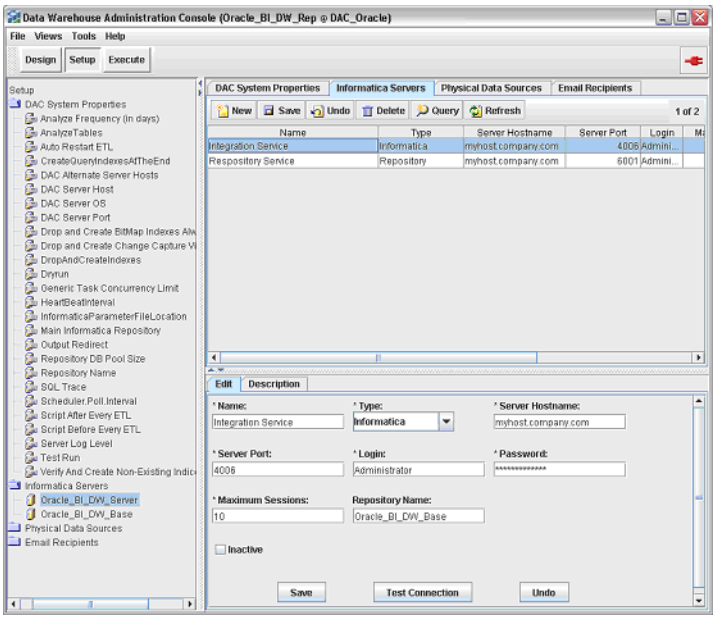
- You must register one or more Integration Services service in the DAC Client.
- You must register one Repository Service in the DAC Client.

> **Tip:** For details of the PowerCenter Services that you need to register, log into the Informatica PowerCenter Administration Console (for more information, see Section A.3, "How to Log Into Informatica PowerCenter Administration Console").

To register Informatica Services in the DAC Client:

1. In the DAC Client, display the Setup view, then display the Informatica Servers tab.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client").



2. For the Informatica Integration Service that you want to register, do the following:

   a. Register the Informatica Integration Services service, by modifying the record with Name = Oracle_BI_DW_Server and entering the following information in the Edit tab:

Field	Enter
Name	Logical name for the Integration Service (for example, Integration_Service).
Type	Choose Informatica.
Server Hostname	Integration Service host name or IP address. For example, myhost.company.com.
Server Port	The port that the DAC uses to communicate with the Integration Services service (for example, 4006). Specify the port number that you configure for the Integration Services service to listen on when setting the ServerPort custom property in Section 4.7.3, "Setting PowerCenter Integration Services Custom Properties".
Login	Informatica Repository user name who has appropriate privileges to execute workflows (for example, Administrator).
Password	Informatica Repository user password.
Maximum Sessions	Maximum number of workflows that can be executed in parallel on the Informatica PowerCenter Integration Services service. If the number of sessions is zero or is not specified, the DAC Server assigns the default value of 10.
Repository Name	Enter the name of the Oracle BI Applications repository that you restored in Section 4.6, "Restoring the Oracle Business Intelligence Pre-Built Repository". For example, Oracle_BI_DW_Base (or Oracle_BI_DW_Teradata for Teradata installations).
Inactive	Indicates whether the Informatica PowerCenter Integration Services service is active or inactive.

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

Note: Integration Services must be running (for more information about starting Integration Services, see Section A.4, "How to Start and Stop Informatica Services").

c. Click Save to save the details.

3. Register the Informatica Repository Service, by modifying the record with Name = Informatica_REP_Server and entering the following information:

Field	Enter
Name	Logical name for the Repository Service (for example, Repository_Service).
Type	Choose Repository.
Server Hostname	Repository Service host name or IP address.
Server Port	Port where the Repository Service connects for requests. (The default port number is 6001.)
Login	Informatica Repository user name that has appropriate privileges to execute workflows (for example, Administrator).
Password	Informatica Repository user password (for example, Administrator).
Maximum Sessions	Maximum number of workflows that can be executed in parallel on the Informatica PowerCenter Integration Services service. If the number of sessions is zero or is not specified, the DAC Server assigns the default value of 10.

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

5. Click Save to save the details.

4.12.3 How to Set Physical Data Sources

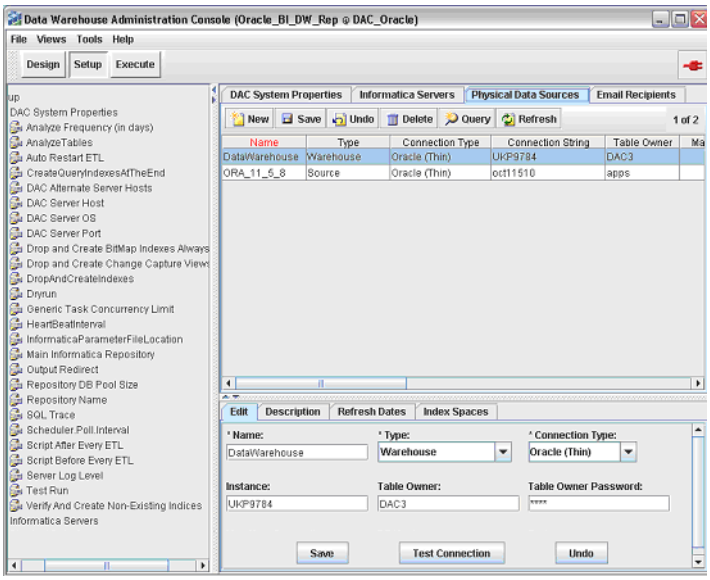
Follow this procedure to specify the transactional and data warehouse data sources in the DAC.

To specify transactional and data warehouse data sources

1. In the DAC Client, display the Setup view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client").

2. Display the Physical Data Sources tab.



The Physical Data Sources tab displays a pre-created record for the data warehouse with name DataWarehouse, and one or more records for the OLTP sources. The records that are created by DAC for the OLTP sources depend on the business application source systems you selected when importing the DAC metadata (for more information, see Section 4.9, "Importing the DAC Metadata").

3. For each record, enter the following information in the Edit sub-tab:

Field	Description
Name	Logical name for the OLAP or OLTP database connection. Do not change the default values. For example, you must specify DataWarehouse as the name of the OLAP data source. **Note**: When you create an Execution Plan to perform an ETL, you need to use the OLAP and OLTP data source names that you specify here as the Parameter values for the Execution Plan parameters DBConnection_OLTP and DBConnection_OLAP. For more information about setting the Parameter values for an Execution Plan, see *Oracle Business Intelligence Data Warehouse Administration Console Guide*.
Type	▪ 'Source' for the database connection for a transactional (OLTP) database. ▪ 'Warehouse' for the database connection for a data warehouse (OLAP) database. Do not change the default values.
Connection Type	Type of database. Possible values are: ▪ Oracle (OCI8) ▪ Oracle (Thin) ▪ DB2 ▪ DB2-390 ▪ MSSQL ▪ Teradata ▪ Flat File Select the Type based on your database.

Field	Description
Instance or TNS Name or Connection String or Database Name (NOTE: The field name changes based on the Connection Type selection.)	Enter the value appropriate for your database. For an Oracle database instance, enter the database SID (that is the SERVICE_NAME = value in the tnsnames.ora file in \network\admin\. For an Oracle TNS Name, enter the TNS entry name that is specified in the tnsnames.ora file in \network\admin\.
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	Unique number assigned to the data source category so that the data can be identified in the data warehouse. For example, the value '1' is used for Siebel data sources. 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 R12 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 Section 7.1.2.1, "How to Configure Data Source Num IDs" 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.
Default Index Space	(Oracle specific) Specifies the table space in which the DAC drops and creates indices against this database connection To use the default table space, leave this field blank.

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

4.13 Configuring Relational Connections In Informatica Workflow Manager

This section explains how to log into Informatica PowerCenter Workflow Manager and define relational connections, and contains the following topics:

- Section 4.13.1, "How to Log Into Informatica PowerCenter Workflow Manager"
- Section 4.13.2, "Configuring Relational Connections in Informatica PowerCenter Workflow Manager"

4.13.1 How to Log Into Informatica PowerCenter Workflow Manager

Note: Before you log into Workflow Manager, start the Informatica services (for more information, see Section A.4, "How to Start and Stop Informatica Services").

To log into Informatica Workflow Manager, do the following:

1. On the Informatica PowerCenter Client machine, start Informatica PowerCenter Workflow Manager.
2. Choose Repository, then Add, to display the Add Repository dialog.
3. At the Add Repository dialog, specify Oracle_BI_DW_Base (or the appropriate repository name) in the **Repository** field, and Administrator in the **Username** field, then click OK to save the details.
4. Choose Repository, then Connect, to display the Connect to Repository dialog.
5. Use the Password field to specify the Administrator password (for example, Administrator).
6. If the Connection Settings area is not displayed, click More.
7. Click Add (next to the **Domain** drop down list) to display the Add Domain dialog.
8. At the Add Domain Dialog, specify the name of the domain that was created when you installed Informatica PowerCenter Services (for example, Domain_<*hostname*>), and the fully qualified hostname for the gateway host (for example, mymachine@us.company.com) and port for the gateway port (for example, 6001).
9. Click OK to save the details and close the Add Domain dialog.
10. Click Connect.

4.13.2 Configuring Relational Connections in Informatica PowerCenter Workflow Manager

To configure the relational connections, you need to do the following:

1. Use Informatica PowerCenter Workflow Manager to configure the relational connections, as follows:
   - For all databases, follow the steps in Section 4.13.2.1, "How to configure the relational connections".
   - Additionally, for Teradata databases, follow the steps in Section 4.13.2.2, "(Teradata specific) How to configure the Informatica Repository for Teradata external loader connections".

Note: The Informatica services must be running to perform these tasks. To start the Informatica services, see Section A.4, "How to Start and Stop Informatica Services".

4.13.2.1 How to configure the relational connections

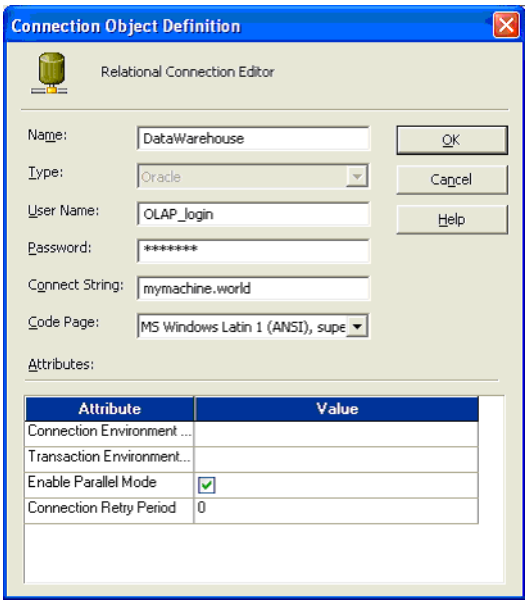
Follow this procedure to configure relational connections:

1. In Informatica PowerCenter Workflow Manager, choose Connections, then Relational to display the Relational Connection Browser.

   You need to create a connection for each transactional (OLTP) database, and a connection for the Oracle Business Analytics Warehouse (OLAP) database.
2. For each database connection that you need to create, do the following:

a. Click New to display the Select Subtype dialog, select the appropriate database type (for example, Oracle), then click OK.

b. Click New to display the Connection Object Definition dialog, and enter the details.



c. Click OK to save the details.

Notes:

- If the target database is Oracle, DB2, or Teradata, use the settings as follows:
  - Click New and select the type as Oracle, DB2, then click OK.
  - Name: DataWarehouse, for connection to the Oracle Business Analytics Warehouse.

    You must specify 'DataWarehouse' exactly as it appears in the Physical Data Sources tab in the DAC Setup View. For the connection to the OLTP, specify the name exactly as it appears in the Physical Data Sources tab in the DAC Setup View. For example, if your source system is Siebel ver 7.8.x then you name this connection as SEBL_78. For more information, see Section 4.12.3, "How to Set Physical Data Sources".
  - User Name: Database user name.
  - Password: Database user password.
  - Connect string: Connect string for the database (refer to the Informatica Help for information about specifying this value).
  - Code Page: Code page compatible with the code page of the database client. If NLS_LANG (for Oracle database) or DB2CODPAGE (for DB2 database) has been set, then the Code Page value should be a code page compatible with the language set by these variables. For more information, see Section 4.3.3, "Code Page and Data Movement Requirements".
- 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, for the connection to the Oracle Business Analytics Warehouse. You must specify 'DataWarehouse' exactly as it appears in the Physical Data Sources tab in the DAC Setup View.
- For the connection to the OLTP, specify the name exactly as it appears in the Physical Data Sources tab in the DAC Setup View. (For example, if your source system is Siebel ver 7.8.x then you name this connection as SEBL_78). For more information, see Section 4.12.3, "How to Set Physical Data Sources".
- User Name: Database user name.
- Password: Database password.
- Connect string: Connect string for the connecting to the database (refer to the Informatica Help for information about specifying this value).
- Code Page: Code page compatible with the code page of the database client.

3. Click Close to close the Relational Connection Browser.

4.13.2.2 (Teradata specific) How to configure the Informatica Repository for Teradata external loader connections

Follow this procedure to configure the Informatica repository in Informatica PowerCenter Workflow Manager for Teradata external loader connections.

1. In Informatica PowerCenter Workflow Manager, choose Connections, then Loader to display the Loader Connection Browser.
2. In the Objects dialog, 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 Configuring the SiebelUnicodeDB Custom Property

If your source to target data movement configuration is Unicode to Unicode, you need to create a custom property called SiebelUnicodeDB on Integration Services. If your source to target data movement configuration is either Code Page to Code Page or Code Page to Unicode, you do not need to create this property. For more information on supported source to target configuration modes, and how to determine the source

to target configuration for your environment, seeSection 4.3.3, "Code Page and Data Movement Requirements".

To create and set the SiebelUnicodeDB custom property on Integration Services:

1. Log in to Informatica PowerCenter Administration Console. For information on how to log into PowerCenter Administration Console, see Section A.3, "How to Log Into Informatica PowerCenter Administration Console".
2. Select the Integration Service.
3. In the Properties tab, scroll down to the Custom Properties area, and click Edit.
4. In the Name field, enter the following:

   SiebelUnicodeDB
5. In the Value field, enter the following:

   [user_OLTP]@[connectString_OLTP] [user_OLAP]@[ConnectString_OLAP]

   Where:

   [user_OLTP] is the database user for the OLTP source database. It must match exactly the value you entered for the User Name field when creating the relational connection for the OLTP in Informatica Workflow Manager.

   [ConnectString_OLTP] is the connect string for the OLTP. It must match exactly the value you entered for the Connect String field when creating the relational connection for the OLTP in Informatica Workflow Manager.

   [user_OLAP] is the database user for the Oracle Business Analytics Warehouse database. It must match exactly the value you entered for the User Name field when creating the relational connection for the data warehouse in Informatica Workflow Manager.

   [ConnectString_OLAP] is the connect string for the data warehouse. It must match exactly the value you entered for the Connect String field when creating the relational connection for the data warehouse in Informatica Workflow Manager.

Note: Always leave a space between the strings for OLTP and OLAP. You must enter the user names and connection strings in the same case as you used for the relational connections in Informatica Workflow Manager.

For more information about Relational Connections, see Section 4.13.2, "Configuring Relational Connections in Informatica PowerCenter Workflow Manager".

4.15 Miscellaneous and Additional Configuration Tasks

This section provides configuration steps that are needed to complete configuration of the Oracle BI Applications environment. Not all the sections may be applicable for your environment. Complete the tasks based on your environment and requirements.

- Section 4.15.1, "How to Set the Code Page File Property for Unicode Environments on Teradata Databases"
- Section 4.15.2, "How to Set Up the HOSTS File for Teradata Installations on Windows"
- Section 4.15.3, "How to Update Siebel Transactional Database Schema Definitions"
- Section 4.15.4, "How to Create Stored Procedures for DB2-UDB"
- Section 4.15.5, "Teradata-Specific Installation and Configuration Checklist"

- Section 4.15.6, "How to Configure Email Recipients in the DAC Client"
- Section 4.15.7, "How to Configure Email in the DAC Server"
- Section 4.15.8, "How to Back Up and Restore the Informatica Repository (Optional)"
- Section 4.15.9, "How to Set DAC Source System Parameters"
- Section 4.15.10, "Migrating Seed Data from a Siebel Source System to the Oracle BI Repository"

4.15.1 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. In Informatica PowerCenter Workflow Manager, drag a Workflow into the Workflow Designer pane.
2. In the Workflow Designer pane, double click the Task (for example, SDE_PSFT_APTermsDimension) to display the Edit Tasks dialog.
3. Display the Mappings tab
4. Select Target in the left pane. In the Properties section, click on the **Set File** link.
5. In the Flat Files - Targets dialog, in the File Properties area, click Advanced.
6. In the Fixed Width Properties - Targets dialog, in the Code Page area, select UTF-8 encoding of Unicode.

4.15.2 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 PowerCenter Integration Services service 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 PowerCenter Integration Services service 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.

4.15.3 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 4.15.3.1, "How to Run the DDLIMP Tool From The Command Line"
- Section 4.15.3.2, "How to Apply Siebel (CRM) Schema Changes For Oracle, DB2/UDB, MSSQL"
- Section 4.15.3.3, "How to Apply CRM Schema Changes For DB2 on 390 or z/os"
- Section 4.15.3.4, "How to Verify the Siebel (CRM) Schema Changes"
- Section 4.15.3.5, "About Delete Triggers"

4.15.3.1 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_DSN> /GSSE_ROLE /F <..\OracleBI\dwrep\DDL_OLTP.CTL> /L <..\oracleBI\dwrep\DDL_OLTP.log>

NOTE: Where <ODBC_DSN> is the ODBC connection created as described in section Section 4.3.4.3, "Create ODBC Database Connections".

\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC_DSN> /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 OTTP 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.

4.15.3.2 How to Apply Siebel (CRM) 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) 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 Business Intelligence 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 Business Intelligence 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.

4.15.3.3 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.
   - %32kBufPool - Replace it with a 32K TBS Buffer Pool name.

2. On the machine where Oracle Business Intelligence 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

4.15.3.4 How to Verify the Siebel (CRM) Schema Changes

After applying Siebel (CRM) schema changes, you need to verify that appropriate tables were created in the transactional database.

To verify Siebel (CRM) 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

4.15.3.5 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.

4.15.4 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 4.15.4.2, "How to Create DB2 stored procedures").

4.15.4.1 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 4.15.4.2, "How to Create DB2 stored procedures").

4.15.4.2 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 4.15.4, "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.

4.15.5 Teradata-Specific Installation and Configuration Checklist

Table 4–4 provides a list of Teradata-specific installation and configuration steps that are performed during the Oracle Business Intelligence 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 4–4 Teradata-Specific Installation Checklist for Windows

Action	Link to Topic
Review Teradata-specific database guidelines.	Section 3.7, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse".
Create or drop data warehouse tables.	Section 4.10.2, "How to Create Data Warehouse Tables on a Teradata Database".
Create or restore the Informatica repository using the Informatica Repository Manager. You cannot create or restore the Informatica repository using the DAC.	Section 4.6, "Restoring the Oracle Business Intelligence Pre-Built Repository".
Configure the source and data warehouse database relational connections in Informatica PowerCenter Workflow Manager. You cannot use the DAC's Data Warehouse Configurator to configure relational connections.	Section 3.7, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse".
Set the Code Page File Property for Unicode Environments.	Section 4.15.1, "How to Set the Code Page File Property for Unicode Environments on Teradata Databases".
Set up the HOSTS file.	Section 4.15.2, "How to Set Up the HOSTS File for Teradata Installations on Windows".

4.15.6 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. In the DAC Client, display the Setup view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client").

2. Display 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: ■ 10 -- Notifies recipient of success or failure of each task. ■ 5 -- Notifies recipient of success of failure of the entire ETL process. ■ 1 -- Notifies recipient that ETL completed successfully.
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).

4.15.7 How to Configure Email in the DAC Server

Follow this procedure to configure the email administrator account In the DAC Client, 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.

> **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. In the DAC Client, choose Tools, then DAC Server Management, then DAC Server Setup.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

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

2. Click Yes.

   The Server Configuration dialog appears.

3. Display the Email Configuration tab, and enter the email details for the email address to which you want to send DAC server information emails.

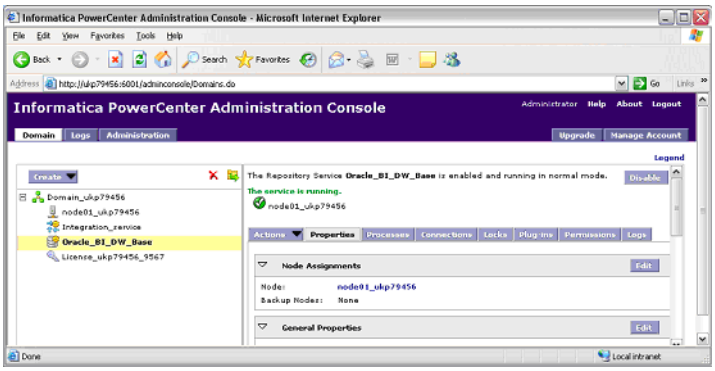
4. Click Save.

Once the configuration has been completed, you can start the DAC Server.

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

To stop and start the Informatica repository

1. In Informatica PowerCenter Administration Console, in the Domain tab, select the Repository Service (for example, Oracle_BI_DW_Base).



2. Do one of the following:

   - To back up the Informatica Repository, choose Actions, then Back Up Contents.
   - To restore the Informatica Repository, choose Actions, then Restore Contents. For detailed information on procedures to Back up and restore repositories, refer to the Informatica Repository Guide.

4.15.9 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.

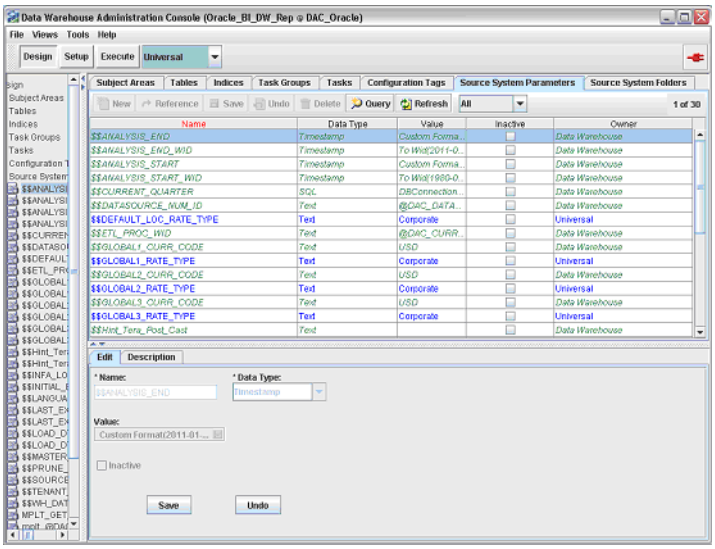
> **Note:** The following preferences are applied to all tasks within a container. If extracting data from more than one source, these preferences will need to be re-applied to each associated container. They can be overridden at the task level by adding the parameter to the specific task and assigning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

To set DAC source system parameters

1. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client").

2. Make sure that you have selected the correct container from the *containers* drop down list. You cannot edit out-of-the box containers. Make a copy of an existing container in order to make edits.
3. Display the Source System Parameters tab.



Note: For information about recommended settings for specific databases, see Chapter 3, "Pre-installation and Predeployment Requirements for Oracle Business Intelligence Applications".

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

4.15.10 Migrating Seed Data from a 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.

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.

4.16 About Running A Full Load ETL

After you have installed and configured Oracle Business Intelligence Applications, your Oracle Business Analytics Warehouse (OLAP) database is empty. You need to perform a full load ETL to populate your Oracle Business Analytics Warehouse.

The ETL processes for Oracle Business Intelligence Applications are created and managed in the DAC. For detailed information about running ETLs In the DAC Client, see *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

For an example of using the DAC to run a Full Load ETL, see Section 4.16.1, "An Example of Running a Full Load ETL".

4.16.1 An Example of Running a Full Load ETL

This section uses an example to show you how to get started quickly with running a full load ETL. In this example, you have installed Oracle Financial Analytics with an Oracle EBS OLTP data source, and you want to load OLTP data for the subject area Receivables, as follows:

1. In the DAC Client, choose File, then New Source System Container, to display the New Source System Container dialog, and specify details of the new container.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client").

2. Select the **Create as a Copy of Existing Container** radio button, and select the appropriate container (for example, Oracle 11.5.10) from the **Existing Containers** drop down list, then click OK.
3. In the Design view, display the Source System Parameters tab and set the parameters appropriately.
4. In the Setup view, display the DAC System Parameters tab and set the properties appropriately.
5. Display the Execute view, and display the Execution Plan tab.
6. Click New, display the Edit tab, and use the **Name** field to specify a name for the ETL process.
7. Display the Subject Areas tab and click Add/Remove to display the Choose Subject Areas dialog.
8. Select the new container that you created in step 4 from the container drop down list at the top of the Choose Subject Areas dialog.
9. Select Financials - Receivables, click Add, then click OK.
10. Display the Parameters tab, and click Generate.
11. On the Parameters tab, edit the parameters as follows:
    - Edit the value of DBConnection_OLAP and set it to the same value as the name of the OLAP database that you specified in the Physical Data Source dialog (for example, DataWarehouse).

- Edit the value of DBConnection_OLTP and set it to the same value as the name of the OLTP database that you specified in the Physical Data Source dialog (for example, ORA_11_5_8).
- If there is a FlatFileConnection parameter, edit the value of FlatFileConnection and set it to the same value as the name of the flat file data source that is specified in the Physical Data Source dialog (for example, ORA_11_5_8_Flatfile).

12. On the Execution Plans tab, click Build.
13. On the Execution Plans tab, click Run Now.

    The DAC will perform a full load for Financials - Receivables.
14. Use the Current Run tab to check the status of the ETL run.

    If the ETL run was successful, you will see 'Success' in the **Run Status** field, and the **End Timestamp** value will be set to the time and date when the ETL was completed.

    If the ETL run fails, use the information in the Description tab and the Audit Trail tab to diagnose the error that caused the failure.

    Tip: If an ETL run fails, you cannot re-run the ETL until the failed ETL has been cleared from the Current Run tab. To clear an ETL from the Current Run tab, right click on the ETL and select Mark As Completed.

Part III

Customization and Security in Oracle Business Intelligence Applications

Part III explains how to customize the out-of-the-box functionality in Oracle Business Intelligence Applications, and contains the following chapters:

- Chapter 5, "Customizing the Oracle Business Analytics Warehouse"
- Chapter 6, "Integrated Security for Oracle Business Intelligence Applications"

Note: For a high level road map for installation, configuration, and customization steps for Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications".

5

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 5.1, "Oracle Business Analytics Warehouse Customization Scenarios"
- Section 5.2, "Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables"
- Section 5.3, "Other Types of Customizations Requiring Special Handling"
- Section 5.4, "Category 2 Customizations: Adding Additional Tables"
- Section 5.5, "Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table"
- Section 5.6, "Configuring Extracts"
- Section 5.7, "Configuring Loads"
- Section 5.8, "Configuring Slowly Changing Dimensions"
- Section 5.9, "About Stored Lookups"
- Section 5.10, "Codes Lookup"
- Section 5.11, "About Resolving Dimension Keys"
- Section 5.12, "About Domain Values"
- Section 5.13, "Configuring the Domain Value Set with CSV Worksheet Files"
- Section 5.14, "Configuring the Domain Value Set Using Informatica PowerCenter Designer"
- Section 5.15, "Configuring Conformed Dimensions"

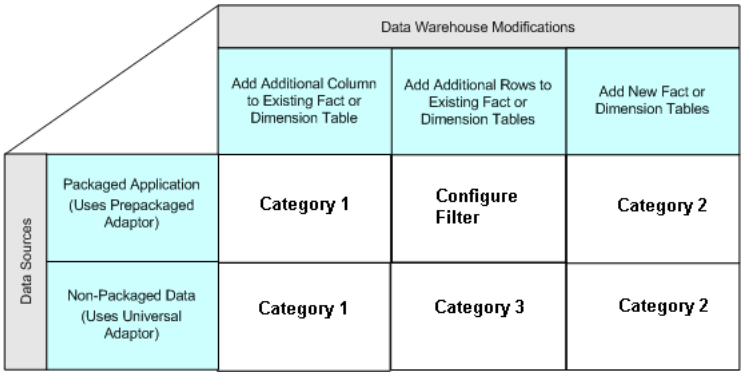
5.1 Oracle Business Analytics Warehouse Customization Scenarios

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 5–1 Supported customizations based on data source



For detailed information about tables and naming conventions, see *Oracle Business Analytics Warehouse Data Model Reference*.

5.1.1 Types of Customizations

Figure 5–1, "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.

5.1.2 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.

5.2 Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables

This section includes the following topics:

- Section 5.2.1, "About Extending Mappings"
- Section 5.2.2, "Impact of Customization on Upgrade"
- Section 5.2.3, "Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse"
- Section 5.2.4, "Scenario for Extending the Oracle Business Analytics Warehouse"

5.2.1 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 5.5, "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.

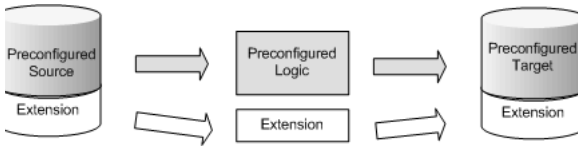
Oracle BI Applications recognizes two types of customization: extension and modification. The supported extension logic allows you to add to existing objects. For example, you can extract additional columns from a source, pass them through existing mappings, and populate new columns added to an existing 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 5–2 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.

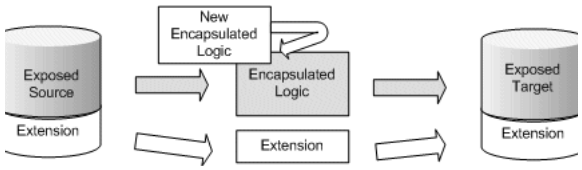
Note: Targets should not be added to a mapping.

5.2.2 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.

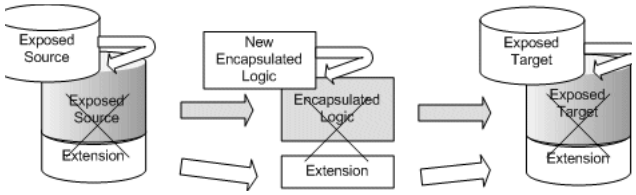
By encapsulating the logic as much as possible, any changes made to the preconfigured logic can be switched as either part of a patch release or upgrade without impacting any extension logic, as shown in the following figure.

Figure 5–3 Encapsulating logic



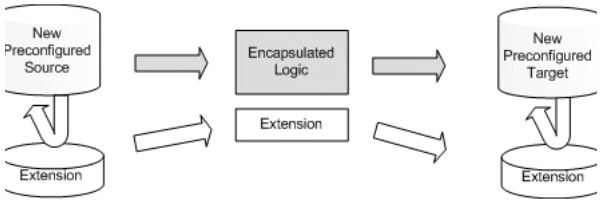
If there is a change to an exposed object, the new logic will always take precedence over the extension logic. However, rather than losing all of the extensions, much of the extension logic is retained and only has to be reapplied to the exposed objects. For example, if you add an additional column from 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 5–4 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 5–5 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.

5.2.2.1 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.

5.2.3 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.

5.2.4 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 5.2.4.1, "Tips for Modifying the SQL Override in a Source Qualifier"
- Section 5.2.4.2, "Example of Extracting Data from a Siebel Source"
- Section 5.2.4.3, "Including a Source Table for the Change Capture Process"
- Section 5.2.4.4, "Example of Extracting Data from a Non-Siebel Packaged Source"
- Section 5.2.4.5, "Example of Extracting Data from a Universal Source"
- Section 5.2.4.6, "Example of Loading Data into an Existing Target Table"
- Section 5.2.4.7, "Updating the DAC"

5.2.4.1 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 5.2.4.2, "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.

5.2.4.2 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_Adaptor.

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,
0 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 Informatica PowerCenter Workflow Manager.

    This is necessary because it is possible that changes made to the mapping may invalidate the session.

5.2.4.3 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.

5.2.4.4 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 run 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_Adaptor.
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 Informatica PowerCenter Workflow Manager.

    This is necessary because it is possible that changes made to the mapping may invalidate the session.

5.2.4.5 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_Adaptor.
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 Informatica PowerCenter Workflow Manager.

    This is necessary because it is possible that changes made to the mapping may invalidate the session.

5.2.4.6 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 Informatica PowerCenter Workflow Manager.

    This is necessary because it is possible that changes made to the mapping may invalidate the session.

5.2.4.7 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 Client, see the Oracle Business Intelligence Data Warehouse Administration Console Guide.

5.3 Other Types of Customizations Requiring Special Handling

This section includes the following topics:

- Section 5.3.1, "Modifying Category 2 SCD Triggers"
- Section 5.3.2, "Adding Dimensions to Existing Facts"
- Section 5.3.3, "Adding Date Dimensions to Existing Facts"
- Section 5.3.4, "Adding Currencies to an Existing Table"

5.3.1 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')))))
```

5.3.2 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 in _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 5–1 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 5–1 Formats to Derive INTEGRATION_ID

Dimension	Foreign Key	When Source is Oracle Application	When Source is Siebel Application
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 5–1 (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_ CONTCT_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.

5.3.3 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.

5.3.4 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 mapplet 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.

5.4 Category 2 Customizations: Adding Additional Tables

This section includes the following topics:

- Section 5.4.1, "About Creating New Dimension or Fact Tables"
- Section 5.4.2, "Using Custom Folders in the Oracle Business Analytics Warehouse"
- Section 5.4.3, "Creating Custom Informatica Workflows"
- Section 5.4.4, "Important Notes About Customizing the Oracle Business Analytics Warehouse"
- Section 5.4.5, "Including a Source Table for Change Capture Process"
- Section 5.4.6, "Adding a New Dimension in the Oracle Business Analytics Warehouse"
- Section 5.4.7, "Adding a New Fact Table in the Oracle Business Analytics Warehouse"
- Section 5.4.8, "Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse"

5.4.1 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.

5.4.1.1 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.

5.4.1.2 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.

5.4.2 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 PowerCenter 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.

5.4.3 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 PowerCenter Designer.
- Set the appropriate source and target connection values in Informatica PowerCenter 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 Client, then in the Properties tab of Informatica PowerCenter 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 Client, 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 Client, with the appropriate properties set. Also, you must register In the DAC Client, source and target tables, task definitions, and dependencies.

5.4.4 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.

5.4.4.1 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 specified 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 Client, 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 mapplet 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.

5.4.5 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 Client, 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.

5.4.6 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.

5.4.7 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 5.4.5, "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 Client, 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.

5.4.8 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.

5.5 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.

5.6 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 5.6.1, "Extracting Additional Data"
- Section 5.6.2, "Setting Up the Delimiter for Source Files"

5.6.1 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.

5.6.1.1 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.

5.6.1.2 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 Informatica 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 Mapplet 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.

5.6.2 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.

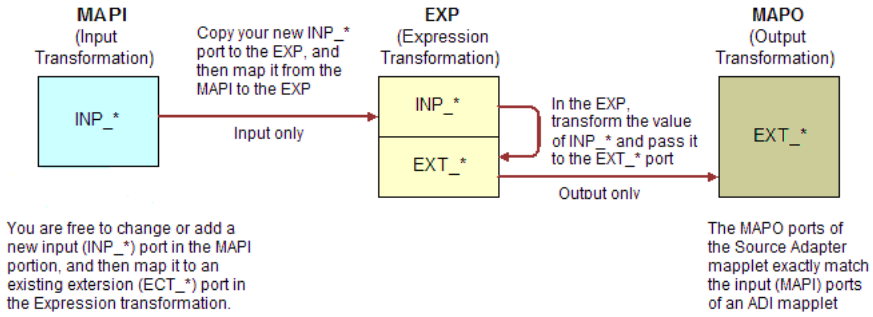
   You can identify an enclosing character by identifying a character that is not present in the source data. Common enclosing characters include single quotes and double quotes.
3. Save and close the CSV file.
4. Identify all the source definitions associated with the modified files.
5. Change the properties for each of these source definitions to use the enclosing character.
6. Validate and save your changes to the repository.

5.6.3 Configuring a Source Adapter Mapplet

The majority of all source-specific transformations occur in the Source Adapter mapplet; source-independent transformations generally take place in the Analytic Data Interface (load mapping). The Source Adapter mapplet 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 mapplet that allow transformations of data to occur. The three components are Mapplet Input (MAPI), Expression transformation (EXP), and Mapplet Output (MAPO).

Figure 5–6 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 Informatica 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.

5.7 Configuring Loads

The Oracle Business Analytics Warehouse prepackages load mappings for every data warehouse table.

5.7.1 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.

5.7.2 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 5–7 Extract and load mappings

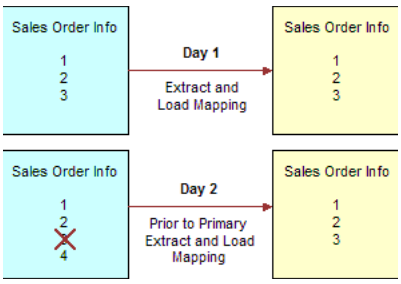


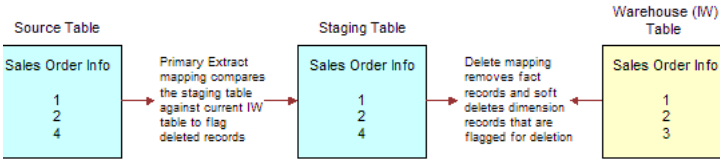
Figure 5–8 shows the primary extract and delete process that occurs when day two's information is extracted and loaded into the Oracle Business Analytics Warehouse from the source. The initial extract brings record four into the Oracle Business Analytics Warehouse. Then, using a primary extract mapping, 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 5–8. When the extract and load mappings run, the new sales order is added to the warehouse.

Figure 5–8 Primary Extract and Delete Mappings



5.7.3 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.

5.7.3.1 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 5.7.3, "About Working with Primary Extract and Delete Mappings".

5.7.3.2 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. In the DAC Client, display the Design view, and select the appropriate container from the drop down list.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Display the 'Tasks' tab.
3. Query for all tasks containing the string 'Delete' or 'Primary'.
4. Uncheck the 'Inactive' check box for those tasks.

5.8 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 Business Intelligence 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.

The table below shows the dimensions that are required using the SCD Type II update.

Table 5–2 Dimensions that are required using the SCD Type II update

Dimension	Adaptor
W_CUSTOMER_FIN_PROFL_D	PeopleSoft
W_EMPLOYEE_D	Oracle E-Business Suite and PeopleSoft
W_INT_ORG_D	PeopleSoft
W_INVENTORY_PRODUCT_D	Oracle E-Business Suite
W_POSITION_D	All
W_PRODUCT_D	Oracle E-Business Suite and Siebel

Note: If you want to turn off the TYPE2_FLG for any of the dimensions listed above, you will need to customize the source dependent extract mapping. The following changes should be made:

1. Override the SQL statement in the source qualifier to filter out the historical records.

   The change will vary depending on the OLTP data model. For example, you can select only records effective from the Informatica parameter $$$SessStartTime.

2. Disconnect the ports that bring the source effective from and to dates.

To turn a dimension to Type II SCD update enabled

1. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Select the appropriate container for your OLTP source system.
3. Display the Tasks tab.
4. Query for the SIL task that populating the dimension.
5. Display the Parameter child tab, and set the value for $$TYPE2_FLG to Y.

5.8.1 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.

5.8.1.1 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.

5.8.2 Type I and Type II Slowly Changing Dimensions

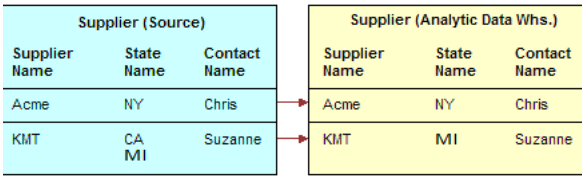
After you have correctly identified your significant and insignificant attributes, you can configure the Oracle Business Analytics Warehouse based on the type of slowly changing dimension (SCD) that best fits your needs—Type I or Type II.

5.8.2.1 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 5–9 An example Type 1 Slowly Changing Dimension



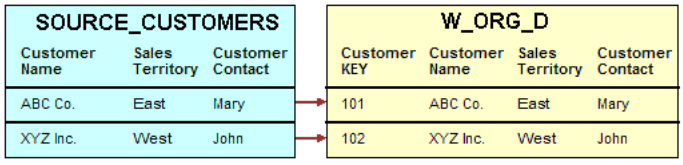
5.8.2.2 Type II Slowly Changing Dimension

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 5–10 An example Type 2 Slowly Changing Dimension

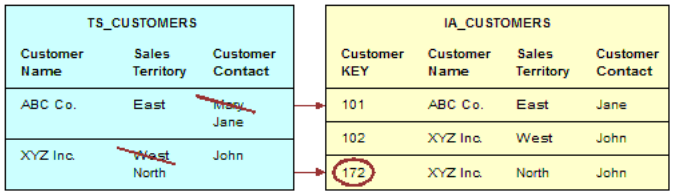


However, this data is not static; the next time a data extract shows a change for your customers in W_ORG_D, the records must change. This situation occurs when slowly changing dimensions are invoked. The figure below shows that records for the two customers, ABC Co., and XYZ inc. have changed when compared 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 5–11 An example Type 2 Slowly Changing Dimension



5.8.2.3 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 5–3.

Table 5–3 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 5–4.

Table 5–4 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 mapplet.

5.9 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 5.10, "Codes Lookup".
- Dimension Key Lookups. For more information, see Section 5.11, "About Resolving Dimension Keys".

5.10 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:

5.10.1 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.

5.10.2 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.

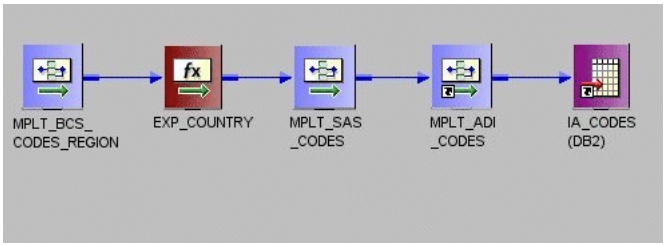
To understand how codes mappings function, it is helpful to first understand the columns within `W_CODES`. Table 5–5 describes these columns.

Table 5–5 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
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 Informatica PowerCenter Designer.

Figure 5–12 Example of a code mapping in Informatica PowerCenter Designer



5.10.3 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.

5.10.4 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 Informatica 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.

5.11 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.

5.11.1 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 5–6.

Table 5–6 Columns Used in the load mapping Dimension Key Lookup

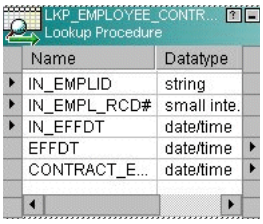
Port	Description
INTEGRATION ID	Uniquely identifies the dimension entity within its source system. Formed from the transaction in the Source Adapter of the fact table.
DATASOURCE_NUM_ID	Unique identifier of the source system instance.
Lookup Date	The primary date of the transaction; for example, receipt 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 5–13 Lookup for Employee Contract Data



5.12 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 made 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.

5.12.1 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.

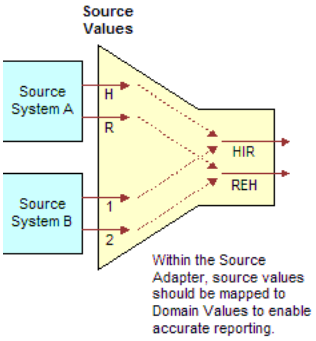
5.12.1.1 Preparing to Define the Rules

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 5.14, "Configuring the Domain Value Set Using Informatica PowerCenter Designer".

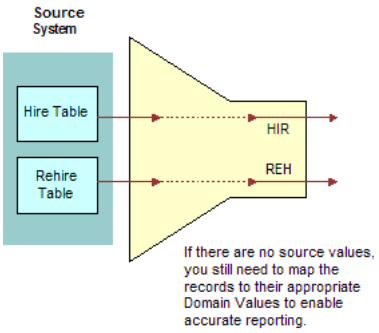
Figure 5–14 illustrates how the source values are converted to the domain values—HIR and REH.

Figure 5–14 Source Values Translated to Domain Values



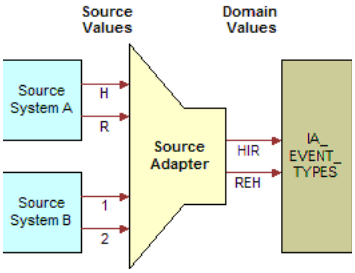
The figure below illustrates a different situation where the records may not contain a source value that flags the record as Hire or Rehire. In this case, the source system stores hires in one table and rehires in another table. To make this work, one possible solution is to modify the extract mappings 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 5–15 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 5–16 HIR and REH Values Populating the W_EVENT_TYPES table



5.12.2 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:

5.12.2.1 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)
```

5.12.2.2 Re-hires 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
```

5.12.2.3 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)
```

5.12.2.4 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)
```

5.12.2.5 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)
```

5.12.2.6 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)
```

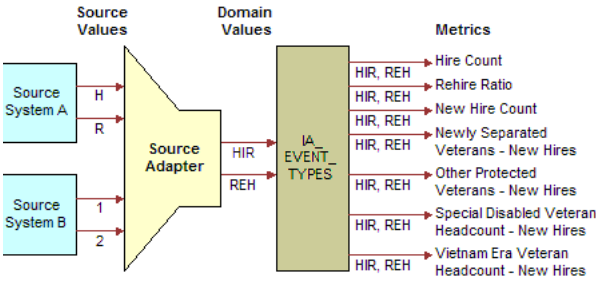
5.12.2.7 Vietnam Era Veteran Head count - New Hires

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 5–17 Metric Values From HIR and REH Domain Values



5.12.3 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.

5.13 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 Informatica PowerCenter Designer. For more information on configuring domain values using Informatica PowerCenter Designer, see Section 5.14, "Configuring the Domain Value Set Using Informatica 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 `$pmserver\server\infa_shared\LkpFiles` folder (for example, `D:\Informatica\PowerCenter8.1.1\server\infa_shared\Lkpfiles`).

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.

5.14 Configuring the Domain Value Set Using Informatica 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 Informatica PowerCenter Designer. For more information on configuring the domain value set with CSV worksheet files, see Section 5.13, "Configuring the Domain Value Set with CSV Worksheet Files".

Configuring the domain value set for a particular column, using Informatica 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 Informatica 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 Informatica 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.

5.15 Configuring Conformed Dimensions

This section provides procedures on configuring objects that apply to more than one application and contains the following topics:

- Section 5.15.1, "Configuring Conformed Dimensions for Universal Adapter"

5.15.1 Configuring Conformed Dimensions for Universal Adapter

This section provides configuration procedures for modifying dimensions that are loaded using Universal Adapter.

5.15.1.1 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 Informatica 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.

6

Integrated Security for Oracle Business Intelligence Applications

This chapter describes the security features in Oracle Business Intelligence Applications, and includes the following main topics:

- Section 6.1, "Types of Security in Oracle Business Intelligence Applications"
- Section 6.2, "Integrating Data Security with Oracle E-Business Suite"
- Section 6.3, "Integrating Data Security with Oracle's PeopleSoft Enterprise Applications"
- Section 6.4, "Metadata Object-Level Security in Oracle Business Intelligence Applications"
- Section 6.5, "Integrating Data Security with the Siebel CRM Application"
- Section 6.6, "Data-Level Security In Oracle Business Intelligence Applications"

6.1 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 Business Intelligence 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.

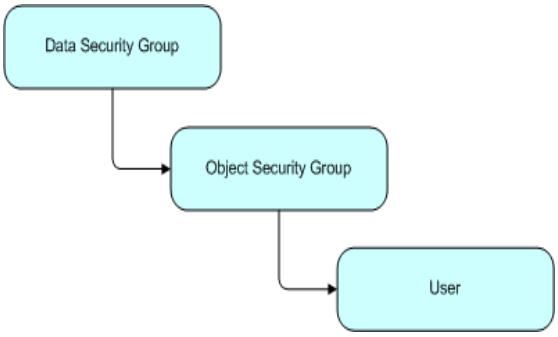
Security in Oracle Business Intelligence Applications can be classified broadly into three different categories:

- 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 Business Intelligence 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 Business Intelligence 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 Business Intelligence 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 Business Intelligence 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 6.2, "Integrating Data Security with Oracle E-Business Suite" and Section 6.3, "Integrating Data Security with Oracle's PeopleSoft Enterprise Applications".

6.1.1 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.

6.1.2 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 from 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.

6.1.3 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.

6.2 Integrating Data Security with Oracle E-Business Suite

This section explains how security in Oracle Business Intelligence 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 6.2.1, "Authorization"
- Section 6.2.2, "Operating Unit-Based Security With Oracle EBS"
- Section 6.2.3, "Inventory Organization-Based Security With Oracle EBS"
- Section 6.2.4, "Company Org-based Security Against Oracle EBS"
- Section 6.2.5, "Ledger Security Against Oracle EBS"
- Section 6.2.6, "Business Group Org-based Security Against Oracle EBS"
- Section 6.2.7, "HR Org-based Security Against Oracle EBS"
- Section 6.2.8, "Human Resource Personnel Data Analyst Security Against Oracle EBS"
- Section 6.2.9, "Employee Based Security with Oracle EBS"

6.2.1 Authorization

The authorization process of Oracle Business Intelligence Applications fetches a user's responsibilities from source Oracle EBS applications, matches them with all Oracle Business Intelligence 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 Business Intelligence Applications supports the following types of data security integrating with Oracle EBS:

- Operating Unit-Based security against Oracle EBS (for more information, see Section 6.2.2, "Operating Unit-Based Security With Oracle EBS").
- Inventory Organization-based security against Oracle EBS (for more information, see Section 6.2.3, "Inventory Organization-Based Security With Oracle EBS").
- Company Org-based security against Oracle EBS (for more information, see Section 6.2.4, "Company Org-based Security Against Oracle EBS").
- Ledger based security (for more information, see Section 6.2.4, "Company Org-based Security Against Oracle EBS").
- Business Group Org-based security against Oracle EBS (for more information, see Section 6.2.6, "Business Group Org-based Security Against Oracle EBS").
- Primary Employee/Position Hierarchy-based Security against Oracle EBS - HRMS application (for more information see Section 6.5.1, "Primary Position based Security").

Oracle Business Intelligence Applications Release 7.9.5 does not support GL flex field or HR security profile-based security against EBS.

6.2.2 Operating Unit-Based Security With Oracle EBS

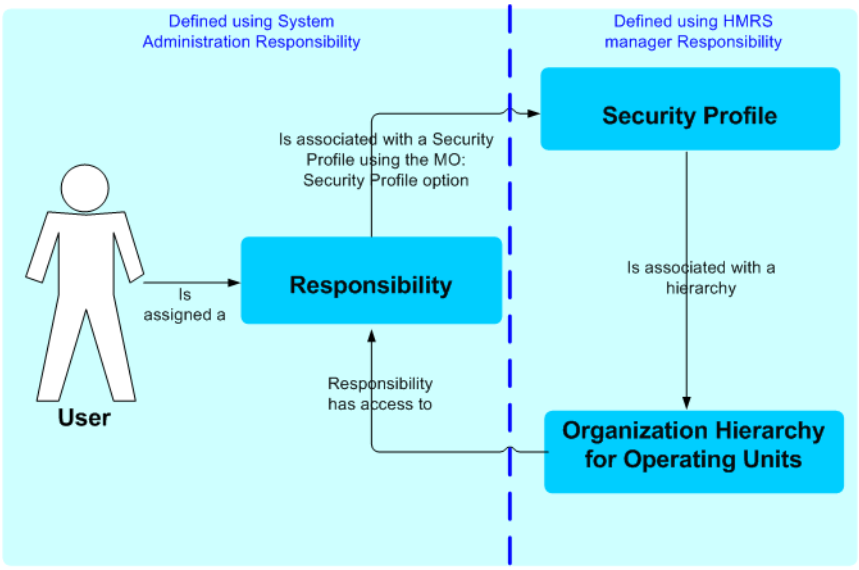
This section covers Operating Unit-Based Security With Oracle EBS.

6.2.2.1 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 6–1 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.

6.2.2.2 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 Business Intelligence 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 Business Intelligence 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 Business Intelligence 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 Business Intelligence 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 -- '<u>Operating Unit Org</u>'

  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))
```

6.2.3 Inventory Organization-Based Security With Oracle EBS

This section explains Inventory Organization-Based Security With Oracle EBS.

6.2.3.1 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.

6.2.3.2 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 Business Intelligence 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 Business Intelligence 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 Business Intelligence 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 Business Intelligence 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)
```

6.2.4 Company Org-based Security Against Oracle EBS

Company Org-based security against EBS was introduced in 7.9.3. It was replaced by Ledger Security in Version 7.9.4.

This section covers Ledger Org-based security against Oracle EBS.

6.2.4.1 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.

6.2.4.2 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 Business Intelligence 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 Business Intelligence 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 Business Intelligence 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 Business Intelligence 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 -- '<u>Companies</u>'

  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.

6.2.5 Ledger Security Against Oracle EBS

Ledger-based security against Oracle EBS was introduced in Oracle Business Intelligence 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 Business Intelligence 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 Business Intelligence 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 Business Intelligence 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 Business Intelligence Applications 7.9.5 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))
```

6.2.6 Business Group Org-based Security Against Oracle EBS

This section explains Business Group Org-based security against Oracle EBS.

6.2.6.1 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.

6.2.6.2 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 Business Intelligence 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 Business Intelligence 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 Business Intelligence 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 Business Intelligence 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 -- '<u>B</u>usiness 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.

6.2.7 HR Org-based Security Against Oracle EBS

The sequence for HR Org-based Security against Oracle EBS is described below:

- When a user logs in to Oracle Business Intelligence Applications, the session variable below is set automatically.

  ```
  USER (System variable)
  ```

- The Oracle BI Server gets the HR organizations corresponding to the USER from the following tables:
  - FND_USER_RESP_GROUPS
  - FND_USER
  - PER_SECURITY_PROFILES
  - PER_SEC_PROFILE_ASSIGNMENTS
  - PER_PERSON_LIST

    Note: Before the PER_PERSON_LIST table can be used, you must ensure that you have run the Oracle EBS HRMS Security List Maintenance process.
  - PER_ALL_ASSIGNMENTS_F
- 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 -- '<u>HR Organizations</u>'

  The initialization block 'HR Organizations' sets value for variable HR_ORG using the following SQL. The actual SQL query differs depending on whether Multiple Security Group (MSG) is set up or not.

  The following SQL should be used when MSG is not in place:

  ```
  SELECT
   DISTINCT 'HR_ORG'
  ,TO_CHAR(SEC_DET.ORGANIZATION_ID)
  FROM
  (
  ```

```
SELECT
  'HR_ORG',
  ASG.ORGANIZATION_ID
FROM
 FND_USER_RESP_GROUPS URP
,FND_USER USR
,PER_SECURITY_PROFILES PSEC
,PER_PERSON_LIST PER
,PER_ALL_ASSIGNMENTS_F ASG
WHERE
    URP.START_DATE < TRUNC(SYSDATE)
AND (CASE WHEN URP.END_DATE IS NULL THEN TRUNC(SYSDATE) ELSE TO_DATE(URP.END_
DATE) END) >= TRUNC(SYSDATE)
AND USR.USER_NAME = ':USER'
AND USR.USER_ID = URP.USER_ID
AND TRUNC(SYSDATE)
   BETWEEN URP.START_DATE AND NVL(URP.END_DATE, HR_GENERAL.END_OF_TIME)
AND PSEC.SECURITY_PROFILE_ID = FND_PROFILE.VALUE_SPECIFIC('PER_SECURITY_
PROFILE_ID', URP.USER_ID, URP.RESPONSIBILITY_ID, URP.RESPONSIBILITY_
APPLICATION_ID)
AND PER.SECURITY_PROFILE_ID = PSEC.SECURITY_PROFILE_ID
AND PER.PERSON_ID = ASG.PERSON_ID
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND URP.RESPONSIBILITY_ID = DECODE(FND_GLOBAL.RESP_ID,
                                   -1, URP.RESPONSIBILITY_ID,
                                   NULL, URP.RESPONSIBILITY_ID,
                                   FND_GLOBAL.RESP_ID)
UNION
SELECT DISTINCT 'HR_ORG',
      ORGANIZATION_ID
FROM PER_ALL_ASSIGNMENTS_F ASG,
     FND_USER USR
WHERE ASG.PERSON_ID = USR.EMPLOYEE_ID
AND USR.USER_NAME = ':USER'
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND ASG.PRIMARY_FLAG = 'Y'
) SEC_DET
```

The following SQL should be used when MSG is in place:

```
SELECT
 DISTINCT 'HR_ORG'
,TO_CHAR(SEC_DET.ORGANIZATION_ID)
FROM
(
SELECT 'HR_ORG',
ASG.ORGANIZATION_ID
FROM FND_USER_RESP_GROUPS URP,
  FND_USER USR,
  PER_SEC_PROFILE_ASSIGNMENTS SASG,
  PER_SECURITY_PROFILES PSEC,
  PER_PERSON_LIST PER,
  PER_ALL_ASSIGNMENTS_F ASG
WHERE  URP.START_DATE < TRUNC(SYSDATE)
AND (CASE WHEN URP.END_DATE IS NULL THEN TRUNC(SYSDATE) ELSE TO_DATE(URP.END_
DATE) END) >= TRUNC(SYSDATE)
AND USR.USER_NAME = ':USER'
AND URP.SECURITY_GROUP_ID = SASG.SECURITY_GROUP_ID
AND URP.USER_ID = USR.USER_ID
AND TRUNC(SYSDATE)
    BETWEEN URP.START_DATE AND NVL(URP.END_DATE, HR_GENERAL.END_OF_TIME)
```

```
AND URP.USER_ID = SASG.USER_ID
AND URP.RESPONSIBILITY_ID = SASG.RESPONSIBILITY_ID
AND URP.RESPONSIBILITY_APPLICATION_ID = SASG.RESPONSIBILITY_APPLICATION_ID
AND PSEC.SECURITY_PROFILE_ID = SASG.SECURITY_PROFILE_ID
AND PSEC.SECURITY_PROFILE_ID = PER.SECURITY_PROFILE_ID
AND PER.PERSON_ID = ASG.PERSON_ID
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND TRUNC(SYSDATE) BETWEEN SASG.START_DATE AND NVL(SASG.END_DATE, HR_
GENERAL.END_OF_TIME)
AND URP.RESPONSIBILITY_ID = DECODE(FND_GLOBAL.RESP_ID,
                                   -1, URP.RESPONSIBILITY_ID,
                                   NULL, URP.RESPONSIBILITY_ID,
                                   FND_GLOBAL.RESP_ID)
UNION
SELECT DISTINCT 'HR_ORG', ORGANIZATION_ID
FROM PER_ALL_ASSIGNMENTS_F ASG,
     FND_USER USR
WHERE ASG.PERSON_ID = USR.EMPLOYEE_ID
AND USR.USER_NAME = ':USER'
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND ASG.PRIMARY_FLAG= 'Y'
) SEC_DET
```

- 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.

6.2.8 Human Resource Personnel Data Analyst Security Against Oracle EBS

HR personnel need to see all data for the Internal Organizations for which they are responsible 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:

  Note: The actual SQL query differs depending on whether Multiple Security Group (MSG) is set up or not.

  The following SQL should be used when MSG is not in place.

```
SELECT
 DISTINCT 'HR_ORG'
,TO_CHAR(SEC_DET.ORGANIZATION_ID)
FROM
(
SELECT
  'HR_ORG',
  ASG.ORGANIZATION_ID
FROM
 FND_USER_RESP_GROUPS URP
,FND_USER USR
,PER_SECURITY_PROFILES PSEC
,PER_PERSON_LIST PER
```

```
,PER_ALL_ASSIGNMENTS_F ASG
WHERE
    URP.START_DATE < TRUNC(SYSDATE)
AND (CASE WHEN URP.END_DATE IS NULL THEN TRUNC(SYSDATE) ELSE TO_DATE(URP.END_
DATE) END) >= TRUNC(SYSDATE)
AND USR.USER_NAME = ':USER'
AND USR.USER_ID = URP.USER_ID
AND TRUNC(SYSDATE)
   BETWEEN URP.START_DATE AND NVL(URP.END_DATE, HR_GENERAL.END_OF_TIME)
AND PSEC.SECURITY_PROFILE_ID = FND_PROFILE.VALUE_SPECIFIC('PER_SECURITY_
PROFILE_ID', URP.USER_ID, URP.RESPONSIBILITY_ID, URP.RESPONSIBILITY_
APPLICATION_ID)
AND PER.SECURITY_PROFILE_ID = PSEC.SECURITY_PROFILE_ID
AND PER.PERSON_ID = ASG.PERSON_ID
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND URP.RESPONSIBILITY_ID = DECODE(FND_GLOBAL.RESP_ID,
                                   -1, URP.RESPONSIBILITY_ID,
                                   NULL, URP.RESPONSIBILITY_ID,
                                   FND_GLOBAL.RESP_ID)
UNION
SELECT DISTINCT 'HR_ORG',
      ORGANIZATION_ID
FROM PER_ALL_ASSIGNMENTS_F ASG,
     FND_USER USR
WHERE ASG.PERSON_ID = USR.EMPLOYEE_ID
AND USR.USER_NAME = ':USER'
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND ASG.PRIMARY_FLAG = 'Y'
) SEC_DET
```

The following SQL should be used when MSG is in place.

```
SELECT
 DISTINCT 'HR_ORG'
,TO_CHAR(SEC_DET.ORGANIZATION_ID)
FROM
(
SELECT 'HR_ORG',
ASG.ORGANIZATION_ID
FROM FND_USER_RESP_GROUPS URP,
  FND_USER USR,
  PER_SEC_PROFILE_ASSIGNMENTS SASG,
  PER_SECURITY_PROFILES PSEC,
  PER_PERSON_LIST PER,
  PER_ALL_ASSIGNMENTS_F ASG
WHERE  URP.START_DATE < TRUNC(SYSDATE)
AND (CASE WHEN URP.END_DATE IS NULL THEN TRUNC(SYSDATE) ELSE TO_DATE(URP.END_
DATE) END) >= TRUNC(SYSDATE)
AND USR.USER_NAME = ':USER'
AND URP.SECURITY_GROUP_ID = SASG.SECURITY_GROUP_ID
AND URP.USER_ID = USR.USER_ID
AND TRUNC(SYSDATE)
   BETWEEN URP.START_DATE AND NVL(URP.END_DATE, HR_GENERAL.END_OF_TIME)
AND URP.USER_ID = SASG.USER_ID
AND URP.RESPONSIBILITY_ID = SASG.RESPONSIBILITY_ID
AND URP.RESPONSIBILITY_APPLICATION_ID = SASG.RESPONSIBILITY_APPLICATION_ID
AND PSEC.SECURITY_PROFILE_ID = SASG.SECURITY_PROFILE_ID
AND PSEC.SECURITY_PROFILE_ID = PER.SECURITY_PROFILE_ID
AND PER.PERSON_ID = ASG.PERSON_ID
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND TRUNC(SYSDATE) BETWEEN SASG.START_DATE AND NVL(SASG.END_DATE, HR_
```

```
GENERAL.END_OF_TIME)
AND URP.RESPONSIBILITY_ID = DECODE(FND_GLOBAL.RESP_ID,
                                   -1, URP.RESPONSIBILITY_ID,
                                   NULL, URP.RESPONSIBILITY_ID,
                                   FND_GLOBAL.RESP_ID)
UNION
SELECT DISTINCT 'HR_ORG', ORGANIZATION_ID
FROM PER_ALL_ASSIGNMENTS_F ASG,
     FND_USER USR
WHERE ASG.PERSON_ID = USR.EMPLOYEE_ID
AND USR.USER_NAME = ':USER'
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND ASG.PRIMARY_FLAG= 'Y'
) SEC_DET
```

- 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 'USER_HR_ORG', ORGANIZATION_ID
FROM PER_ALL_ASSIGNMENTS_F ASG,
     FND_USER USR
WHERE ASG.PERSON_ID = USR.EMPLOYEE_ID
AND USR.USER_NAME = ':USER'
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND ASG.PRIMARY_FLAG= 'Y'
```

- 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, then it will be compared next to the user's own organization. If they are different, then there is no further check, and the record is selected. If they are the same, then 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.

6.2.9 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 6.5.2, "Primary Position Based Security for CRM Vertical applications".

6.3 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 6.3.1, "Authorization"
- Appendix 6.3.2, "About Operating Unit-based Security Against PeopleSoft Financials"
- Appendix 6.3.3, "About Company Org-based Security Against PeopleSoft Financials or PeopleSoft HR"
- Appendix 6.3.4, "About Ledger-based Security against Oracle's PeopleSoft Enterprise Applications"
- Appendix 6.3.5, "About HR Org-based Security Against PeopleSoft HR"
- Appendix 6.3.6, "About Payables Org-based security Against PeopleSoft Financials"
- Appendix 6.3.7, "About Receivables Org-based Security Against PeopleSoft Financials"
- Appendix 6.3.8, "About SET ID-based Security Against PeopleSoft Financials"
- Appendix 6.3.9, "About Human Resource Personnel Data Analyst Security Against PeopleSoft HR"

6.3.1 Authorization

The authorization process of Oracle Business Intelligence Applications fetches a user's role from source PeopleSoft applications, matches the role with all Oracle Business Intelligence 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"

6.3.2 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 Business Intelligence 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.

6.3.3 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 Business Intelligence 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 -- '<u>Companies</u>'

  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.

6.3.4 About Ledger-based Security against Oracle's PeopleSoft Enterprise Applications

Ledger-based security against PeopleSoft Enterprise Applications was 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 Business Intelligence 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.

6.3.5 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 Business Intelligence 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 -- '<u>HR Organizations</u>'

  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.

6.3.6 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 Business Intelligence 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 -- '<u>Payables Organizations</u>'

  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.

6.3.7 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 Business Intelligence 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 -- '<u>Receivables Organizations</u>'

  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.

6.3.8 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 Business Intelligence 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.

6.3.9 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.

6.3.10 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 6.5.2, "Primary Position Based Security for CRM Vertical applications".

6.4 Metadata Object-Level Security in Oracle Business Intelligence Applications

Object-level security controls access to various Analytics objects, primarily metadata and Presentation Services objects.

6.4.1 Metadata Object-Level Security (Repository Groups)

Repository groups control access to metadata objects such as subject areas, tables and columns.

6.4.1.1 Where is Repository Groups Security Configured?

Metadata object security is configured in the Analytics repository (EnterpriseBusinessAnalytics.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 6–1.

Table 6–1 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
Channel Managers	All Channel applications groups
Partner Managers	All Partner application groups

6.4.1.2 Metadata Object-Level (Presentation Services) Security in Oracle Business Intelligence 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 6–1 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.

6.5 Integrating Data Security with the Siebel CRM Application

This section describes how Oracle Business Intelligence Applications security integrates with Siebel CRM security, and contains the following topics:

- Section 6.5.1, "Primary Position based Security"
- Section 6.5.2, "Primary Position Based Security for CRM Vertical applications"
- Section 6.5.3, "Team Based Visibility"
- Section 6.5.4, "Embedded Oracle Business Intelligence support for Changing Positions"
- Section 6.5.5, "Business Unit based Security"
- Section 6.5.6, "Primary Owner Security"

6.5.1 Primary Position based Security

This section covers Primary Position based Security and contains the following sections:

- Section 6.5.1.1, "Introduction"
- Section 6.5.1.2, "Primary Position based Security Group"
- Section 6.5.1.3, "Configuring Oracle BI Repository Table Joins for Position/Employee Based Security"

6.5.1.1 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_POSTION_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.

6.5.1.2 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 6.6.3, "Initialization Blocks Used For Security Oracle Business Intelligence 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.

6.5.1.3 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, click the Permissions box and select the Filter tab.
    c. To add a new filter, click on the Add button.
    d. In the new dialog, 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, click the Permissions box and select the Filter tab.
    c. To add a new filter, click on the Add button.
    d. In the new dialog, 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.

6.5.2 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 6.5.2.1, "Oracle Partner Analytics Security Settings"
- Section 6.5.2.2, "Consumer Sector Analytics Security Settings"

- Section 6.5.2.3, "Communications, Media, and Energy (CME) Analytics Security Settings"
- Section 6.5.2.4, "Financial Services Analytics Security Settings"
- Section 6.5.2.5, "Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics Security Settings"
- Section 6.5.2.6, "Usage Accelerator Analytics Security Settings"

6.5.2.1 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.

6.5.2.1.1 Partner Manager Role-Based Interactive Dashboards Mapping

Table 6–2 provides the dashboard and page tab mapping for specific responsibilities in the Siebel PRM Partner Manager application.

Table 6–2 Siebel Responsibilities for PRM Analytics

Responsibility	Dashboard	Page Tab Name
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

Table 6–2 (Cont.) Siebel Responsibilities for PRM Analytics

Responsibility	Dashboard	Page Tab Name
	Channel Executive	Service
	Channel Segmentation	Channel Mix
	Channel Segmentation	Partner Territory
	Channel Segmentation	Partner Tier
	Channel Segmentation	Partner Type
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
	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

Table 6–2 (Cont.) Siebel Responsibilities for PRM Analytics

Responsibility	Dashboard	Page Tab Name
	Channel Training	Overview
	Channel Training	Performance

6.5.2.1.2 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 6–3.

Table 6–3 Responsibilities for PRM Partner Portal Analytics

Responsibility	Dashboard	Page Tab Name
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
	Partner Service	Service Requests
	Partner Training	Training
Partner Sales Manager Analytics User	Partner Commerce	Orders
	Partner Commerce	Overview
	Partner Commerce	Quotes
	Partner Sales	Pipeline
	Partner Sales	Revenue
	Partner Sales	Subordinates
	Partner Training	Subordinates
Partner Sales Rep Analytics User	Partner Commerce	Orders
	Partner Commerce	Overview
	Partner Commerce	Quotes
	Partner Sales	Pipeline
	Partner Sales	Revenue
	Partner Sales	Subordinates
	Partner Training	Subordinates

Table 6–3 (Cont.) Responsibilities for PRM Partner Portal Analytics

Responsibility	Dashboard	Page Tab Name
Partner Service Manager Analytics User	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

6.5.2.1.3 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 6–4, where a X indicates that subject area is visible for the user holding that responsibility.

Table 6–4 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	

6.5.2.1.4 PRM Analytics Subject Area Visibility The subject area visibility for roles in Partner Portal is shown in Table 6–5, where a X indicates that subject area is visible for the user holding that responsibility.

Table 6–5 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						
Products	X	X	X	X	X	X
Real-Time Activity						
Real-Time Assets						
Service Requests	X	X			X	X

6.5.2.1.5 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 6.6.2, "Implementing Data-Level Security in the Oracle Business Intelligence Applications Repository". To change Partner Service Representative and Sales Representative to Organization-based security, follow the procedure shown in the topic Section 6.5.4, "Embedded Oracle Business Intelligence support for Changing Positions".

Table 6–6 shows the data-level security settings included for the responsibilities in Partner Manager and Partner Portal.

Table 6–6 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

Table 6–6 (Cont.) Oracle PRM Data-Level Security Settings

Responsibility	Data-Level Security	Type	Comments
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.

6.5.2.2 Consumer Sector Analytics Security Settings

Table 6–7 describes the consumer sector responsibilities associated with each CS Dashboard.

Table 6–7 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

6.5.2.3 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 6.1, "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 6–8.

Table 6–8 CME Responsibilities Associated with Each CME Dashboard

CME Responsibility	CME Dashboard	Dashboard Pages
CM Marketing Analytics User CM Marketing Analytics Administrator	Loyalty Management	▪ Customer Lifetime Value ▪ Churn Propensity ▪ Selling Propensity ▪ Financial Risk ▪ Actual Churn
CM Sales Analytics User CM Sales Analytics Administrator	Revenue Management	▪ Sales Portal ▪ Service Activations ▪ Service Modifications ▪ Service Disconnections
	Account Management	▪ Sales Portal ▪ Service Activations ▪ Service Modifications ▪ Service Disconnections
CM Service Analytics User CM Service Analytics Administrator	Account Management	▪ Trouble Tickets ▪ Customer Satisfaction

6.5.2.4 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 6–9.

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.

6.5.2.4.1 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 6–9.

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 6–9 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 6–11.

Table 6–9 Financial Services Responsibility Required to View FS Dashboards

FS Responsibilities	Dashboards
Financial Analytics User	Credit
	Credit Card
	Private Banking
	Consumer Banking
	Corporate and Commercial Banking
	Investment Holdings
	Separate Account Management
	Wealth Management
	Institutional Sales
	Investment Banking
	Finance Marketing
	Finance Executive
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)	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

6.5.2.5 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 6–10 shows Pharma Analytics Responsibilities and Functions.

Table 6–10 Pharma Analytics Responsibilities and Functions

Responsibility	Use
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.

Table 6–10 (Cont.) Pharma Analytics Responsibilities and Functions

Responsibility	Use
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.
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.

6.5.2.6 Usage Accelerator Analytics Security Settings

Table 6–11 describes the additional security configurations that may be necessary and the particular responsibilities associated with the Oracle Usage Accelerator dashboards.

Table 6–11 Usage Accelerator Responsibilities and Dashboards

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
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

Table 6–11 (Cont.) Usage Accelerator Responsibilities and Dashboards

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
		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
		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

Table 6–11 (Cont.) Usage Accelerator Responsibilities and Dashboards

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
		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.

6.5.3 Team Based Visibility

This section covers team based visibility and contains the following sections:

- Section 6.5.3.1, "Summary"
- Section 6.5.3.2, "Business Requirements Overview"
- Section 6.5.3.3, "Design Details"
- Section 6.5.3.4, "Data Model"
- Section 6.5.3.5, "Fact Security"
- Section 6.5.3.6, "ETL Design"
- Section 6.5.3.7, "DAC Metadata"
- Section 6.5.3.8, "Oracle Business Intelligence Applications Repository Metadata"

6.5.3.1 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 Business Intelligence 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.

6.5.3.2 Business Requirements Overview

In Siebel transactional application, the 'my team' view allows users to see records owned by the user's team. It has been noted that a 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 Business Intelligence 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.

6.5.3.3 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.

6.5.3.4 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 6–12 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).
PARTY_TYPE	'P' for Position party type, 'O' for Organization party type, 'E' for Employee party type.

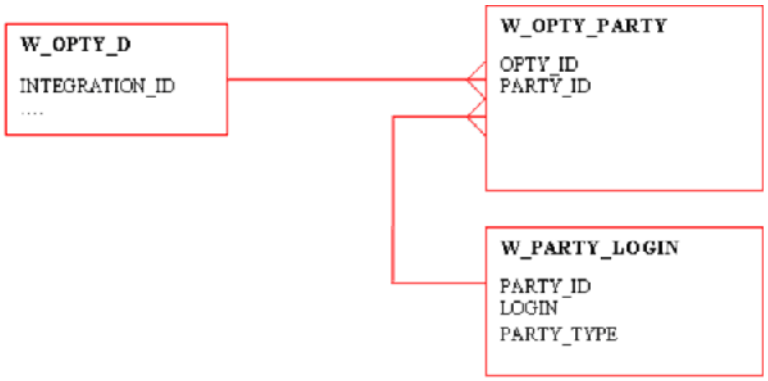
The transactional source for this table is S_PARTY_RPT_REL. All possible pairs of position_id and its parent's login at any level are selected from the source table during ETL and inserted in this table. The same table is being used to store hierarchy relationships for other party objects (for example, 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 6–13 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 6–14 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.

6.5.3.5 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.

6.5.3.6 ETL Design

The sections covers ETL Design.

6.5.3.6.1 Available Mappings There will be a set of ETL routine to populate the tables specified 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.

6.5.3.6.2 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.

6.5.3.6.3 DAC Metadata All the Visibility tables and corresponding tasks specified 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 the DAC Client, 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.

6.5.3.7 DAC Metadata

All the Visibility tables and corresponding tasks specified 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 the DAC Client, 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.

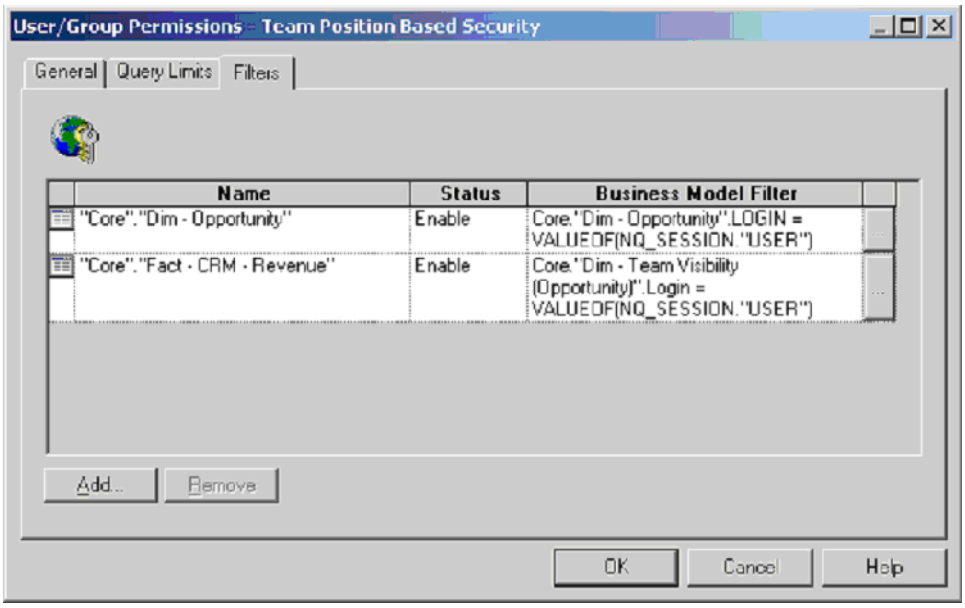
6.5.3.8 Oracle Business Intelligence Applications Repository Metadata

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 Business Intelligence 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 Business Intelligence 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.

6.5.4 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`

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\EmbeddedAnaltyics\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.

6.5.5 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'
```

6.5.6 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'
```

6.6 Data-Level Security In Oracle Business Intelligence Applications

This section describes the data-level security features in Oracle Business Intelligence Applications, and contains the following topics:

- Section 6.6.1, "Data-Level Security in Oracle Business Intelligence Applications"
- Section 6.6.2, "Implementing Data-Level Security in the Oracle Business Intelligence Applications Repository"
- Section 6.6.3, "Initialization Blocks Used For Security Oracle Business Intelligence Applications"

6.6.1 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.

6.6.1.1 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 Business Intelligence 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 Business Intelligence 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 release 7.9.4, Oracle Business Intelligence 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, buy 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 6–15 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	*Available since 7.9*	Available since 7.9		*Available since 7.5*
Inventory Org-based Security	Available since 7.9			
Company Org-based Security	*Available in 7.9.3 and obsolete in 7.9.4*	*Available in 7.9.3*	*Available in 7.9.3*	
Business Group Org-based Security	*Available since 7.9.3*			
HR Org-based Security			*Available since 7.9.3*	
Payables Org-based Security		*Available since 7.9.3*		
Receivables Org-based Security		*Available since 7.9.3*		
SET ID-based security		*Available since 7.9.3*	*Available since 7.9.3*	
Position-based security	*Available since 7.9.4 for HRMS*		*Available since 7.9.3*	*Available since 7.5*
Ledger-based security	*Available since 7.9.4*	*Available since 7.9.4*		

6.6.2 Implementing Data-Level Security in the Oracle Business Intelligence Applications Repository

Data-level security in Oracle Business Intelligence 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 6.6.3, "Initialization Blocks Used For Security Oracle Business Intelligence 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 6.5.1, "Primary Position based Security" for Employee/Position Based security, and Section 6.2, "Integrating Data Security with Oracle E-Business Suite" and Section 6.3, "Integrating Data Security with Oracle's PeopleSoft Enterprise Applications" for Organization based security.

6.6.3 Initialization Blocks Used For Security Oracle Business Intelligence 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 Business Intelligence 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.

6.6.3.1 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 6–16 Data Security Groups in Oracle Business Intelligence 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.

Table 6–16 (Cont.) Data Security Groups in Oracle Business Intelligence Applications

Security Group Name	Application	Description
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.
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 mandatory and optional steps required to configure Oracle Business Intelligence Applications, and contains the following chapters:

- Chapter 7, "Configuring Common Areas and Dimensions"
- Chapter 8, "Configuring Oracle Procurement and Spend Analytics"
- Chapter 9, "Configuring Oracle Financial Analytics"
- Chapter 10, "Configuring Oracle Supply Chain and Order Management Analytics"
- Chapter 11, "Configuring Oracle Human Resources Analytics"
- Chapter 12, "Configuring Oracle Sales Analytics"
- Chapter 13, "Configuring Oracle Contact Center Telephony Analytics"
- Chapter 14, "Configuring the Oracle Business Intelligence Applications Repository"

Note: For a high level road map for installation, configuration, and customization steps for Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications".

7

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 7.1, "Source Independent Configuration Steps"
- Section 7.2, "Oracle EBS-Specific Configuration Steps"
- Section 7.3, "PeopleSoft-Specific Configuration Steps"
- Section 7.4, "Siebel-Specific Configuration Steps"

7.1 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 7.1.1, "Configuration Required Before A Full Load"
- Section 7.1.2, "Configuration Steps for Controlling Your Data Set for All Source Systems"

7.1.1 Configuration Required Before A Full Load

This section contains configuration steps required before a full data load that apply to Oracle Business Intelligence Applications deployed with any source system, and contains the following topics:

- Section 7.1.1.1, "How to Configure Initial Extract Date"
- Section 7.1.1.2, "How to Configure Global Currencies"
- Section 7.1.1.3, "How to Configure Exchange Rate Types"
- Section 7.1.1.4, "How to Configure Fiscal Calendars"
- Section 7.1.1.5, "How to Configure the General Ledger Account Hierarchies"
- Section 7.1.1.6, "Configuring General Ledger Account Hierarchies Using General Ledger Accounting Flexfield value sets definitions"
- Section 7.1.1.7, "How to Configure General Ledger Account Hierarchies Using Financial Statement Generator (FSG) Report Definition"

7.1.1.1 How to Configure Initial Extract Date

Initial Extract Date is required when you extract data for a full load. It reduces the volume of data in the initial load. The specified initial extract date will be used as a filter on the creation date of OLTP data in the selected full extract mapping.

When you set the Initial Extract Date parameter, make sure that you set it to the beginning of an accounting period, and not a date in the middle of an accounting period. For example, if you decide to extract data from June 2005, and the June 2005 accounting period starts from 5th June, set the date to 5th June, 2005.

Extract mappings that use the Initial Extract Date parameter include:

- SDE_ORA_FreightTermsDimension
- SDE_ORA_GL_AP_LinkageInformation_Extract
- SDE_ORA_GL_AR_REV_LinkageInformation_ Extract
- SDE_ORA_GL_COGS_LinkageInformation_Extract
- SDE_ORA_PayrollFact_Agg_Items_Derive

To configure the initial extract date in your warehouse:

1. In the DAC Client, display the Design view, and select a container from the drop down list to the right of the Execute button.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Display the 'Source System Parameters' tab.
3. Edit the value of the $$INITIAL_EXTRACT_DATE parameter.
4. Save your changes.

7.1.1.2 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 Business Intelligence 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. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Select a container from the drop down list to the right of the Execute button.
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 that you spell the currencies as they are spelled in your source OLTP system.

5. Save your changes.

7.1.1.3 How to Configure Exchange Rate Types

When Oracle Business Intelligence 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 Business Intelligence Applications also allows you to specify the exchange rate type to use to perform the conversion. Oracle Business Intelligence Applications also provides three global exchange rate types for you to configure.

Oracle Business Intelligence 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. In the DAC Client, select a container from the drop down list to the right of the Execute button.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Display the 'Source System Parameters' tab.

3. 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.
4. Save your changes

7.1.1.4 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 7.1.1.4.1, "Overview To Setting Up The Time Dimension"
- Section 7.1.1.4.2, "Setting Up Fiscal Calendar"
- Section 7.1.1.4.3, "Setting Up Fiscal Calendar By Fiscal Week"
- Section 7.1.1.4.4, "Setting Up Fiscal Calendar By Fiscal Month"
- Section 7.1.1.4.5, "Process of Setting Up The Fiscal Calendar"
- Section 7.1.1.4.6, "Reloading the Time Dimension Tables After Your Data Warehouse is Populated"
- Section 7.1.1.4.7, "Notes"

7.1.1.4.1 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 7–1 shows the tables used to store calendar information.

Table 7–1 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.

7.1.1.4.2 Setting Up Fiscal Calendar Installed out of the box, Oracle Business Intelligence 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.

7.1.1.4.3 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.

7.1.1.4.4 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.

7.1.1.4.5 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 \OracleBI\dwrep\Informatica\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 the DAC. Then right click and push to references.
3. Activate the task SIL_DayDimension_FiscalWeek_Extract in the Datawarehouse container in the 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 \OracleBI\dwrep\Informatica\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 the 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.

7.1.1.4.6 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 the DAC Client, display the Setup view, display the Physical Data Sources tab 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. In the Design view, display the Tasks 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.

7.1.1.4.7 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.
- As installed out-of-the-box, Oracle Business Intelligence Applications can generate up to 65536 rows. If you need more than 65536 rows, you can increase the capacity to 262144 rows (718 years) by doing the following:
  1. Duplicate 'SIL_DayDimension_GenerateRows7'.
  2. Rename it 'SIL_DayDimension_GenerateRows8'.
  3. Run this immediately after SIL_DayDimension_GenerateRows7.

7.1.1.5 How to Configure the General Ledger Account Hierarchies

These configuration steps are required if you are deploying Oracle Financial Analytics, Oracle Procurement and Spend Analytics, or Oracle Supply Chain and Order Management Analytics.

The 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

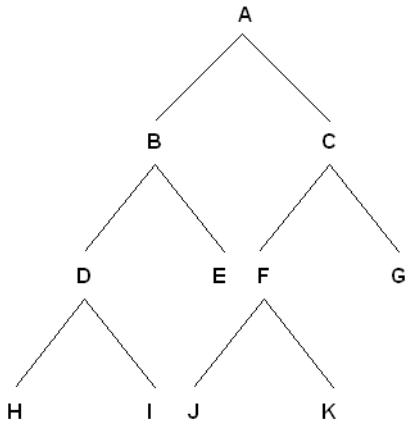
Whichever method you choose to set up General Ledger Account hierarchies, you store the hierarchy information in the W_HIERARCHY_D table.

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 7–1 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 7–2 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

7.1.1.6 Configuring General Ledger Account Hierarchies Using General Ledger Accounting Flexfield value sets definitions

These configuration steps are required if you are deploying Oracle Financial Analytics, Oracle Procurement and Spend Analytics, or Oracle Supply Chain and Order Management Analytics.

Oracle EBS supports up to 30 segments in which to store accounting flexfields. flexfields 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 7–3 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
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.

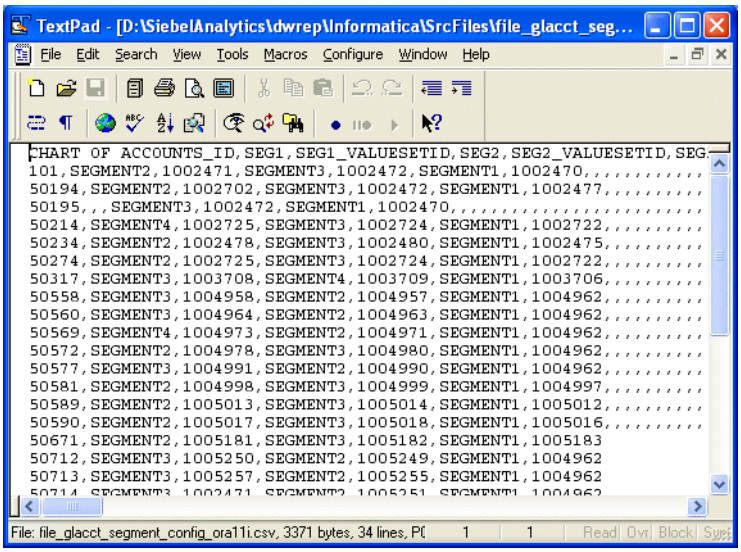
For example, we can store Company segments from COA 4256 and 4257 in segment1 column in W_GL_ACCOUNT_D; and Cost Center segments from COA 4256 and 4257 in segment2 column in W_GL_ACCOUNT_D, and so forth.

About Configuring the ETL Process for GL Accounting flexfields

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:

$PMSERVER/srcfile/file_glacct_segment_configur_ora.csv

Figure 7–2 Screen Shot of file_glacct_segment_config_ora.csv File Opened in a Text Editor



In the file_glacct_segment_config_ora.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.

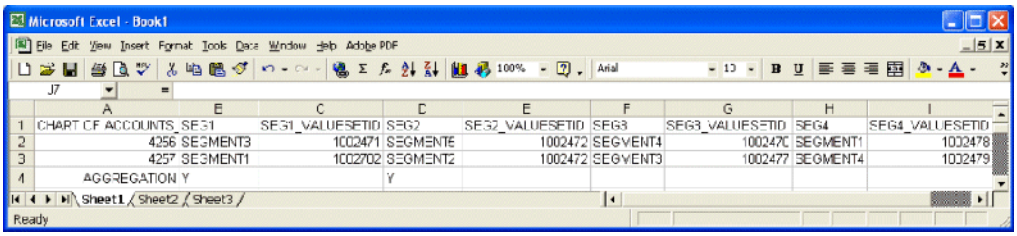
- Store all Natural Account segments from all COAs in ACCOUNT_SEG3_CODE column in W_GL_ACCOUNT_D.

- 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_ora.csv would be configured to implement the business requirements specified above.

Figure 7–3 Screenshot of file_glacct_segment_config_ora.csv in text editor



To Set Up Hierarchies With General Ledger Accounting flexfields

1. Use the /srcfile/file_glacct_segment_configur_ora.csv file to specify the segments that you want to analyze.
2. In the DAC Client, do the following:
   a. For the Subject Area 'Financials – General Ledger', in the 'Configuration Tags' 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.

      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. Using the Oracle BI Administration Tool, in the RPD Physical Layer, create additional aliases or change the names of the existing alias against the table W_HIERARCHY_D.

   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
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. Using the Oracle BI Administration Tool, in the Physical Layer, 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_Compnay_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. Join the aliases of W_HIERARCHY_D to the corresponding columns in the table W_GL_BALANCE_A.

   - Dim_Company_Hierarchy_D.HIER20_CODE = Fact_Agg_W_GL_BALANCE_A.ACCOUNT_SEG1_CODE AND
   - Dim_Company_Hierarchy_D.HIER_CODE = Fact_Agg_W_GL_BALANCE_A.ACCOUNT_SEG1_ATTRIB

   Note: W_GL_BALANCE_A has only 6 segment columns. So, if you have more than 6 hierarchies, join only the first 6 to W_GL_BALANCE_A but join all hierarchies to W_GL_ACCOUNT_D in the previous step.

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. You must set the HIER_CODE to be filtered by the Valueset IDs that are applicable for that particular hierarchy. The list of the valueset ids would be same as the valueset ids you configured in the csv file in the first set.

   To specify a filter in the Business Model Layer to restrain the output of logical table, do the following:

   a. Expand each logical table for the hierarchies, say Dim - GL ValueSetHierarchy1, and open the logical table source under it.

   b. Go to the Content tab.

   c. In the 'use WHERE clause…' text box, apply a filter on the corresponding physical table alias of W_HIERARCHY_D.

For example:

```
"Oracle Data Warehouse"."Catalog"."dbo"." Dim_W_HIERARCHY_D_
ValueSetHierarchy1"."HIER_CODE" IN (<comma seperated valuesetids>)
```

9. A HIER_CODE filter should be specified in the Business Model Layer to restrain the output of the logical table to be one Hierarchy only. To specify a filter in the Business Model Layer to restrain the output of the logical table, do the following:
   a. Right click logical table Dim_W_Hierarchy_D_ValueSet1.
   b. Choose Properties.
   c. Choose the Source tab.
   d. Select 'Dim_W_Hierarchy_D_Valueset1'.
   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_Valueset1"."HIER_CODE" = <Value Set Hierarchy Id>
      ```

      Where <*Value Set Hierarchy ID*> is the Value Set Hierarchy ID of the segment for which you are creating a hierarchy, for example, 1001.
10. Using the Oracle BI Administration Tool, in the Business Model Layer of the Oracle BI Analytics Warehouse, create additional hierarchies using the dimensions that you created in the previous step.

    For more information about example hierarchies that are installed with Oracle Business Intelligence Applications, refer to the following hierarchies in the Oracle Business Analytics Warehouse:

    – GL ValueSetHierarchy1
    – GL ValueSetHierarchy2
    – GL ValueSetHierarchy3
11. In the Business Model Layer, look for all the logical fact table that has logical join to the logical hierarchy table Dim - GL ValueSetHierarchy1. You will need to create a similar logical table between the new logical hierarchy dimensions that you created and these logical facts, as follows:
    a. Under each of the logical fact table, open the logical table sources and go to the content tab. Under aggregation content, select the **Show unmapped** check box. It will show all the hierarchies you created in the previous step. For each of these hierarchies, select the logical level to 'Detail'.
    b. In the business model diagram, create a new Complex Join between each of the new logical hierarchy tables and each of the logical facts. In the join, make sure the cardinality is (0,1) on the dimension side and N on the fact side.
12. Using the Oracle BI Administration Tool, in the Presentation Layer of the Oracle BI Analytics Warehouse, drag the new hierarchies into the presentation folder.

    Note: You can rename the hierarchies in the Presentation Layer if required.

7.1.1.7 How to Configure General Ledger Account Hierarchies Using Financial Statement Generator (FSG) Report Definition

These configuration steps are required if you are deploying Oracle Financial Analytics, Oracle Procurement and Spend Analytics, or Oracle Supply Chain and Order Management Analytics.

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 Oracle Business Intelligence Applications 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.

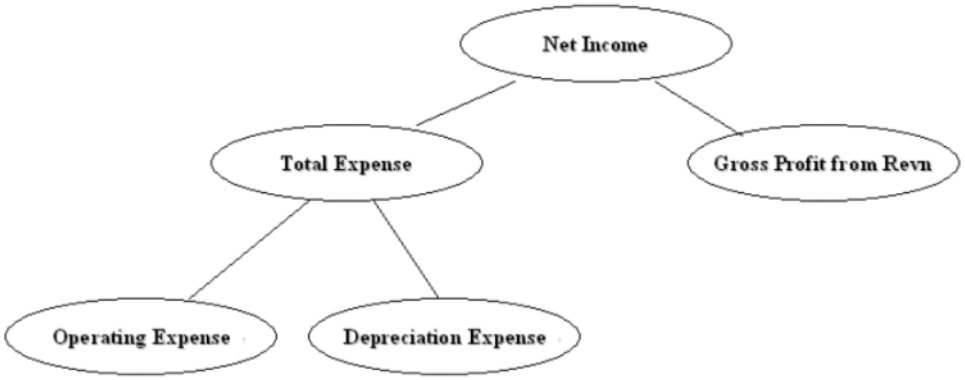
- RG_REPORT_CALCULATIONS

  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:

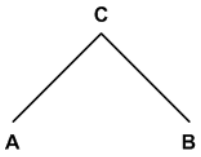
Table 7–4 Example flattened hierarchy stored in W_HIERARCHY_D

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, if you want to drill using the third hierarchy, you use W_GL_ACCOUNT_D.hier3_wid = W_HIERARCHY_D.row_wid.

Note: Mathematical operators, such as '+', '-', '*', '/' (addition, subtraction, multiplication, division, etc) are not extracted from the FSG definitions. For example, both A + B = C and A - B = C would give the same hierarchy, with a node C having two child nodes A and B (see diagram below).

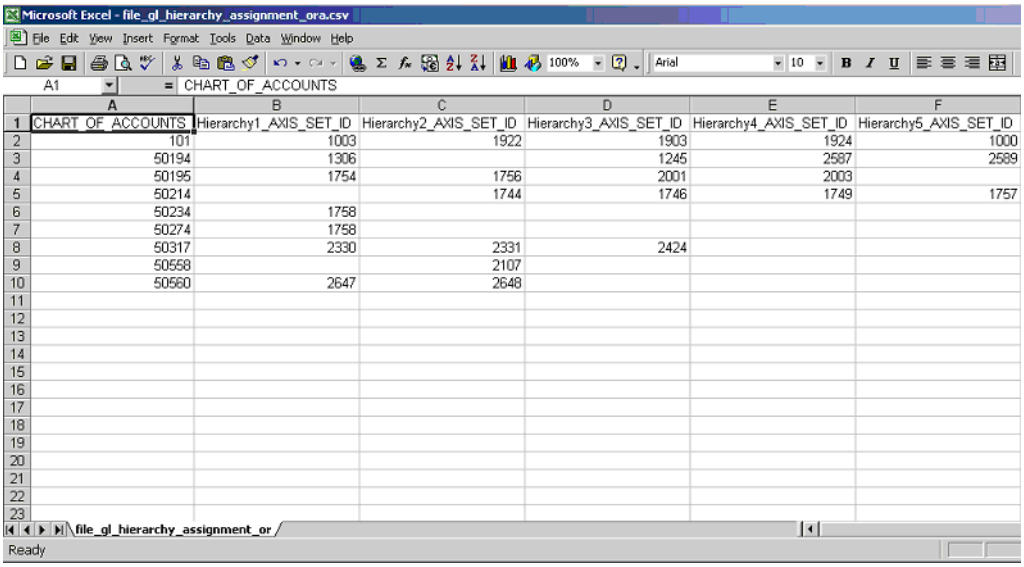


About Configuring the ETL Process for Oracle Financial Statement Generator Report

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 7–4 Screen Shot of file_gl_hierarchy_assignment_ora.csv File Opened in a Text Editor



	A	B	C	D	E	F
1	CHART_OF_ACCOUNTS	Hierarchy1_AXIS_SET_ID	Hierarchy2_AXIS_SET_ID	Hierarchy3_AXIS_SET_ID	Hierarchy4_AXIS_SET_ID	Hierarchy5_AXIS_SET_ID
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			

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 Business Intelligence 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 Client, do the following:
   a. Navigate to the Subject Area 'Financials – General Ledger'.
   b. In the 'Configuration Tags' tab, make sure that the tag 'Oracle – Extract FSG Hierarchies' is unchecked for the 'Inactive' checkbox.
   c. In the 'Configuration Tags' 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. Using the Oracle BI Administration Tool, in the RPD Physical Layer, 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. Using the Oracle BI Administration Tool, in the Physical Layer of the Oracle BI Analytics Warehouse, 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. Using the Oracle BI Administration Tool, in the Business Model Layer of the Oracle BI Analytics Warehouse, 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…' text box).

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. Using the Oracle BI Administration Tool, in the Business Model Layer of the Oracle BI Analytics Warehouse, create new dimension based on the logical table that you created in the previous step.

   Please refer to 'FSG Hierarchy 1' as an example.

8. In the Business Model Layer, look for all the logical fact table that has logical join to the logical hierarchy table Dim - FSG Hierarchy1. You will need to create a similar logical join between the new logical hierarchy dimensions that you created and these logical facts, as follows:

   a. Under each of the logical fact table, open the logical table sources and go to the content tab. Under aggregation content, select the **Show unmapped** check box. It will show all the hierarchies you created in the previous step. For each of these hierarchies, select the logical level to 'Detail'.

   b. In the business model diagram, create a new Complex Join between each of the new logical hierarchy tables and each of the logical facts. In the join, make sure the cardinality is (0,1) on the dimension side and N on the fact side.

9. Using the Oracle BI Administration Tool, in the Presentation Layer of the Oracle BI Analytics Warehouse, drag the new hierarchies into the presentation folder.

   Note: You can rename the hierarchies in the Presentation Layer if required.

7.1.2 Configuration Steps for Controlling Your Data Set for All Source Systems

This section contains additional configuration steps that apply to Oracle Business Intelligence Applications deployed with any source system, and contains the following topics:

- Section 7.1.2.1, "How to Configure Data Source Num IDs"

7.1.2.1 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 4.12.3, "How to Set Physical Data Sources".

The table below shows the data sources that are supported by Oracle Business Intelligence and their associated DATASOURCE_NUM_ID values.

Table 7–5 Data Sources and Associated DATASOURCE_NUM_ID Values

Data Source Name	Data Source Number
ORA_11_5_8	2

Table 7–5 (Cont.) Data Sources and Associated DATASOURCE_NUM_ID Values

Data Source Name	Data Source Number
ORA_11_5_9	5
ORA_11_5_10	4
ORACLE_R12	9
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. In the DAC Client, display the Setup view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
2. Display the 'Physical Data Sources' tab.
3. Select a data source from the list.
4. On the Edit tab, edit the value in the **Data Source Number** field.
5. Click Save.

If you do choose to modify the Data Source Number and if you implement Procurement and Spend Analytics Family of Products, it is Mandatory that you finish the steps in section Section 8.2.2.1, "How to Configure the DAC parameter for Purchase Cycle Line".

7.2 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 7.2.1, "Configuration Required Before A Full Load for Oracle EBS"
- Section 7.2.2, "Configuration Steps for Controlling Your Data Set for Oracle EBS"

7.2.1 Configuration Required Before A Full Load for Oracle EBS

This section contains configuration steps required before a full data load that apply to Oracle Business Intelligence Applications deployed with Oracle EBS source systems, and contains the following topics:

- Section 7.2.1.1, "Configuration of Product hierarchy (except for GL, HR modules)"
- Section 7.2.1.2, "Configure the Master Inventory Organization in Product dimension Extract for Oracle 11i adapter (Except for GL & HR Modules)"
- Section 7.2.1.3, "How to Map Oracle General Ledger Natural Accounts to Group Account Numbers"
- Section 7.2.1.4, "How to do Group Account correction for Oracle Application"

7.2.1.1 Configuration of Product hierarchy (except for GL, HR modules)

This section contains configuration points for product hierarchy in the Product dimension table and the Inventory Product dimension table.

Oracle E-Business Suite enables users to groups the products using categories and category sets. While E-Business Suite provides pre-packaged category sets, you can also define your own category sets.

As configured out-of-the-box, 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.

To configure your product category sets, do the following:

- Identify the category sets you want to report (for more information, see Section 7.2.1.1.1, "How to identify Category Sets from EBS").
- Configure the DAC parameters for product hierarchy (for more information, see Section 7.2.1.1.2, "Configure the DAC Source System Parameters for Product Hierarchy").
- Configure the hierarchy mapping with the segments (for more information, see Section 7.2.1.1.3, "Configure the Hierarchy Mapping With the Segments").

7.2.1.1.1 How to identify Category Sets from EBS These steps are part of the task in Section 7.2.1.1, "Configuration of Product hierarchy (except for GL, HR modules)".

To find out the category sets that your organization is using:

1. Log in to the EBS 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. Choose Help, then Diagnostics, then Examine, and specify the apps user password.
5. Click the Field LOV button and select CATEGORY_SET_ID, and note down the value.
6. Repeat steps 3-5 for the Purchasing functional area.

7.2.1.1.2 Configure the DAC Source System Parameters for Product Hierarchy These steps are part of the task in Section 7.2.1.1, "Configuration of Product hierarchy (except for GL, HR modules)".

To configure the DAC Source System Parameters for product hierarchy:

1. In the DAC Client, display the Design view, and choose your Oracle E-Business Suite container from the container drop down list.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Display the Source System Parameters tab.

3. Locate the INVPROD_CAT_SET_ID and PROD_CAT_SET_ID parameters, and for each parameter do the following:

   a. Use the **Value** field to specify a category set value (that is, replace the default Category Set ID 2 or 27 with a new value).

      Note: The value for INVPROD_CAT_SET_ID parameter should be set to the appropriate Inventory Category Set value. The value for the PROD_CAT_SET_ID parameter should be set to the appropriate Purchase Category Set value.

   b. Click Save.

Note: The grain of the Product dimension is at the Master level. Therefore, the category set chosen as a value for the Product Dimension parameter (PROD_CAT_SET_ID) must be a Category Set controlled at a Master Level but not at the Org level.

7.2.1.1.3 Configure the Hierarchy Mapping With the Segments These steps are part of the task in Section 7.2.1.1, "Configuration of Product hierarchy (except for GL, HR modules)".

To configure the hierarchy mapping with the segments:

1. In Informatica PowerCenter Designer, open the appropriate Oracle Applications v11i folder, and expand the Mapplets folder.

2. In the Mapplet Designer, edit the mplt_SA_ORA_ProductDimension mapplet.

3. Double-click the EXP_PRODUCTS expression transformation to display the Edit Transformations dialog.

4. In the Ports tab, scroll down to find the hierarchy code ports named EXT_PROD_HIER<*X*>_CODE.

   Hierarchy levels are named with the convention EXT_PROD_HIER<*X*>_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 pre-configured 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. Click the **Expression** field for the appropriate EXT_PROD_HIER<*X*>_CODE port, then click the down arrow to display the Expression Editor, and modify the expression.

6. Validate and save your changes to the repository.

Notes

- 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.
  3. If you have more than two hierarchy levels, then map the other columns from the appropriate source columns.

7.2.1.2 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. As installed out-of-the-box, the organization ID (that is set by the $$MASTER_ORG parameter) is set to 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.

To set the Master Inventory Organization in Product dimension Extract:

1. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
2. Select a container from the drop down list to the right of the Execute button.
3. Display the Tasks tab, and select the SDE_ORA_ProductDimension_Derive task.
4. Specify or create the $$MASTER_ORG parameter with an appropriate value.

   For example, $$MASTER_ORG=204.
5. Save your changes.

7.2.1.3 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 Siebel Customer-Centric Enterprise Warehouse Data Model Reference.

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 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 Siebel Customer-Centric Enterprise Warehouse Data Model Reference. 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 7–6 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_`ora.`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_codes_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 file_group_acct_names_ora.csv 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.

7.2.1.4 How to do Group Account correction for Oracle Application

Note: Refer to the section Section 7.2.1.3, "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 Business Intelligence 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 Client, 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 Tags 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 the 'Financials – Group Account Number Cleanup' subject area and do the following:

- If in step 3, 'Financials – Calculate GL Balance' is marked as 'inactive', then the Configuration Tags '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 Tags '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

7.2.2 Configuration Steps for Controlling Your Data Set for Oracle EBS

This section contains additional configuration steps that apply to Oracle Business Intelligence Applications deployed with Oracle EBS source systems, and contains the following topics:

- Section 7.2.2.1, "Configuring the Country Region and State Region Name"
- Section 7.2.2.2, "Configuring the State Name"
- Section 7.2.2.3, "Configuring the Country Name"
- Section 7.2.2.4, "Configuring the Configuring the Make-Buy Indicator"

7.2.2.1 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 5.10, "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 mapplet you want to edit.

   The following is a list of all Source Adapter mapplets that use the EXT_ COUNTRY_REGION column:

   - mplt_SA_ORA_SupplierDimension
   - mplt_SA_ORA_BusinessLocationDimension_Plan
   - mplt_SA_ORA_BusinessLocationDimension_StorageLocation
3. Double-click the Expression transformation to open the Edit Transformations dialog, 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 mapplet, 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 mapplet you want to edit.

   The following is a list of all Source Adapter mapplets that use the EXT_STATE_ REGION column:

   - mplt_SA_ORA_SupplierDimension
   - mplt_SA_ORA_BusinessLocationDimension_Plan
   - mplt_SA_ORA_BusinessLocationDimension_StorageLocation
3. Double-click the Expression transformation to open the Edit Transformations dialog, 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 mapplet, and save your changes to the repository.

7.2.2.2 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 mapplet 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 5.10, "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, 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.

7.2.2.3 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 5.10, "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, 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.

7.2.2.4 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, 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.

7.3 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 7.3.1, "Configuration Required Before A Full Load for PeopleSoft"
- Section 7.3.2, "Configuration Steps for Controlling Your Data Set for PeopleSoft"

7.3.1 Configuration Required Before A Full Load for PeopleSoft

Not applicable to Oracle Business Intelligence Applications Release 7.9.5.

7.3.2 Configuration Steps for Controlling Your Data Set for PeopleSoft

This section contains additional configuration steps that apply to PeopleSoft, and contains the following topics:

- Section 7.3.2.1, "Internal Organization Hierarchy Extraction Customization"

7.3.2.1 Internal Organization Hierarchy Extraction Customization

Internal Organization Hierarchy extracts the hierarchy related to Business Units. The PeopleSoft Tree Name and SETID are the only settings that require modification.

Table 7–7 Internal Organization 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 other 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. In Informatica PowerCenter Designer, create a new mapping by copying the existing SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_ Extract mapping.

   Name the new mapping SDE_PSFT_Stage_InternalOrganizationDimension_ CompanyHierarchy2_Extract.

2. Open the new mapping SDE_PSFT_Stage_InternalOrganizationDimension_ CompanyHierarchy2_Extract and edit 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. Save the changes in the Informatica Folder.

4. In Informatica PowerCenter Workflow Manager, add Corresponding Sessions and Workflows, as follows:

   a. In Informatica PowerCenter Workflow Manager, locate the appropriate PeopleSoft folder SDE_PSFT_xx_adapter (for example, 'SDE_PSFT_89_ adapter', 'SDE_PSFT_90_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 the newly created session.

   d. Save your changes.

   e. In Informatica PowerCenter Workflow Manager, create a Workflow with the same name as the Session.

f. Drag the newly created session to the panel and connect the start with the session.

5. Save and Check-in all your changes in Informatica PowerCenter Designer and Informatica PowerCenter Workflow Manager.

6. Re-configure the DAC settings, as follows:

   a. Login into the DAC, open your container, and display the Tasks tab.

   b. Add the workflow that you created as a new task.

      For information about what values to set task property values to, refer to the task SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Extract.

   c. Right click on each task and select 'Synchronize tasks'.

      The Synchronize tasks process reads in the source table and target table information for these tasks from the Informatica repository. To see this information, display the 'Source Tables' tab and 'Target Tables' tab for these tasks.

   d. Add your tasks into Configuration Tags: navigate to Configuration Tags panel (click 'Design', choose 'Configuration Tags'). Select 'PeopleSoft - Populate Business Unit Related Hierarchies' item. Select 'Tasks' tab in the panel below. Click 'Add/Remove' button; query the new mappings you just entered into the DAC. Select mappings; add them into the right panel. Click OK, then click Save.

   e. Reassemble the Financial or Human Resources 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) or human resources subject areas (HR - Event Information, HR - Operation, HR - Payroll), right click on it, and choose Assemble.

   f. 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) or PeopleSoft Human Resources Application (for example, Human Resources - PeopleSoft 8.8), right click on it, and choose Build.

7.4 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 7.4.1, "Configuration Required Before A Full Load for Siebel Source Systems"
- Section 7.4.2, "Configuration Steps for Controlling Your Data Set for Siebel Source Systems"

7.4.1 Configuration Required Before A Full Load for Siebel Source Systems

Not applicable to Oracle Business Intelligence Applications Release 7.9.5.

7.4.2 Configuration Steps for Controlling Your Data Set for Siebel Source Systems

Not applicable to Oracle Business Intelligence Applications Release 7.9.5.

8

Configuring Oracle Procurement and Spend Analytics

This chapter describes how to configure Oracle Procurement and Spend Analytics for particular sources to meet your business needs.

It contains the following topics:

- Section 8.1, "Overview of Oracle Procurement and Spend Analytics"
- Section 8.2, "Configuration Required Before A Full Load for Oracle Procurement and Spend Analytics"

8.1 Overview of Oracle Procurement and Spend Analytics

Oracle Procurement and Spend Analytics comprises the following:

- Oracle Procurement and Spend Analytics (for more information, see Section 8.1.1, "Oracle Procurement and Spend Analytics Module").
- Oracle Supplier Performance Analytics (for more information, see Section 8.1.2, "Supplier Performance Analytics Module").

Oracle Procurement and Spend Analytics enable organizations to optimize their supply chain performance by integrating data from across the enterprise supply chain and enabling executives, managers, and frontline employees to make more informed and actionable decisions. Organizations using Oracle Procurement and Spend Analytics benefit from increased visibility into the complete Procurement and Spend process, including comprehensive supplier performance analysis and supplier payables analysis. Through complete end-to-end insight into the factors that impact Procurement and Spend performance, organizations can significantly reduce costs, enhance profitability, increase customer satisfaction, and gain competitive advantage. Oracle Procurement and Spend Analytics also integrate with the other applications in Oracle Business Intelligence Applications product line. They deliver this insight across the organization to increase the company's effectiveness in managing its customers, suppliers, and financial decisions.

8.1.1 Oracle 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 Oracle 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.

8.1.2 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.

8.2 Configuration Required Before A Full Load for Oracle Procurement and Spend Analytics

This section contains configuration steps that you need to perform on Oracle Procurement and Spend Analytics before you do a full data load, and contains the following topics:

- Section 8.2.1, "Configuration Steps for Oracle Procurement and Spend Analytics For All Source Systems"
- Section 8.2.2, "Configuration Steps for Oracle Procurement and Spend Analytics For Oracle EBS"
- Section 8.2.3, "Configuration Steps for Oracle Procurement and Spend Analytics For PeopleSoft"
- Section 8.2.4, "Configuration Steps for Oracle Procurement and Spend Analytics For Universal"
- Section 8.2.5, "Configuration Steps for Controlling Your Data Set"

8.2.1 Configuration Steps for Oracle Procurement and Spend Analytics 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 8.2.1.1, "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 7, "Configuring Common Areas and Dimensions").

8.2.1.1 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 Business Intelligence 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.

8.2.2 Configuration Steps for Oracle Procurement and Spend Analytics 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 8.2.2.1, "How to Configure the DAC parameter for Purchase Cycle Line"
- Section 8.2.2.2, "Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics"

8.2.2.1 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. In the DAC Client, display the Design view, then display the Tasks tab, and query for task SIL_PurchaseCycleLinesFact.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
2. Display the Parameters tab in the lower pane. 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
3. 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*.

8.2.2.2 Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics

If you have modify or extend a seeded list of values, you must configure the CSV files for Oracle Procurement and Spend 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.

8.2.2.2.1 List of Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics

The table below lists the CSV worksheet files and the domain values for Oracle Procurement and Spend Analytics in the `$pmserver\LkpFiles` folder.

Table 8–1 Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics

Worksheet File Name	Description	Session
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 Section 8.2.2.2.2, "To configure domainValues_PaymentType_ora12.csv".	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 Section 8.2.2.2.9, "To configure domainValues_ShipmentType_ora12.csv".	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 Section 8.2.2.2.3, "To configure domainValues_Status_Purch_Approve_ora11i.csv".	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 Section 8.2.2.2.4, "To configure domainValues_Status_Purch_Cycle_ora11i.csv".	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 Section 8.2.2.2.8, "To configure domainValues_Xact_Types_PO_Line_Type_CODE1_ora11i.csv".	SDE_ORA_TransactionTypeDimension_PO_Line_Type
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 Section 8.2.2.2.7, "To configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv".	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 Section 8.2.2.2.5, "To configure domainValues_Xact_Types_Purch_Orders_ora11i.csv".	SDE_ORA_TransactionTypeDimension_PurchaseOrder

Table 8–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics

Worksheet File Name	Description	Session
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 Section 8.2.2.2.6, "To configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv".	SDE_ORA_TransactionTypeDimension_PurchaseRequest

8.2.2.2.2 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 $pmserver\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.

8.2.2.2.3 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.

8.2.2.2.4 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 $pmserver\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.

8.2.2.2.5 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 $pmserver\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.

8.2.2.2.6 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 $pmserver\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.

8.2.2.2.7 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 $pmserver\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.

8.2.2.2.8 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 $pmserver\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.

8.2.2.2.9 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 $pmserver\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.

8.2.3 Configuration Steps for Oracle Procurement and Spend Analytics For PeopleSoft

This section contains configuration steps that apply to PeopleSoft before you do a full data load.

Not applicable to Oracle Business Intelligence Applications Release 7.9.5.

8.2.4 Configuration Steps for Oracle Procurement and Spend Analytics For Universal

This section contains configuration steps that apply to Universal before you do a full data load.

Not applicable to Oracle Business Intelligence Applications Release 7.9.5.

8.2.5 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Procurement and Spend Analytics.

- Section 8.2.5.1, "Configuration Steps for Oracle Procurement and Spend Analytics For All Source Systems"
- Section 8.2.5.2, "Configuration Steps for Oracle Procurement and Spend Analytics For Oracle EBS"
- Section 8.2.5.3, "Configuration Steps for Oracle Procurement and Spend Analytics For PeopleSoft"
- Section 8.2.5.4, "Configuration Steps for Oracle Procurement and Spend Analytics For Universal"

8.2.5.1 Configuration Steps for Oracle Procurement and Spend Analytics For All Source Systems

This section contains configuration steps that apply to all source systems, and contains the following topics:

- Section 8.2.5.1.1, "About Configuring the Purchase Receipts Aggregate Table"
- Section 8.2.5.1.2, "How to Configure the Purchase Receipts Aggregate Table"
- Section 8.2.5.1.3, "About Configuring the Purchase Cycle Lines Aggregate Table"
- Section 8.2.5.1.4, "How to Configure the Purchase Cycle Lines Aggregate Table"

8.2.5.1.1 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.

8.2.5.1.2 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 Purchase Receipts Aggregate Table

1. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Select the appropriate container for your OLTP source system.
3. Display the Tasks tab.
4. For each of the following tasks, display the Parameters tab and create the specified parameter name and parameter value
   - SIL_PurchaseReceiptAggregate_Derive_PreLoadImage $$TIME_GRAIN 'MONTH'
   - SIL_PurchaseReceiptAggregate_Derive_PreSoftDeleteImage $$TIME_GRAIN 'MONTH'
   - PLP_PurchaseReceiptAggregate_Derive_PostLoadImage $$TIME_GRAIN 'MONTH'
   - PLP_PurchaseReceiptAggregate_Load_Full $$GRAIN 'MONTH'

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.

8.2.5.1.3 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.

8.2.5.1.4 How to Configure the Purchase Cycle Lines Aggregate Table

Before you load the Purchase Cycle Lines aggregate table (`W_PURCH_CYCLE_LINE_A`), you need to configure the post-load-processing parameter run the initial workflow and then the incremental workflow, as follows.

To configure the Purchase Cycle Lines Aggregate Table

1. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Select the appropriate container for your OLTP source system.
3. Display the Tasks tab.
4. For each of the following tasks, display the Parameters tab and create the specified parameter name and parameter value

   - SIL_PurchaseCycleLinesAggregate_Derive_PreLoadImage $$TIME_GRAIN 'MONTH'
   - SIL_PurchaseCycleLinesAggregate_Derive_PreSoftDeleteImage $$TIME_GRAIN 'MONTH'

- PLP_PurchaseCycleLinesAggregate_Derive_PostLoadImage $$TIME_GRAIN 'MONTH'
- PLP_PurchaseCycleLinesAggregate_Load_Full $$GRAIN 'MONTH'

8.2.5.2 Configuration Steps for Oracle Procurement and Spend Analytics For Oracle EBS

This section contains configuration steps that apply to Oracle EBS, and contains the following topics:

- Section 8.2.5.2.1, "How to Extract Particular Purchase Order Records"

8.2.5.2.1 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 mapplet. 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 Informatica PowerCenter Designer, open the SDE_ORA115<*Ver*>_Adaptor.
2. Open the MPLT_BC_ORA_PURCHASEORDERFACT mapplet.
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 mapplet.

8.2.5.3 Configuration Steps for Oracle Procurement and Spend Analytics For PeopleSoft

This section contains configuration steps that apply to PeopleSoft.

Not applicable to Oracle Business Intelligence Applications Release 7.9.5.

8.2.5.4 Configuration Steps for Oracle Procurement and Spend Analytics For Universal

This section contains configuration steps that apply to Universal, and contains the following topics:

- Section 8.2.5.4.1, "About Configuring Oracle Procurement and Spend Analytics for Universal Source"
- Section 8.2.5.4.2, "How to Configure the Preferred Merchant Flag"
- Section 8.2.5.4.3, "How to Configure the Customer Billable Indicator"

- Section 8.2.5.4.4, "How to Configure the Receipts Indicator"
- Section 8.2.5.4.5, "How to Configure Expense Payment Types"
- Section 8.2.5.4.6, "How to Configure the Default Expense Distribution Percentage"
- Section 8.2.5.4.7, "How to Configure Lookup Dates for Currency Conversion"

8.2.5.4.1 About Configuring Oracle Procurement and Spend 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.

8.2.5.4.2 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 Informatica 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.

8.2.5.4.3 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 Informatica 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.

8.2.5.4.4 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 Informatica 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.

8.2.5.4.5 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 Informatica 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.

8.2.5.4.6 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 Informatica 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.

8.2.5.4.7 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 Informatica 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.

9

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 9.1, "Overview of Oracle Financial Analytics"
- Section 9.2, "Configuration Required Before A Full Load for Financial Analytics"

9.1 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.

9.2 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 9.2.1, "Configuration Steps for Financial Analytics For All Source Systems"
- Section 9.2.2, "Configuration Steps for Financial Analytics For Oracle EBS"
- Section 9.2.3, "Configuration Steps for Financial Analytics For PeopleSoft"
- Section 9.2.4, "Configuration Steps for Financial Analytics For Universal"
- Section 9.2.5, "Configuration Steps for Controlling Your Data Set"

9.2.1 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 7, "Configuring Common Areas and Dimensions").

9.2.2 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 9.2.2.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics"
- Section 9.2.2.2, "How to configure domainValues_Xact_Types_DocTypes_ora12.csv for Oracle General Ledger and Profitability Analytics"
- Section 9.2.2.3, "How to configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics"

- Section 9.2.2.4, "How to Specify the Ledger or Set of Books for which GL Data is Extracted"

9.2.2.1 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 sections explain 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.

Typically, the number of rows returned by the source system will not match the number of rows provided in the domain valuesXXX.csv files. If so, you need to delete extra rows or insert additional rows in the CSV file to accommodate all the values returned by the source system in context. You also need to determine which row should have 'Return' and which row should have 'Delivery' in the W_XACT_TYPE_ CODE column.

Table 9–2 lists the CSV worksheet files and the domain values for Financial Analytics and Oracle EBS in the $pmserver\LkpFiles folder.

Table 9–1 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 Section 9.2.2.2, "How to configure domainValues_Xact_ Types_DocTypes_ora12.csv for Oracle General Ledger and Profitability Analytics").	SDE_ORA_ TransactionTypeDimensio n_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 Section 9.2.2.3, "How to configure domainValues_Xact_ Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics").	SDE_ORA_ TransactionTypeDimensio n_GLCOGSDerive

9.2.2.2 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 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_ora12.csv file using a text editor in the $pmserver\lkpfiles 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 PowerCenter Designer, open the mapping SDE_ORA_GLCOGSFact in the folder SDE_ORAxxx_Adaptor.

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 PowerCenter Workflow Manager to the SQL Override in the full load session SDE_GL_COGSFACT_Full.

9.2.2.3 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. Oracle Inventory Application is a data source from which Oracle General Ledger and Profitability Analytics extract data.

2. Open the domainValues_Xact_Types_DocTypes_ora11i.csv file using a text editor in the $pmserver\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 PowerCenter Designer, open the mapping SDE_ORA_GLCOGSFact in the folder SDE_ORAxxx_Adaptor.
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 PowerCenter Workflow Manager to the full load sessions that correspond to the mapping SDE_ORA_GLCOGSFact_Full.

9.2.2.4 How to Specify the Ledger or Set of Books for which GL Data is Extracted

If you have an Oracle EBS source system, you can specify from which set of books or ledgers you extract the GL data.

Oracle Business Intelligence Applications enables you to configure:

- The list of ledgers or set of books from which to extract the GL data.
- The types of ledgers or set of books for which 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 Business Intelligence 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. In the DAC Client, display the Design view, and open the source container that you want to configure by selecting a container from the drop down list to the right of the Execute button.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

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_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, do the following:

1. In the DAC Client, display the Design view, and open the source container that you want to configure by selecting a container from the drop down list to the right of the Execute button.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
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 EBS R12 customers, to configure the list of ledgers to extract the GL data for, do the following:

1. In the DAC Client, in the Design view, open the source container that you want to configure by selecting a container from the drop down list to the right of the Execute button.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
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 EBS 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. In the DAC Client, in the Design view, open the source container that you want to configure by selecting a container from the drop down list to the right of the Execute button.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
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.

9.2.3 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 9.2.3.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics"
- Section 9.2.3.2, "How to configure Domain Values for Oracle Receivables Analytics"
- Section 9.2.3.3, "How to insert beginning GL Balances into Oracle Business Intelligence Applications"
- Section 9.2.3.4, "GL Account Dimension, Chartfields and Hierarchy for PeopleSoft"
- Section 9.2.3.5, "Group Account Configuration"

9.2.3.1 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 9.2.3.2, "How to configure Domain Values for Oracle Receivables Analytics" 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 9–2 lists the CSV worksheet files and the domain values for Financial Analytics and PeopleSoft in the $pmserver\LkpFiles folder.

Table 9–2 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 Section 9.2.3.2, "How to configure Domain Values for Oracle Receivables Analytics").	SDE_PSFT_TransactionTypeDimension_ARDerive

9.2.3.2 How to configure Domain Values for Oracle Receivables Analytics

This section explains how to configure domain values for Oracle Receivables Profitability Analytics.

To configure domain values for Oracle Receivables Analytics:

1. Identify the Entry Types in your Receivables Application by using the following SQL:

   ```
   SELECT DISTINCT ENTRY_TYPE, DESCR FROM PS_ENTRY_TYPE_TBL
   ```

   This query gives the transaction type codes in your Oracle Inventory Application and their corresponding descriptions. Oracle Inventory Application is a data source from which Oracle General Ledger and Profitability Analytics extract data.

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 PowerCenter Designer, open mapplets mplt_BC_PSFT_ARTransactionFact_Item and mplt_BC_PSFT_ARTransactionFact_ItemActivity in the SDE_PSFTxxx_Adaptor 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 PowerCenter Workflow Manager to the full load sessions that correspond to the SDE_PSFT_ARTransactionFact_Item_Full and SDE_PSFT_ARTransactionFact_ItemActivity_Full mappings.

9.2.3.3 How to insert beginning GL Balances into Oracle Business Intelligence Applications

Before you run the initial ETL, make sure that you set the Source System Parameters in order to compute the GL Balances correctly, as follows:

1. In the DAC Client, display the Design View, then display the Source System Parameters tab.

   For information about how to log into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Set the following parameters:

   - Set the parameter $$INITIAL_EXTRACT_DATE to the date from which you want to extract data from your PeopleSoft application to the data warehouse. Ensure that this Parameter is the beginning of an accounting period and not just any date in the middle of a period. For example, if you decide to extract data from June 2005, and the June 2005 accounting period starts from 5th June, set the date to 5th June, 2005.
   - Setup the parameter $$YEAR to the same as the fiscal year of $$INITIAL_EXTRACT_DATE parameter above. For example, if June 2005 is fiscal year 2005, set this parameter to 2005.

- Set the parameter $$ACCOUNTING_PERIOD to the accounting period number of the $$INITIAL_EXTART_DATE parameter above. For example, if June 2005 is the 6th period of fiscal year 2005, set this parameter to 6.

9.2.3.4 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>\dwrep\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 9–3 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.

9.2.3.5 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 Fusion Edition Data Model Reference*. 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 9–4 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.

9.2.4 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 Business Intelligence Applications Release 7.9.5.

9.2.5 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Financial Analytics, and contains the following topics:

- Section 9.2.5.1, "Configuration Steps for Financial Analytics For All Source Systems"
- Section 9.2.5.2, "Configuration Steps for Financial Analytics For Oracle EBS"
- Section 9.2.5.3, "Configuration Steps for Financial Analytics For PeopleSoft"
- Section 9.2.5.4, "Configuration Steps for Financial Analytics Chain For Universal"

9.2.5.1 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 9.2.5.1.1, "How to include UnApplied Payments in AR Aging Calculations"
- Section 9.2.5.1.2, "How to Configure the AP/AR Aging Tables"

9.2.5.1.1 How to include UnApplied Payments in AR Aging Calculations To include UnApplied Payments in Aging Calculations for AR Aging tables

1. In Informatica Designer, open the mapping PLP_ARSnapshotInvoiceAging in the PLP Folder.
2. Edit the Source Qualifier transformation SQ_IA_AR_XACTS, and open the SQL Query in the Properties tab.
3. In the WHERE clause of the SQL statement, change the SQL text from:

   ```
   W_XACT_TYPE_D.W_XACT_SUBTYPE_CODE IN ('INVOICE','CR MEMO','DR MEMO') AND
   ```

   To:

   ```
   W_XACT_TYPE_D.W_XACT_SUBTYPE_CODE IN ('INVOICE','CR MEMO','DR MEMO', 'PAYMENT')
   AND
   ```

4. Save the changes.

   The next ETL will populate the Aging tables using UnApplied payments in the calculations.

9.2.5.1.2 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.

9.2.5.2 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 9.2.5.2.1, "How to Configure How GL Balances Is Populated in Oracle EBS"
- Section 9.2.5.2.2, "How to Configure Oracle Profitability Analytics Transaction Extracts"
- Section 9.2.5.2.3, "How to Configure Cost Of Goods Extract (Oracle EBS 11i-specific)"
- Section 9.2.5.2.4, "How to Configure AP Balance ID for Oracle Payables Analytics"
- Section 9.2.5.2.5, "How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics"
- Section 9.2.5.2.6, "How to Configure the AR Adjustments Extract for Oracle Receivables Analytics"
- Section 9.2.5.2.7, "How to Configure the AR Schedules Extract"
- Section 9.2.5.2.8, "How to Configure the AR Cash Receipt Application Extract for Oracle Receivables Analytics"
- Section 9.2.5.2.9, "How to Configure the AR Credit-Memo Application Extract for Oracle Receivables Analytics"
- Section 9.2.5.2.10, "How to Setup Drill Down in Oracle BI Answers from General Ledger to Subledger"

9.2.5.2.1 How to Configure How GL Balances Is Populated in Oracle EBS

Oracle Business Intelligence 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 the DAC Client, for the Subject Area 'Financials – General Ledger', in the 'Configuration Tags' 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 the DAC Client, for the Subject Area 'Financials – General Ledger', in the 'Configuration Tags' 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.

9.2.5.2.2 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 Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. In Mapplet Designer, open the `mplt_BC_ORA_GLRevenueFact` mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations dialog, 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.

9.2.5.2.3 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 mapplet. You need to modify the Business Component mapplet for both the regular extract mapplet as well as the primary extract mapplet. 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 Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. In Mapplet Designer, open `mplt_BC_ORA_GLCOGSFact`.
3. Double-click the Source Qualifier to open the Edit Transformations dialog, 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 `mplt_BC_ORA_GLCOGSFact_Primary`.
5. Double-click the Source Qualifier to open the Edit Transformations dialog, 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 Informatica PowerCenter 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.

9.2.5.2.4 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 Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. In Mapplet Designer, open the mapplet (for example, mplt_SA_ORA_ APTransactionFact_LiabilityDistribution).
3. Double-click the Expression transformation to open the Edit Transformations dialog.
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.

9.2.5.2.5 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 Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
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 dialog.
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.

9.2.5.2.6 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 Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. In Mapplet Designer, open the `mplt_BC_ORA_ARTransactionFact_Adjust` mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations dialog, 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.

9.2.5.2.7 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 Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. In Mapplet Designer, open the `mplt_BC_ORA_ARTransactionFact_ARSchedules` mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations dialog.

   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.

9.2.5.2.8 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 Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).

2. In Mapplet Designer, open the `mplt_BC_ORA_ARTransactionFact_ReceivableApplication` mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations dialog, 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_ReceivableApplicationPrimary mapplet.

9.2.5.2.9 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 Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. In Mapplet Designer, open the `mplt_BC_ORA_ARTransactionFact_CreditmemoApplication` mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations dialog, 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.

9.2.5.2.10 How to Setup Drill Down in Oracle BI Answers from General Ledger to Subledger

Oracle Business Intelligence 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 for 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.

9.2.5.3 Configuration Steps for Financial Analytics For PeopleSoft

This section contains additional configuration steps that apply to PeopleSoft, and contains the following topics.

- Section 9.2.5.3.1, "About PeopleSoft Trees in Oracle Financial Analytics"
- Section 9.2.5.3.2, "GL ChartField Hierarchy and Internal Organization Hierarchy Customization Process"
- Section 9.2.5.3.3, "General Ledger Chartfield 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 Business Intelligence Applications can be read as Financial Analytics.

9.2.5.3.1 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 account hierarchies, Profit Center hierarchies, Cost Center 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 Chartfields (Account, Dept, and Project, and so on), items, locations, and so on, are organized into user-defined trees.

Table 9–5 lists the default PeopleSoft Trees that are sourced by Oracle Financial Analytics.

Table 9–5 PeopleSoft Trees for Oracle 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

Table 9–5 (Cont.) PeopleSoft Trees for Oracle Financial Analytics

PeopleSoft Data Structure	Repository Source Name	Description
PeopleSoft Tree	OPERUNIT	Operating Unit

Table 9–6 lists the mappings and sessions using PeopleSoft Trees.

Table 9–6 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

9.2.5.3.2 GL ChartField Hierarchy and Internal Organization Hierarchy Customization Process

Oracle Business Intelligence Applications provides a set of mechanisms to support various tree structures in PeopleSoft applications. The following tables illustrate the supported structure types and level types.

Table 9–7 Structure Types

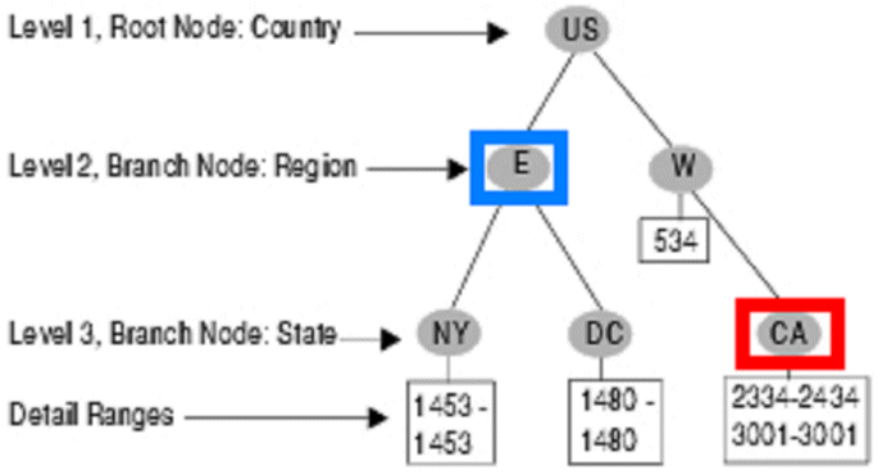
Structure Type	Support
Winter Tree	Y
Detail Tree	Y
Summary Tree	Y

Table 9–8 Level Types

Level Type	Support
Strict-level Tree	Y
Loose-level Tree	Y

To support above the functionality in an efficient way, a flattened structure is adopted to store the hierarchy-related data. Each record of W_HIERARCHY_D and other hierarchy-related dimension tables (e.g. W_INT_ORG_DH) represents a single node in the tree. It stores the path information from the top-most node to that node. For the bottom nodes in the detail tree, the extraction process generates a number of records according to the node's range and the record in the relevant table, and put those records into the hierarchy table. The following examples provide a detailed explanation of this functionality.

Figure 9–1 Example GL Chartfield hierarchy



For any nodes in the tree, except for the bottom nodes, W_HIERARCHY_D will have one corresponding record. For example, for the 'E' node in above tree (the one highlighted with a blue square), you will have the following record in the database.

Table 9–9 Example of how the node E is stored in the W_HIERARCHY_D table.

HIER1_CODE	HIER2_CODE	HIER3_CODE	HIER4_CODE	HIER[5-20]_ 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 levels, then the levels 6~20 will repeat the value of level 5.

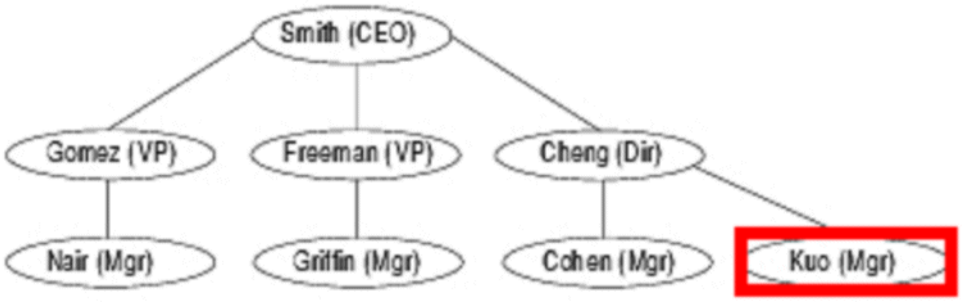
For the leaf node in the detail tree shown above, W_HIERARCHY_D will have a record for every existing record within this detail range (the extraction process will find the single record in the relevant table according to the tree's structure). For example, in the CA node (the one with red square), the range is 2334-2434 and 3001-3001. The tree structure table (for example, State_TBL) contains two records: 2340, 3001. When generating the CA node, the extraction process will extract these two records, put them into W_HIERARCH_D as following structure.

Table 9–10 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
US	W	CA	2334	2334

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 9–2 The Winter Loose-level Tree



The record 'Kuo' is stored in the database as follows:

Table 9–11 Example CSV Chartfield Mapping Values

HIER1_CODE	HIER2_CODE	HIER3_CODE	HIER4_CODE	HIER[5-20]_ CODE
Smith	Cheng	Kuo	Kuo	Kuo

Oracle Business Intelligence Applications uses the following temporary tables and staging tables to handle the extraction and load process of the tree. They are standard and not tree-specific or tree-structure specific.

This table below lists the tables used during the General Ledger Hierarchy extraction process. The sequence indicates the orders in which these tables are populated.

Table 9–12 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 9–13 Internal Organization Hierarchy

Sequence	Table Name
1	W_PSFT_INT_ORG_TREE_TMP
1	W_PSFT_INT_ORG_GLBU_DH_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 Business Intelligence 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 activate or de-activate a set of tasks in your ETL process:

1. In the DAC Client, in the Design view, open the corresponding PeopleSoft container by selecting a container from the drop down list to the right of the Execute button.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
2. Display the Configuration Tags tab.
3. Clear the Inactive checkbox of the hierarchy you want to enable, as follows:
   - For GL Chartfield, clear the Inactive checkbox for PeopleSoft - Populate Non Business Unit Related Hierarchies.
   - For Business Unit, clear the Inactive checkbox for PeopleSoft – Populate Business Unit Related Hierarchies.
4. Click Save.
5. Display the Subject Areas tab.
6. For each Financials subject area, do the following:
   a. Select the Financials subject area in the Name list.

      For example, Cost of Goods Sold, General Ledger, Payables and so on.
   b. In the bottom pane, display the Configuration Tags tab.
   c. Clear the Inactive checkbox of the hierarchy you want to enable, as follows:

      - For GL Chartfield, clear the Inactive checkbox for PeopleSoft - Populate Non Business Unit Related Hierarchies.

      - For Business Unit, clear the Inactive checkbox for PeopleSoft – Populate Business Unit Related Hierarchies.
   d. Save the changes.
7. Click Assemble to redesign the subject areas that you have modified.

8. Display the Execute tab at the left hand side.
9. For every execution plan which contains the subject areas that you modified, rebuild those execution plans.
10. Save the execution plans that you modified.

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 naming parameters.

9.2.5.3.3 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 SDE_PSFT_Stage_GLHierarchy_AccountHierarchy 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 9–14 General Ledger Chartfield Hierarchy Mappings

Execution 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. In Informatica PowerCenter Designer, 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. Open the mapping SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract and edit the settings 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. Open the mapping SDE_PSFT_Stage_GLHierarchy_GetDepartmentChartField and edit the settings as follows:

   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 PowerCenter Workflow Manager, as follows:

   a. In Informatica PowerCenter Workflow Manager, locate the appropriate PeopleSoft folder (for example, SDE_PSFT_xx_Adaptor').

   b. 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 9–15 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

Table 9–15 (Cont.) General Ledger Chartfield Hierarchy Mappings

Tab Name	Parameter Name	Value
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.

c. Save your changes

d. In Informatica PowerCenter Workflow Manager, create a Workflow with the same name as Session.

e. Drag the newly created session to the panel.

f. Connect the start with the session.

6. Save and Check-in all your changes in Informatica PowerCenter Designer and Informatica PowerCenter Workflow Manager.

7. Re-configure the DAC settings, as follows:

a. Login into the DAC, open your container, and display the Tasks tab.

b. Add the two workflows that you created as new tasks.

For information about what values to set task property values to, refer to the tasks SDE_PSFT_Stage_GLHierarchy_AccountHierarchy_Extract and SDE_ PSFT_Stage_GLHierarchy_GetAccountChartField.

c. Right click on each task and select 'Synchronize tasks'.

The Synchronize tasks process reads in the source table and target table information for these tasks from the Informatica repository. To see this information, display the 'Source Tables' tab and 'Target Tables' tab for these tasks.

d. Add your tasks into Configuration Tags: navigate to Configuration Tags panel (click 'Design', choose 'Configuration Tags'). 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 the DAC. Select mappings; add them into the right panel. Click OK, then click Save.

e. 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.

f. 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.

9.2.5.4 Configuration Steps for Financial Analytics Chain For Universal

This section contains additional configuration steps that apply to Universal, and contains the following topics:

- Section 9.2.5.4.1, "About the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics"
- Section 9.2.5.4.2, "How to Configure the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics"
- Section 9.2.5.4.3, "How to Setup Drill Down in Oracle BI Answers from General Ledger to Subledger"

9.2.5.4.1 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 Business Intelligence 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.

9.2.5.4.2 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_ora11i.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.

9.2.5.4.3 How to Setup Drill Down in Oracle BI Answers from General Ledger to Subledger

Oracle Business Intelligence 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.

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.

9.2.5.4.4 How to configure Budgets for Oracle Profitability Analytics

If you are implementing Oracle Profitability Analytics and would like to view the budget data in your dashboards, you need to import budget data into your data warehouse, as described in this section.

About Configuring Universal Source Files

The following tables describe the columns in the universal source CSV files file_budget.csv and file_acct_budget.csv, their data types, and how to populate them where applicable.

The table below shows the structure of the file_budget.csv file. The records in the file_acct_budget.csv will be loaded into W_BUDGET_D.

Table 9–16 Universal Source for Budget Fact (file_budget.csv)

Column Name	Datatype	Size	Description
BUDGET_NAME	string	80	Budget name.
BUDGET_VERSION	string	30	Budget version.
BUDGET_STATUS	string	30	Budget status.
BUDGET_TYPE	string	30	Budget type.
CREATED_BY_ID	string	80	ID of created by user. Populate with Integration_ID from w_user_d.
CHANGED_BY_ID	string	80	ID of changed by user. Populate with Integration_ID from w_user_d.
CREATED_ON_DT	string	14	Created date.
CHANGED_ON_DT	string	14	Changed date. Used for updating an existing record in the warehouse. Increase the date if you want to update the record. If a record with the same integration_ID already exists in the target table W_BUDGET_D, then the load process will compare the CHANGED_ON_DT values between this record and the record in W_BUDGET_D. If this record's CHANGED_ON_DT is later than the record in W_BUDGET_D, then the load process will perform an update against the record in W_BUDGET_D; otherwise the load process will ignore this record, and no update or insertion will occur. If there is no matching record in W_BUDGET_D with the same integration_ID, then the load process will insert this record into W_BUDGET_D.
AUX1_CHANGED_ON_DT	string	14	
AUX2_CHANGED_ON_DT	string	14	
AUX3_CHANGED_ON_DT	string	14	
AUX4_CHANGED_ON_DT	string	14	
DELETE_FLG	string	1	
DATASOURCE_NUM_ID	number	10	A number for your data source. Populate the same datasource_num_id as your main source application.
INTEGRATION_ID	string	80	A unique identifier for the record.
TENANT_ID	string	80	
X_CUSTOM	string	10	

The table below shows the structure of the file_acct_budget.csv file. The records in the file_acct_budget.csv will be loaded into W__ACCT_BUDGET_F.

Table 9–17 Universal Source for Budget Fact (file_acct_budget.csv)

Column Name	Datatype	Size	Description
GL_ACCOUNT_ID	string	80	GL Account identifier. Populate with integration_id from w_gl_account_d.
PROJECT_ID	string	80	
PRODUCT_ID	string	80	Product identifier. Populate with integration_id from w_product_d.
COMPANY_ORG_ ID	string	80	Company Org identifier. Populate with integration_id from w_int_org_d where company_flg = Y.
BUDG_BUSN_ AREA_ORG_ID	string	80	Company Org identifier. Populate with integration_id from w_int_org_d where business_area_flg = Y.
BUDG_FIN_AREA_ ORG_ID	string	80	Company Org identifier. Populate with integration_id from w_int_org_d where fin_area_ flg = Y.
BUDG_CTRL_ AREA_ORG_ID	string	80	Company Org identifier. Populate with integration_id from w_int_org_d where ctrl_ area_flg = Y.
COST_CENTER_ID	string	80	Cost Center identifier. Populate with integration_ id from w_cost_center_d.
PROFIT_CENTER_ ID	string	80	Profit Center identifier. Populate with integration_id from w_profit_center_d.
BUDGET_ID	string	80	Populate with the value from integration_id in file_budget.csv
POSTED_ON_DT	string	14	A date on which this transaction can be reported.
PERIOD_BEGIN_DT	string	14	
PERIOD_END_DT	string	14	Populate with your budget period's end date. If your budget is monthly, populate with the month end date.
BUDGET_DOC_ AMT	number	22	Budget amount in document currency.
BUDGET_LOC_ AMT	number	22	Budget amount in local currency.
BUDGET_GRP_ AMT	number	22	
DOC_CURR_CODE	string	30	Document currency code.
LOC_CURR_CODE	string	30	Local currency code.
GRP_CURR_CODE	string	30	
CREATED_BY_ID	string	80	ID of created by user. Populate with Integration_ ID from w_user_d.
CHANGED_BY_ID	string	80	ID of changed by user. Populate with Integration_ID from w_user_d.
CREATED_ON_DT	string	14	Created date.

Table 9–17 (Cont.) Universal Source for Budget Fact (file_acct_budget.csv)

Column Name	Datatype	Size	Description
CHANGED_ON_DT	string	14	Changed date. Used for updating an existing record in the warehouse. Increase the date if you want to update the record. If a record with the same integration_ID exists in the target table W_ACCT_BUDGET_F already, then the load process will compare the CHANGED_ON_DT values between this record and the record in W_ACCT_BUDGET_F. If this record's CHANGED_ON_DT is later than the record in W_ACCT_BUDGET_F, then the load process will perform an update against the record in W_ACCT_BUDGET_F; otherwise it'll ignore this record, no update or insertion will occur. If there's no matching record in W_ACCT_BUDGET_F with the same integration_ID, then the load process will insert this record into W_ACCT_BUDGET_F.
AUX1_CHANGED_ON_DT	string	14	
AUX2_CHANGED_ON_DT	string	14	
AUX3_CHANGED_ON_DT	string	14	
AUX4_CHANGED_ON_DT	string	14	
DELETE_FLG	string	1	
DATASOURCE_NUM_ID	number	10	A number for your data source. Populate the same datasource_num_id as your main source application.
INTEGRATION_ID	string	80	A unique identifier for the record.
TENANT_ID	string	80	
X_CUSTOM	string	10	

Note: date columns are to populated in the csv file as a number in YYYYMMDDHH24MISS format.

Use the following table to understand how the integration_id (key) of some of the key dimensions are constructed for the Oracle EBS source system. You can use this information to populate the dimension foreign key identifiers in the above universal source CSV file for budget fact, if you have to use budget fact in conjunction with dimensions populated from Oracle EBS.

Table 9–18 Populating the integration_id fields in Oracle EBS Source Systems

Field	How to populate
GL_ACCOUNT_ID (w_gl_account_d)	ccid
COMPANY_ORG_ID (w_int_org_d)	No need to populate; will be calculated based on GL Account ID.
COST_CENTER_ID(w_cost_center_d)	No need to populate; will be calculated based on GL Account ID.
PROFIT_CENTER_ID(w_profit_center_d)	No need to populate; will be calculated based on GL Account ID.

Table 9–18 (Cont.) Populating the integration_id fields in Oracle EBS Source Systems

Field	How to populate
LEDGER_ID(w_ledger_d)	For Oracle 11i populate as set of book ID For Oracle R12 populate as ledger ID.

How to import budget data into your data warehouse

1. Populate the file_budget.csv and file_acct_budget.csv files with your budget data.

   The CSV files are located in the OracleBI\dwrep\Informatica\Sample Universal Source Files\ directory of the Oracle Business Intelligence Applications installation machine.

   Refer the tables above for details of how to populate these files.

2. In the DAC Client, create a new execution plan using the subject area 'Financials – Budget' in the 'Universal' container.

3. Run the new execution plan that you created in the previous step.

   Note: This Execution Plan must be run after the regular execution plan to populate the data warehouse for the other subject areas has completed.

4. Load new Budget data or changes to existing budget data.

   Repeat Step 1 and Step 3 as needed to load new budget for the next fiscal period or make corrections to already loaded budget data.

10

Configuring Oracle Supply Chain and Order Management Analytics

This chapter describes how to configure Oracle Supply Chain and Order Management Analytics for particular sources to meet your business needs.

It contains the following topics:

- Section 10.1, "Overview of Oracle Supply Chain and Order Management Analytics"
- Section 10.2, "Configuration Required Before A Full Load for Oracle Supply Chain and Order Management Analytics"

10.1 Overview of Oracle Supply Chain and Order Management Analytics

The Oracle Supply Chain and Order Management Analytics application allows you to analyze:

- Bill of materials.
- Bookings.
- Financial and Operational Backlogs.
- Inventory held by an organization.
- Inventory movements in, out, and through manufacturing plants, distribution centers, or storage locations.
- Invoices.
- The movement of sales orders through different stages of the sales cycle.

The Oracle Supply Chain and Order Management Analytics application consists of orders, invoices, backlogs and inventory. 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. This coverage includes insight into which items are booked, backlogged, and invoiced. This allows you to evaluate the sales performance of individual sales representatives or departments. Oracle Supply Chain and Order Management Analytics application also provides you with information on Inventory Transactions, Inventory Balances, Bill of Materials and Customer and Supplier Returns. 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 following sources can populate Orders, Inventory and Revenue:

- Oracle 11i
- Oracle R12
- Universal source

The Oracle Supply Chain and Order Management Analytics application also requires post-load processing mappings to populate its aggregate and derived tables.

10.2 Configuration Required Before A Full Load for Oracle Supply Chain and Order Management Analytics

This section contains configuration steps that you need to perform on Oracle Supply Chain and Order Management Analytics before you do a full data load, and contains the following topics:

- Section 10.2.1, "Configuration Steps for Oracle Supply Chain and Order Management Analytics For All Source Systems"
- Section 10.2.2, "Configuration Steps for Oracle Supply Chain and Order Management Analytics For Oracle EBS"
- Section 10.2.3, "Configuration Steps for Oracle Supply Chain and Order Management Analytics For PeopleSoft"
- Section 10.2.4, "Configuration Steps for Oracle Supply Chain and Order Management Analytics For Universal"
- Section 10.2.5, "Configuration Steps for Controlling Your Data Set"

10.2.1 Configuration Steps for Oracle Supply Chain and Order Management 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 7, "Configuring Common Areas and Dimensions").

10.2.2 Configuration Steps for Oracle Supply Chain and Order Management Analytics For Oracle EBS

This section contains configuration steps required before you do a full data load that apply to Oracle EBS.

- Section 10.2.2.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics".
- Section 10.2.2.2, "How to Configure Invoice Type Domain Values".
- Section 10.2.2.3, "How to Configure Pick Types Domain Values".
- Section 10.2.2.4, "How to Configure Order Types Domain Values"
- Section 10.2.2.5, "How to Configure Pick Status Domain Values"
- Section 10.2.2.6, "How to Configure Invoice Status Domain Values"
- Section 10.2.2.7, "How to Configure Order Overall Status Domain Values"

- Section 10.2.2.8, "How to Configure Pay Method Domain Values"
- Section 10.2.2.9, "To Configure Movement Types Domain Values"
- Section 10.2.2.10, "How to Configure the Bill of Materials Explosion"
- Section 10.2.2.11, "How to Configure Quantity Types for Product Transactions"

10.2.2.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics

The table below lists the CSV worksheet files and the domain values for Oracle Supply Chain and Order Management Analytics in the `$pmserver\LkpFiles` folder.

Table 10–1 Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management 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 Section 10.2.2.2, "How to Configure Invoice Type Domain Values".	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 Section 10.2.2.3, "How to Configure Pick Types Domain Values".	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 Section 10.2.2.4, "How to Configure Order Types Domain Values".	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 Section 10.2.2.5, "How to Configure Pick Status Domain Values".	SDE_ORA_StatusDimension_SalesPickLines
domainValues_PayMethodCode_ora11i.csv	Lists the method code column and the corresponding domain value for the application.	SDE_ORA_PaymentMethodDimension

Table 10–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics

Worksheet File Name	Description	Session
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 Section 10.2.2.6, "How to Configure Invoice Status Domain Values".	SDE_ORA_StatusDimension_SalesInvoiceLine
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 Section 10.2.2.7, "How to Configure Order Overall Status Domain Values".	SDE_ORA_StatusDimension_SalesOrderLineCycle

For general information about domain values with CSV worksheet files, see Section 5.12, "About Domain Values" and Section 5.13, "Configuring the Domain Value Set with CSV Worksheet Files".

10.2.2.2 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.

10.2.2.3 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.

10.2.2.4 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 $pmserver\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.

10.2.2.5 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 $pmserver\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.

10.2.2.6 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 $pmserver\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.

10.2.2.7 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 $pmserver\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.

10.2.2.8 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 $pmserver\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.

10.2.2.9 To Configure Movement Types Domain Values

This section explains how to configure Movement Types domain values.

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 $pmserver\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.

10.2.2.10 How to Configure the Bill of Materials Explosion

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, then click Run, then type regedit, then 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_Adaptor.
     - For Oracle 11.5.10 Version, use SDE_ORA11510_Adaptor.
     - For Oracle R12 Version, use SDE_ORAR12_Adaptor.
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/pmserver.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_Adaptor.
   - For Oracle R12 Version, use SDE_ORAR12_Adaptor.
3. Re-start the PowerCenter server (Informatica Service).

10.2.2.11 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_TYP E_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 Informatica PowerCenter Designer, open the SDE_ORA<Ver>_Adaptor.
2. Open the mplt_SA_ORA_ProductTransactionFact mapplet.
3. Double-click the Expression transformation to open the Edit Transformations dialog, 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.

10.2.3 Configuration Steps for Oracle Supply Chain and Order Management Analytics For PeopleSoft

This section contains configuration steps required before you do a full data load that apply to PeopleSoft.

Not applicable to Oracle Business Intelligence Applications Release 7.9.5.

10.2.4 Configuration Steps for Oracle Supply Chain and Order Management Analytics For Universal

This section contains configuration steps required before you do a full data load that apply to Universal.

Not applicable to Oracle Business Intelligence Applications Release 7.9.5.

10.2.5 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Supply Chain and Order Management Analytics, and contains the following topics:

- Section 10.2.5.1, "Configuration Steps for Oracle Supply Chain and Order Management Analytics For All Source Systems"
- Section 10.2.5.2, "Configuration Steps for Oracle Supply Chain and Order Management Analytics For Oracle EBS"
- Section 10.2.5.3, "Configuration Steps for Oracle Supply Chain and Order Management Analytics For PeopleSoft"
- Section 10.2.5.4, "Configuration Steps for Oracle Supply Chain and Order Management Analytics For Universal"

10.2.5.1 Configuration Steps for Oracle Supply Chain and Order Management Analytics For All Source Systems

This section contains configuration steps that apply to all source systems, and contains the following topics:

- Section 10.2.5.1.1, "Tracking Multiple Attribute Changes in Bookings"
- Section 10.2.5.1.2, "Process of Aggregating Oracle Supply Chain and Order Management Analytics Tables"
- Section 10.2.5.1.3, "About Tracking Multiple Products for Oracle Supply Chain and Order Management Analytics"
- Section 10.2.5.1.4, "Adding Dates to the Order Cycle Time Table for Post-Load Processing"
- Section 10.2.5.1.5, "About Configuring the Backlog Period Date for Oracle Supply Chain and Order Management Analytics"
- Section 10.2.5.1.6, "Example of How Backlog Data Is Stored in the Backlog History Table"
- Section 10.2.5.1.7, "About Configuring the Customer Status History Fact for Post-Load Processing In Oracle Supply Chain and Order Management Analytics"
- Section 10.2.5.1.8, "Configuring the Customer Status History Fact table"
- Section 10.2.5.1.9, "How to Configure the Customer Status History Fact table"
- Section 10.2.5.1.10, "About Configuring the Inventory Monthly Balance Table"
- Section 10.2.5.1.11, "How to Configure the Inventory Monthly Balance"
- Section 10.2.5.1.12, "About Configuring the Product Transaction Aggregate Table"
- Section 10.2.5.1.13, "How to Configure the Product Transaction Aggregate Table"

10.2.5.1.1 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 Informatica PowerCenter Designer, open theSDE_ORA115<*ver*>_Adaptor or SDE_ORAR12_Adaptor folder.
2. Open the mplt_SA_ORA_SalesOrderLinesFact mapplet.
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.

10.2.5.1.2 Process of Aggregating Oracle Supply Chain and Order Management Analytics Tables

This section contains Oracle Supply Chain and Order Management 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.

- The 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.
- The 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.
- The 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.
- The 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 Sales Invoice Lines Aggregate Table

1. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Select the appropriate container for your OLTP source system.
3. Display the Tasks tab.
4. For each of the following tasks, display the Parameters tab and specify an appropriate value in the **Value** field:
   - SIL_SalesInvoiceLinesAggregate_Derive_PreLoadImage
   - SIL_SalesInvoiceLinesAggregate_Derive_PreSoftDeleteImage
   - PLP_SalesInvoiceLinesAggregate_Derive_PostLoadImage
   - PLP_SalesInvoiceLinesAggregate_Load

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. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
2. Select the appropriate container for your OLTP source system.
3. Display the Tasks tab.
4. For each of the following tasks, display the Parameters tab and specify an appropriate value in the **Value** field:
   - SIL_SalesOrderLinesAggregate_Derive_PreLoadImage
   - SIL_SalesOrderLinesAggregate_Derive_PreSoftDeleteImage

10.2.5.1.3 About Tracking Multiple Products for Oracle Supply Chain and Order Management 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 10–2. 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 10–2 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 10–2 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 10–3.

Table 10–3 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

10.2.5.1.4 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 Informatica 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.

10.2.5.1.5 About Configuring the Backlog Period Date for Oracle Supply Chain and Order Management 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.

10.2.5.1.6 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 10–4.

Table 10–4 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 10–5.

Table 10–5 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 10–6.

Table 10–6 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 10–7.

Table 10–7 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 10–7, 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 10–8 provides a list of all applicable mappings and their corresponding Expression transformations that you must modify.

Table 10–8 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).

10.2.5.1.7 About Configuring the Customer Status History Fact for Post-Load Processing In Oracle Supply Chain and Order Management Analytics

In Oracle Supply Chain and Order Management 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.

10.2.5.1.8 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 10.2.5.1.9, "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 targetting 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 10.2.5.1.9, "How to Configure the Customer Status History Fact table".

10.2.5.1.9 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 10.2.5.1.8, "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. In the DAC Client, display the Design view, and navigate to the appropriate container for your OLTP source system.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
2. Click on the Tasks tab and query for the following two tasks:
   - PLP_CustomerStatusHistoryFact_New_Customers_Load
   - PLP_CustomerStatusHistoryFact_Status_Revalidate
3. 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.

4. Save the changes.

10.2.5.1.10 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.

10.2.5.1.11 How to Configure the Inventory Monthly Balance

Before you run the initial ETL session or incremental ETL sessions to load the Inventory Monthly Balance table, configure the Inventory Monthly Balance as follows.

To configure the Inventory Monthly Balance

1. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Select the appropriate container for your OLTP source system.
3. Display the Tasks tab.
4. For each of the following tasks, display the Parameters tab and create the specified parameter name and parameter value
   - PLP_InventoryMonthlyBalance $$GRAIN 'MONTH'
   - PLP_InventoryDailyBalance_Trim $$KEEP_PERIOD 'MONTH'
   - PLP_InventoryDailyBalance_Trim $$NUM_OF_PERIOD 3

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.

10.2.5.1.12 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).

10.2.5.1.13 How to Configure the Product Transaction Aggregate Table

Before you run the initial ETL and then the incremental ETL to load the Product Transaction aggregate table, you need to configure the Product Transaction Aggregate Table, as follows.

To configure the Product Transaction Aggregate Table

1. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Select the appropriate container for your OLTP source system.
3. Display the Tasks tab.
4. For each of the following tasks, display the Parameters tab and create the specified parameter name and parameter value
   - PLP_ProductTransactionAggregate $$REFRESH_PERIOD 'MONTH'
   - PLP_ProductTransactionAggregate $$GRAIN 'MONTH'
   - PLP_ProductTransactionAggregate $$NUM_OF_PERIOD 3
   - PLP_ProductTransactionAggregate_Full $$GRAIN 'MONTH'

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.

10.2.5.2 Configuration Steps for Oracle Supply Chain and Order Management Analytics For Oracle EBS

This section contains configuration steps that apply to Oracle EBS, and contains the following topics:

- Section 10.2.5.2.1, "How to Configure Sales Order Lines Data Storage"
- Section 10.2.5.2.3, "How to Configure Sales Schedule Lines Data Storage"
- Section 10.2.5.2.3, "How to Configure Sales Schedule Lines Data Storage"
- Section 10.2.5.2.4, "About the Handling of Booked and Nonbooked Orders in the Sales Schedule Lines Table"
- Section 10.2.5.2.5, "About Loading Bookings at the Schedule Line Level"
- Section 10.2.5.2.6, "How to Configure Early and Late Tolerances for Shipments"
- Section 10.2.5.2.7, "How to Configure Sales Invoice Lines Data Storage"
- Section 10.2.5.2.8, "How to Configure the Sales Invoice Extract"
- Section 10.2.5.2.9, "How to Configure the Order Lines Execution Plan and OTB ETL"
- Section 10.2.5.2.10, "How to Configure Procurement and Spend Analytics for Oracle Supply Chain and Order Management Analytics"
- Section 10.2.5.2.11, "How to Configure Oracle Financial Analytics for Oracle Supply Chain and Order Management Analytics"
- Section 10.2.5.2.12, "About Tracking Attribute Changes in Bookings"
- Section 10.2.5.2.13, "About Viewing the Data Warehouse Changes by Salesperson ID"
- Section 10.2.5.2.14, "How to Configure Different Types of Backlog Calculations"
- Section 10.2.5.2.15, "Adding Closed Orders to Backlog Calculations"
- Section 10.2.5.2.16, "How to Configure the Bill of Materials Explosion Options"
- Section 10.2.5.2.17, "How to Configure the Left Bound and Right Bound Calculation Option"

10.2.5.2.1 How to Configure Sales Order Lines Data Storage

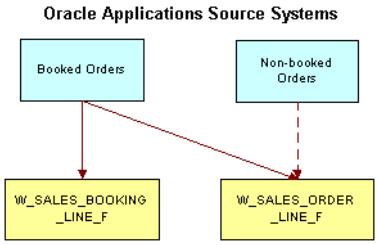
Sales order lines are the itemized lines that make up a sales order. This information is stored in the W_SALES_ORDER_LINE_F table. This topic describes how to modify the type of information stored in this table.

10.2.5.2.2 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 10–1. 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.

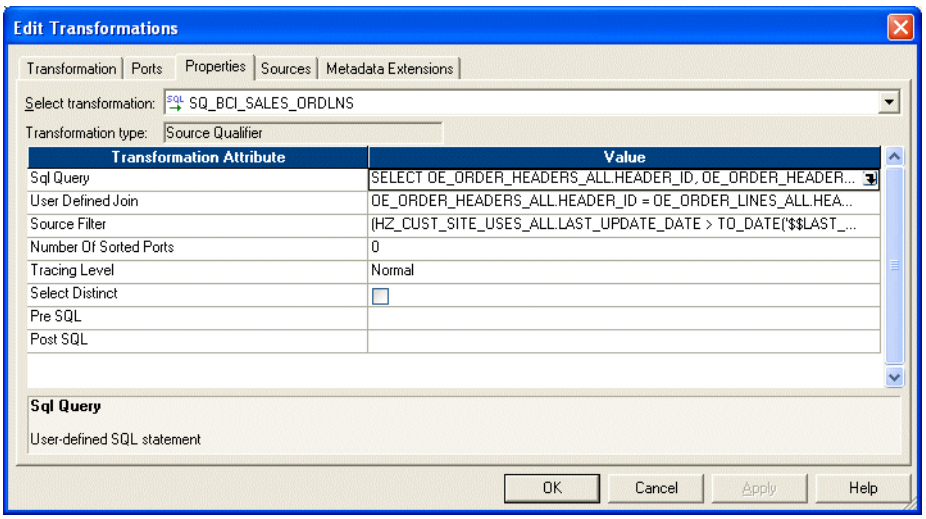
Figure 10–1 Handling Booked and Nonbooked Orders



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).

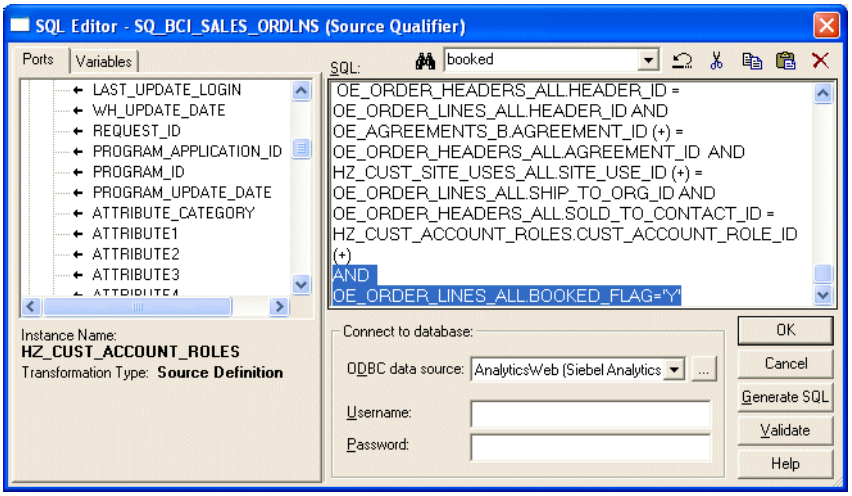
To include nonbooked orders in the Sales Order Lines tables

1. In Informatica PowerCenter Designer, open the SDE_ORA115<*ver*>_Adaptor or SDE_ORAR12_Adaptor folder.
2. Open the `mplt_BC_ORA_SalesOrderFact` mapplet in the Mapplet 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.

10.2.5.2.3 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.

10.2.5.2.4 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

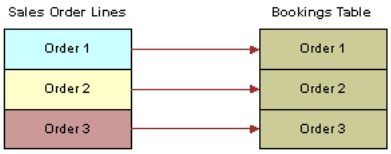
1. In Informatica PowerCenter Designer, open the SDE_ORA115<*ver*>_Adaptor or SDE_ORAR12_Adaptor folder.

2. Open the `mplt_BC_ORA_SalesScheduleLinesFact` mapplet in the Mapplet 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.

10.2.5.2.5 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 10–2 Sales Order Lines and Bookings Table



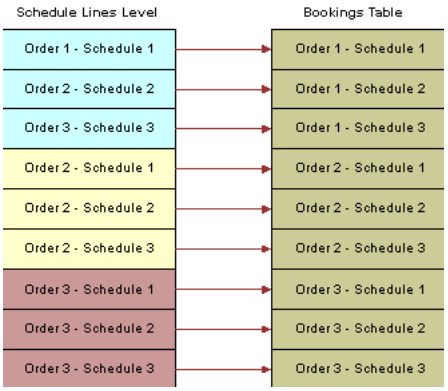
There are two subject areas in the SDE_ORA115<*ver*>_Adaptor or SDE_ORAR12_ Adaptor containers:

- Enterprise Sales - Booking Lines & Order Lines
- Enterprise Sales - Booking Lines & Schedule Lines

The execution plans that are installed with Oracle Business Intelligence 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 10–3 Bookings recorded at the Schedule Line level

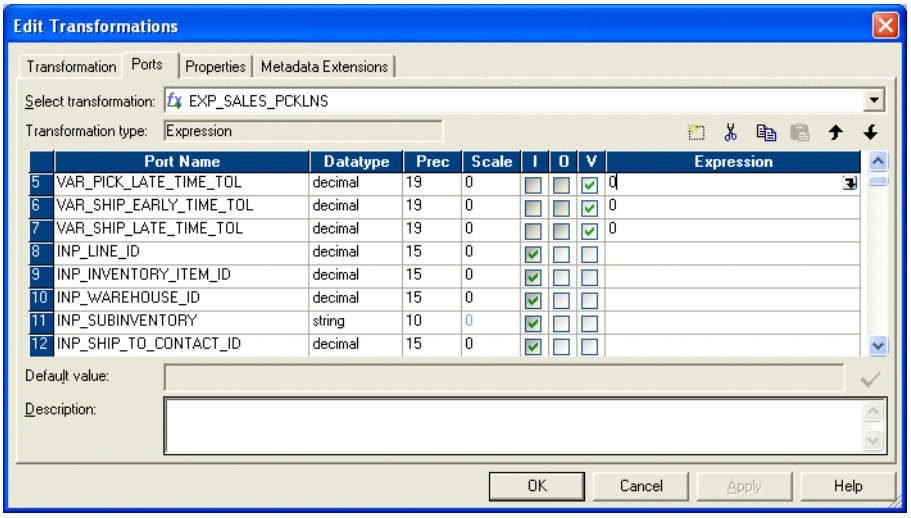


10.2.5.2.6 How to Configure Early and Late Tolerances for Shipments You configure the definition of early and late shipments by editing the `EXP_SALES_PCKLNS` expression in the `mplt_SA_ORA_SalesPickLinesFact` mapplet. The `mplt_SA_ORA_SalesPickLinesFact` mapplet is used by the `SDE_ORASalesPickLinesFact` 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 Informatica PowerCenter Designer, open the SDE_ORA115<*ver*>_Adaptor or SDE_ORAR12_Adaptor 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.

10.2.5.2.7 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.

10.2.5.2.8 How to Configure the Sales Invoice Extract By default, the Oracle Supply Chain and Order Management 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 Informatica PowerCenter Designer, open the SDE_ORA115<*ver*>_Adaptor or SDE_ORAR12_Adaptor 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 mplt_BC_ORA_SalesInvoiceLinesFact_Primary mapplet.

10.2.5.2.9 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 Informatica PowerCenter Designer, open the PLP folder.
2. Open the `PLP_SalesCycleLinesFact` mapplet in the Mapplet 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. In the DAC Client, in the Design tab, expand the Configuration Points node, and select the Sales PLP Optional Tasks.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
10. Activate the appropriate Subject Areas.
11. Open the Source Qualifier in SDE_ORA_SalesPickLinesFact.
12. 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`.

10.2.5.2.10 How to Configure Procurement and Spend Analytics for Oracle Supply Chain and Order Management Analytics The Oracle Supply Chain and Order Management Analytics application uses tables that are also used in Oracle's Procurement and Spend Analytics Family of Products.

For Oracle 11i and Oracle R12, you need to use the following configuration steps for Procurement and Spend Analytics to configure Oracle Supply Chain and Order Management Analytics:

- Section 7.2.2.1, "Configuring the Country Region and State Region Name"
- Section 7.2.2.2, "Configuring the State Name"
- Section 7.2.2.3, "Configuring the Country Name"
- Section 7.2.2.4, "Configuring the Configuring the Make-Buy Indicator"

10.2.5.2.11 How to Configure Oracle Financial Analytics for Oracle Supply Chain and Order Management Analytics

The Oracle Supply Chain and Order Management 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 Supply Chain and Order Management Analytics:

- Section 9.2.5.2.1, "How to Configure How GL Balances Is Populated in Oracle EBS"
- Section 9.2.5.2.2, "How to Configure Oracle Profitability Analytics Transaction Extracts"

10.2.5.2.12 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.

10.2.5.2.13 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 PowerCenter 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 10–9.

Table 10–9 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 10–9 is entered into the IA Bookings table (`W_SALES_BOOKING_LINE_F`) as shown in Table 10–10.

Table 10–10 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 10–11.

Table 10–11 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 10–12.

Table 10–12 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

10.2.5.2.14 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 Supply Chain and Order Management 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.

10.2.5.2.15 Adding Closed Orders to Backlog Calculations By default, the Oracle Supply Chain and Order Management 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 Informatica PowerCenter Designer, open the SDE_ORA115<*ver*>_Adaptor or SDE_ORAR12_Adaptor folder.
2. Open the `mplt_BC_ORA_SalesOrderLinesFact` mapplet in the Mapplet 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` mapplet in the Mapplet 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.

10.2.5.2.16 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 10–13 Variables for the bompexpl.exploder_userexit Stored Procedure

Input Variable	Preconfigured Value	Description
VERIFY_FLAG	0	A verify flag with a value of 1 only applies to standard BOM.
ORG_ID	ORGANIZATION_ID	Organization ID
ORDER_BY	1	Controls the order of the records. 1—Operation Sequence Number, Item Number 2—Item Number, Operation Sequence Number
GRP_ID	Negative Sequence ID -1, -2, and so on.	Unique value to identify current explosion.
SESSION_ID	Negative Sequence ID -1, -2, and so on.	Unique value to identify current session.
LEVELS_TO_EXPLODE	10	Levels to explode.
BOM_OR_ENG	1	1—BOM 2—ENG
IMPL_FLAG	1	1—Implemented Only 2—Implemented and Non-implemented
PLAN_FACTOR	2	1—Yes 2—No
EXPLODE_OPTION	2	1—All 2—Current 3—Current and Future

Table 10–13 (Cont.) Variables for the bompexpl.exploder_userexit Stored Procedure

Input Variable	Preconfigured Value	Description
MODULE	2	1—Costing 2—BOM 3—Order Entry 4—ATO 5—WSM
CST_TYPE_ID	0	Cost type ID for cost explosion.
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 Informatica PowerCenter Designer, open the SDE_ORA<Ver>_Adaptor.
2. Open the SDE_ORA_BomItemFact mapping.
3. Double-click the `EXP_BOMPEXPL_EXPLODER_USEREXIT` expression transformation to open the Edit Transformations dialog, 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, 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 Informatica PowerCenter Designer, open the SDE_ORA<Ver>_Adaptor.
2. Open the SDE_ORA_BOMItemFact mapping.
3. Double-click the EXP_BOMPEXPL_EXPLODER_USEREXIT expression transformation to open the Edit Transformations dialog, 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, 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 Informatica PowerCenter Designer, open the SDE_ORA<Ver>_Adaptor.
2. Open the mplt_BC_ORA_BOMHeaderDimension mapplet.
3. Double-click the SQL qualifier `SQ_BOM_INVENTORY_COMPONENTS` to open the Edit Transformations dialog, 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
   ```

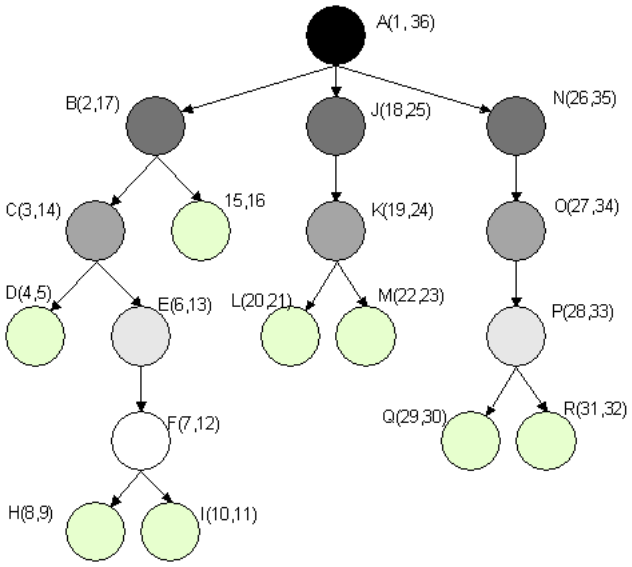
Note: You can also remove these two filters to extract all types of BOM.

5. Click Apply, and Validate the mapping and save your changes to the repository.

10.2.5.2.17 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 10–4 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 Informatica PowerCenter Designer, open the SDE_ORA<Ver>_Adaptor.
2. Open the SIL_BOMItemFact mapping.
3. Double-click the COMPUTE_BOUNDS stored procedure transformation to open the Edit Transformations dialog, 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.

10.2.5.3 Configuration Steps for Oracle Supply Chain and Order Management Analytics For PeopleSoft

This section contains configuration steps that apply to PeopleSoft.

Not applicable to Oracle Business Intelligence Applications Release 7.9.5.

10.2.5.4 Configuration Steps for Oracle Supply Chain and Order Management Analytics For Universal

This section contains configuration steps that apply to Universal.

Not applicable to Oracle Business Intelligence Applications Release 7.9.5.

11

Configuring Oracle Human Resources Analytics

This chapter describes how to configure Oracle Human Resources Analytics for particular sources to meet your business needs.

It contains the following topics:

- Section 11.1, "Overview of Oracle Human Resources Analytics"
- Section 11.2, "Configuration Required Before A Full Load for Oracle HR Analytics"

> **Note:** All references to Oracle 11i also apply to Oracle R12.

11.1 Overview of Oracle Human Resources Analytics

Oracle Human Resources 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.
- **Workforce Profile.** The Workforce 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.

11.2 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 11.2.1, "Configuration Steps for Oracle HR Analytics For All Source Systems"
- Section 11.2.2, "Configuration Steps for Oracle HR Analytics For Oracle EBS"
- Section 11.2.3, "Configuration Steps for Oracle HR Analytics For PeopleSoft"
- Section 11.2.4, "Configuration Steps for Oracle HR Analytics For Universal"
- Section 11.2.5, "Configuration Steps for Controlling Your Data Set"

11.2.1 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 7, "Configuring Common Areas and Dimensions").

11.2.2 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 11.2.2.1, "About Domain Values and CSV Worksheet Files for Oracle HR Analytics"
- Section 11.2.2.2, "How to Configure the domainValues_Employee_Ethnic_Group_Code_ora<ver>.csv"
- Section 11.2.2.3, "How to Configure the domainValues_Employee_Sex_MF_ora<ver>.csv"
- Section 11.2.2.4, "How to Configure the domainValues_Employee_Veteran_Status_Code_ora<ver>.csv"
- Section 11.2.2.5, "How to Configure the domainValues_Employment_Cat_ora<ver>.csv"
- Section 11.2.2.6, "How to Configure the domainValues_Employment_Exempt_Flg_ora<ver>.csv"

- Section 11.2.2.7, "How to Configure the domainValues_Employment_Full_Time_Flg_ora<ver>.csv"
- Section 11.2.2.8, "How to Configure the domainValues_Employment_Status_ora<ver>.csv"
- Section 11.2.2.9, "How to Configure the domainValues_EventTypes_ora<ver>.csv"
- Section 11.2.2.10, "How to Configure the domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv"
- Section 11.2.2.11, "How to Configure the domainValues_HRPosition_Exempt_Flg_ora<ver>.csv"
- Section 11.2.2.12, "How to Configure the domainValues_Job_Eeo_Cat_Code_ora<ver>.csv"
- Section 11.2.2.13, "How to Configure the domainValues_Job_Flsa_Stat_Code_ora<ver>.csv"
- Section 11.2.2.14, "How to Configure the domainValues_Pay_Type_Grp_Code_ora<ver>.csv"
- Section 11.2.2.15, "How to Configure the domainValues_Pay_Type_Flg_ora<ver>.csv"
- Section 11.2.2.16, "How to Configure Address Types for HR Profile"
- Section 11.2.2.17, "How to Configure Phone Types for HR Profile"
- Section 11.2.2.18, "How to Configure Education Degree Codes for Employee Dimension"
- Section 11.2.2.19, "About Configuring Key Flexfields"
- Section 11.2.2.20, "How to Configure the Key Flexfields for the Job Dimension"
- Section 11.2.2.21, "How to Configure the Key Flexfields for the HR Position Dimension"
- Section 11.2.2.22, "How to Configure the Key Flexfields for the Pay Grade Dimension"
- Section 11.2.2.23, "How to Configure multi-segmented Flexfields"
- Section 11.2.2.24, "How to Configure Flags for the Pay Type Dimension"
- Section 11.2.2.25, "How to Configure Classification Names for Payroll"

11.2.2.1 About Domain Values and CSV Worksheet Files for Oracle HR Analytics

You configure Oracle HR Analytics by mapping domain values to columns in the CSV files located in the `$pmserver\server\infa_shared\LkpFiles` folder. For example, `D:\Informatica\PowerCenter8.1.1\server\infa_shared\Lkpfiles`.

For more information on configuring domain values with CSV worksheet files, see Section 5.12, "About Domain Values" and Section 5.13, "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 11–1 lists the CSV worksheet files and the domain values for Oracle HR Analytics in the `$pmserver\server\infa_shared\LkpFiles` folder.

Table 11–1 Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_ Employee_Ethnic_ Group_Code_ ora<ver>.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_ ora<ver>.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_ ora<ver>.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_ ora<ver>.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_ Employment Dimension, SDE_ ORA_Employment Dimension_Full
domainValues_ Employment_ Exempt_Flg_ ora<ver>.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_ Employment Dimension, SDE_ ORA_Employment Dimension_Full
domainValues_ Employment_Full_ Time_Flg_ ora<ver>.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_ Employment Dimension, SDE_ ORA_Employment Dimension_Full
domainValues_ Employment_Status_ ora<ver>.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_ Employment Dimension, SDE_ ORA_Employment Dimension_Full

Table 11–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_ EventTypes_ ora<ver>.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_ora<ver>.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_ora<ver>.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_ ora<ver>.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_ ora<ver>.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_ ora<ver>.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 11–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_Pay_Type_Flg_ora<ver>.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_Sex_MF_ora<ver>.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_ora<ver>.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_ora<ver>.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_Employment Dimension, SDE_ORA_Employment Dimension_Full
domainValues_Employment_Exempt_Flg_ora<ver>.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_Employment Dimension, SDE_ORA_Employment Dimension_Full
domainValues_Employment_Full_Time_Flg_ora<ver>.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_Employment Dimension, SDE_ORA_Employment Dimension_Full
domainValues_Employment_Status_ora<ver>.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_Employment Dimension, SDE_ORA_Employment Dimension_Full

Table 11–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_ EventTypes_ ora<ver>.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_ora<ver>.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_ora<ver>.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_ ora<ver>.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_ ora<ver>.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_ ora<ver>.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 11–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_Pay_Type_Flg_ora<ver>.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

Note: When editing CSV files, make sure that you:

- Do no change the case of values in the CSV file.

  For example, do not change 'CONTRACTOR' to 'Contractor'.

- Do not add new values to the W_ columns, which are not already included in the CSV file.

  In other words, you can add new rows to the spreadsheet, but the W_ values must map to those in the out-of-the-box spreadsheet.

11.2.2.2 How to Configure the domainValues_Employee_Ethnic_Group_Code_ora<ver>.csv

This section explains how to configure the domainValues_Employee_Ethnic_Group_Code_ora<ver>.csv.

1. Identify the Ethnic Group Codes in your Oracle 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_ora<ver>.csv file using a text editor in the $pmserver\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.

11.2.2.3 How to Configure the domainValues_Employee_Sex_MF_ora<ver>.csv

This section explains how to configure the domainValues_Employee_Sex_MF_ora<ver>.csv.

1. Identify the Sex Codes in your Oracle 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_ora<ver>.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.

11.2.2.4 How to Configure the domainValues_Employee_Veteran_Status_Code_ora<ver>.csv

This section explains how to configure the domainValues_Employee_Veteran_Status_Code_ora<ver>.csv

1. Identify the Veteran Status Codes in your Oracle 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_ora<ver>.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.

11.2.2.5 How to Configure the domainValues_Employment_Cat_ora<ver>.csv

This section explains how to configure the domainValues_Employment_Cat_ora<ver>.csv.

1. Identify the User Person Types in your Oracle 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_ora<ver>.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 (results of the SQL query) to one of the delivered W_EMPLOYEE_CATEGORY_CODE domain values in the delivered example csv file.

   You can map more than one User Person Type to the same W_EMPLOYEE_CATEGORY_CODE by adding in new rows. For example:

```
Contractor          CONTRACTOR  CONTRACTOR
Contingent Worker   CONTRACTOR  CONTRACTOR
```

   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.

11.2.2.6 How to Configure the domainValues_Employment_Exempt_Flg_ora<ver>.csv

This section explains how to configure the domainValues_Employment_Exempt_Flg_ora<ver>.csv

1. Identify the FLSA Statuses in your Oracle source system by using the following SQL:

   ```
   SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS  ORDER BY 1
   ```

2. Open the domainValues_Employment_Exempt_Flg_ora<ver>.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.

11.2.2.7 How to Configure the domainValues_Employment_Full_Time_Flg_ora<ver>.csv

This section explains how to configure the domainValues_Employment_Full_Time_Flg_ora<ver>.csv.

1. Identify the Employment Categories in your Oracle 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_ora<ver>.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.

11.2.2.8 How to Configure the domainValues_Employment_Status_ora<ver>.csv

This section explains how to configure the domainValues_Employment_Status_ora<ver>.csv

1. Identify the Per System Statuses in your Oracle 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_ora<ver>.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.

11.2.2.9 How to Configure the domainValues_EventTypes_ora<ver>.csv

This section explains how to configure the domainValues_EventTypes_ora<ver>.csv

1. Identify the Event Types in your Oracle 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_ora<ver>.csv file using a text editor in the $pmserver\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.

11.2.2.10 How to Configure the domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv

This section explains how to configure the domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv.

1. Identify the Position Statuses in your Oracle 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_ora<ver>.csv file using a text editor in the $pmserver\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.

11.2.2.11 How to Configure the domainValues_HRPosition_Exempt_Flg_ora<ver>.csv

This section explains how to configure the domainValues_HRPosition_Exempt_Flg_ora<ver>.csv.

1. Identify the FLSA Statuses in your Oracle source system by using the following SQL:

```
SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS
ORDER BY 1
```

2. Open the domainValues_HRPosition_Exempt_Flg_ora<ver>.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.

11.2.2.12 How to Configure the domainValues_Job_Eeo_Cat_Code_ora<ver>.csv

This section explains how to configure the domainValues_Job_Eeo_Cat_Code_ ora<ver>.csv

1. Identify the EEO Job Categories in your Oracle 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_ora<ver>.csv file using a text editor in the $pmserver\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.

11.2.2.13 How to Configure the domainValues_Job_Flsa_Stat_Code_ora<ver>.csv

This section explains how to configure the domainValues_Job_Flsa_Stat_Code_ ora<ver>.csv.

1. Identify the FLSA Statuses in your Oracle 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_ora<ver>.csv file using a text editor in the $pmserver\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.

11.2.2.14 How to Configure the domainValues_Pay_Type_Grp_Code_ora<ver>.csv

This section explains how to configure the domainValues_Pay_Type_Grp_Code_ ora<ver>.csv.

1. Identify the Pay Elements in your Oracle 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_ora<ver>.csv file using a text editor in the $pmserver\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.

   Note: Do not change lines one to four in the domainValues_Pay_Type_Grp_Code_ora<ver>.csv file.

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.

11.2.2.15 How to Configure the domainValues_Pay_Type_Flg_ora<ver>.csv

This section explains how to configure the domainValues_Pay_Type_Flg_ora<ver>.csv.

1. Identify the Costing (Debit or Credit) in your Oracle 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_ora<ver>.csv file using a text editor in the $pmserver\lkpfiles 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.

11.2.2.16 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 Informatica PowerCenter Designer, open the SDE_ORA1158_Adaptor 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.

11.2.2.17 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 Informatica PowerCenter Designer, open the SDE_ORA1158_Adaptor folder.
2. In Mapplet Designer, open the mplt_SA_ORA_EmployeeDimension mapplet.
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.

11.2.2.18 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. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
2. Select a container from the drop down list to the right of the Execute button.
3. Display the Tasks tab, and select the SDE_ORA_EmployeeDimension task.
4. Specify or create the $$QUALIFICATION_CATEGORY_LIST parameter with an appropriate value.

   For example, $$QUALIFICATION_CATEGORY_LIST = 'DEGREE', 'DT'.
5. Save your changes.

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.

11.2.2.19 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 11.2.2.20, "How to Configure the Key Flexfields for the Job Dimension"
- Section 11.2.2.21, "How to Configure the Key Flexfields for the HR Position Dimension"
- Section 11.2.2.22, "How to Configure the Key Flexfields for the Pay Grade Dimension"
- Section 11.2.2.23, "How to Configure multi-segmented Flexfields"

11.2.2.20 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 the Key Flexfields for the Job Dimension

1. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
2. Select a container from the drop down list to the right of the Execute button.
3. Display the Tasks tab, and select the SDE_ORA_JobDimension task.
4. Specify or create the following parameters with an appropriate value.
   - $$JOBCODE_FLXFLD_SEGMENT_COL = the Column Name of Job Code.
   - $$JOBFAMILYCODE_FLXFLD_SEGMENT_COL = the Column name of Job Group Code.
5. In the Tasks tab, and select the SDE_ORA_CodeDimension_Job task.
6. Specify or create the following parameters with an appropriate value.
   - $$JOBCODE_FLXFLD_SEGMENT_COL = the Column Name of Job Code.
   - $$JOBNAME_FLXFLD_SEGMENT_COL = the Column Name of the Job Name.
7. Save your changes.

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.

11.2.2.21 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. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
2. Select a container from the drop down list to the right of the Execute button.
3. Display the Tasks tab, and select the SDE_ORA_HRPositionDimension task.
4. Specify or create the following parameter with an appropriate value.
   - $$POSITION_NUM_FLXFLD_SEGMENT_COL = the Column Name of Position Num.
5. In the Tasks tab, and select the SDE_ORA_HRPositionDimension_Full task.
6. Specify or create the following parameter with an appropriate value.
   - $$POSITION_NUM_FLXFLD_SEGMENT_COL = the Column Name of Position Num.
7. Save your changes.

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.

11.2.2.22 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. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
2. Select a container from the drop down list to the right of the Execute button.
3. Display the Tasks tab, and select the SDE_ORA_CodeDimension_Pay_Level task.
4. Specify or create the following parameter with an appropriate value.

- $$PAYLVLNAME_FLXFLD_SEGMENT_COL = the Column Name for the Pay level Name.

5. Save your changes.

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.

11.2.2.23 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 that 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 character (for example, a period '.'), 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 Business Intelligence 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 period (that is '.') 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 PowerCenter 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.

To obtain the OLTP table names:

1. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Select a container from the drop down list to the right of the Execute button.

3. Display the Tasks tab, and select the SDE_ORA_JobDimension and SDE_ORA_JobDimension tasks (using the example job name flexfield example).
4. Note the value of the following parameter (.
   - $$JOBNAME_FLXFLD_SEGMENT_COL

   In this case, the mapplet name that you need to derive is mplt_BC_ORA_Codes_Job.
5. Launch Informatica PowerCenter Designer and log on to the repository and navigate to the appropriate folder.
6. 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.

11.2.2.24 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 specified 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_ora<ver>.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.

11.2.2.25 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 11–2 Classification Name Parameters

Parameter Name	Description
$$GROSS_PAY_CLASSIFICATION_NAME	Add all the Classification Names that add up to the Total Gross Pay.
$$TOTAL_DEDUCTION_CLASSIFICATION_NAME	Add all the Classification Names that correspond to any deduction.
$$TOTAL_TAXES_CLASSIFICATION_NAME	Add all the Classification Names that correspond to various Taxes.
$$TOTAL_GROSS_FOR_NETPAY_EARNINGS	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 $$GROSS_PAY_CLASSIFICATION_NAME parameter.
$$TOTAL_GROSS_FOR_NETPAY_DEDUCTIONS	This is same as the parameter $$TOTAL_DEDUCTION_CLASSIFICATION_NAME. 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 Business Intelligence 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. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Select a container from the drop down list to the right of the Execute button.
3. Display the Tasks tab, and select the SDE_ORA_PayrollFact_Agg_Items.
4. Specify the following parameters as appropriate with an appropriate value.
   - $$CLASSIFICATION_NAMES_FOR_GROSS_PAY='PTO Accruals','Earnings','Imputed Earnings'
   - $$CLASSIFICATION_NAMES_FOR_TOTAL_DEDUCTION='Involuntary Deductions','Pre-Tax Deductions','Voluntary Deductions','Tax Deductions'
   - $$CLASSIFICATION_NAMES_FOR_TOTAL_TAXES='Tax Deductions'
   - $$CLASSIFICATION_NAMES_FOR_NET_PAY_EARNINGS='PTO Accruals','Earnings'
   - $$CLASSIFICATION_NAMES_FOR_NET_PAY_DEDUCTIONS='Involuntary Deductions','Pre-Tax Deductions','Voluntary Deductions','Tax Deductions'
5. Save your changes.

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 ()
```

11.2.3 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 11.2.3.1, "Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM"
- Section 11.2.3.2, "How to configure the domainValues_Employee_Ethnic_Group_Code_psft.csv"
- Section 11.2.3.3, "How to configure the domainValues_Employee_Sex_MF_psft.csv"
- Section 11.2.3.4, "How to configure the domainValues_Employee_Veteran_Status_Code_psft.csv"
- Section 11.2.3.5, "How to configure the domainValues_Employment_Category_psft.csv"
- Section 11.2.3.6, "How to configure the domainValues_Employment_Exempt_Flg_psft.csv"
- Section 11.2.3.7, "How to configure the domainValues_Employment_Full_Time_Flg_psft.csv"
- Section 11.2.3.8, "How to configure the domainValues_Employment_Status_psft.csv"
- Section 11.2.3.9, "How to configure the domainValues_Emp_EventType_Grp_Code_psft.csv"
- Section 11.2.3.10, "How to configure the domainValues_Emp_EventType_SubG_Code_psft.csv"
- Section 11.2.3.11, "How to configure the domainValues_HRPosition_Active_Pos_Flg_psft.csv"
- Section 11.2.3.12, "How to configure the domainValues_HRPosition_Exempt_Flg_psft.csv"
- Section 11.2.3.13, "How to configure the domainValues_Job_Eeo_Cat_Code_psft.csv"
- Section 11.2.3.14, "How to configure the domainValues_Job_Flsa_Stat_Code_psft.csv"
- Section 11.2.3.15, "How to configure the domainValues_Pay_Type_Grp_Code_psft.csv"
- Section 11.2.3.16, "How to configure the domainValues_Emp_EventType_EventReason_Code_psft.csv"
- Section 11.2.3.17, "How to configure the domainValues_Emp_EventType_EventClass_Code_psft.csv"

- Section 11.2.3.18, "How to configure the file_pension_comp_flg.csv"
- Section 11.2.3.19, "How to Configure Compensation Flag and Pension Compensation Flag for Pay Type Dimension"
- Section 11.2.3.20, "How to Configure the Performance Review Type ETL runtime parameter"
- Section 11.2.3.21, "How to configure Event Types using domainValues_EventTypes_psft.csv"

11.2.3.1 Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM

The table below lists the CSV worksheet files and the domain values for PeopleSoft HR Analytics in $pmserver\LkpFiles folder located on Informatica Server box.

Table 11–3 Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft 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 11–3 (Cont.) Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft 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 11–3 (Cont.) Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft 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 Debit/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	Lists the event type class codes.	SDE_PSFT_EventTypeDimension_Stage2
domainValues_Emp_EventType_EventReason_Code_psft.csv	NA	Lists the event type reason codes.	SDE_PSFT_EventTypeDimension_Stage2

11.2.3.2 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 $pmserver\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.

11.2.3.3 How to configure the domainValues_Employee_Sex_MF_psft.csv

1. Identify the Sex Codes in your PeopleSoft 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\lkpfiles 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.

11.2.3.4 How to configure the domainValues_Employee_Veteran_Status_Code_psft.csv

1. Identify the Veteran Status Codes in your PeopleSoft 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\lkpfiles 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.

11.2.3.5 How to configure the domainValues_Employment_Category_psft.csv

1. Identify the User Person Types in your PeopleSoft 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\lkpfiles 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.

11.2.3.6 How to configure the domainValues_Employment_Exempt_Flg_psft.csv

1. Identify the FLSA Statuses in your PeopleSoft 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.

11.2.3.7 How to configure the domainValues_Employment_Full_Time_Flg_psft.csv

1. Identify the Employment Categories in your PeopleSoft 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.

11.2.3.8 How to configure the domainValues_Employment_Status_psft.csv

1. Identify the Per System Statuses in your PeopleSoft 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 $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.

11.2.3.9 How to configure the domainValues_Emp_EventType_Grp_Code_psft.csv

1. Identify the Event Types Group Codes in your PeopleSoft 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 $pmserver\lkpfiles 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.

11.2.3.10 How to configure the domainValues_Emp_EventType_SubG_Code_psft.csv

1. Identify the Event Types Subgroup Codes in your PeopleSoft 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 $pmserver\lkpfiles 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.

11.2.3.11 How to configure the domainValues_HRPosition_Active_Pos_Flg_psft.csv

1. Identify the Position Statuses in your PeopleSoft 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 $pmserver\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.

11.2.3.12 How to configure the domainValues_HRPosition_Exempt_Flg_psft.csv

1. Identify the FLSA Statuses in your PeopleSoft 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 $pmserver\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.

11.2.3.13 How to configure the domainValues_Job_Eeo_Cat_Code_psft.csv

1. Identify the EEO Job Categories in your PeopleSoft 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 $pmserver\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.

11.2.3.14 How to configure the domainValues_Job_Flsa_Stat_Code_psft.csv

1. Identify the FLSA Statuses in your PeopleSoft 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 $pmserver\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.

11.2.3.15 How to configure the domainValues_Pay_Type_Grp_Code_psft.csv

The table below shows pay type sources and their associated group codes.

Table 11–4 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 $pmserver\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.

11.2.3.16 How to configure the domainValues_Emp_EventType_EventReason_Code_psft.csv

1. Identify the Costing (Debit or Credit) in your PeopleSoft 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 $pmserver\lkpfiles folder.
3. Copy EVENT_REASON the 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.

11.2.3.17 How to configure the domainValues_Emp_EventType_EventClass_Code_psft.csv

1. The shipped Domain Value for PeopleSoft 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 $pmserver\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.

11.2.3.18 How to configure the file_pension_comp_flg.csv

1. Identify the Earnings Codes in your PeopleSoft 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 $pmserver\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
```

11.2.3.19 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 $pmserver\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.

11.2.3.20 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 the DAC Repository.
2. Select 'Design' Mode and pick up 'PeopleSoft *<version>*' 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 source system:

```
SELECT EP_REVIEW_TYPE, EP_REVW_DESCR FROM PS_EP_REVW_TYP_TBL
```

11.2.3.21 How to configure Event Types using domainValues_EventTypes_psft.csv

To configure Event Types using the domainValues_EventTypes_psft.csv:

1. Identify the Actions and Action reasons in your PeopleSoft Source system using the following SQL:

```
SELECT A.ACTION,A.ACTION_REASON
FROM PS_ACTN_REASON_TBL A, (SELECT ACTION,ACTION_REASON, MAX(EFFDT) EFFDT
FROM PS_ACTN_REASON_TBL
GROUP BY ACTION,ACTION_REASON) B
WHERE A.ACTION=B.ACTION AND
A.ACTION_REASON = B.ACTION_REASON AND
A.EFFDT = B.EFFDT
```

2. Using a text editor, open the file domainValues_EventTypes_psft.csv in $pmserver\lkpfiles.
3. Copy the Action and Action Reason from the SQL result starting from the 6th line.

   Use commas to separate the entries.
4. Map each combination of Action and Action reason to one domain value for each of the domain value columns (that is, W_EVENT_GRP_CODE, W_EVENT_SUBG_CODE, W_EVENT_REASON_CODE and E_EVENT_CLASS).
5. Save and close the file.

11.2.4 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 Business Intelligence Applications Release 7.9.5.

11.2.5 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle HR Analytics.

- Section 11.2.5.1, "Configuration Steps for Oracle HR Analytics For All Source Systems"
- Section 11.2.5.2, "Configuration Steps for Oracle HR Analytics For Oracle EBS"
- Section 11.2.5.3, "Configuration Steps for Oracle HR Analytics For PeopleSoft"
- Section 11.2.5.4, "Configuration Steps for Oracle HR Analytics For Universal"

11.2.5.1 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 11.2.5.1.1, "How to Aggregate the Payroll Table for Oracle HR Analytics"
- Section 11.2.5.1.3, "How to control the rolling period for storing Employee Daily Snapshot data"
- Section 11.2.5.1.4, "How to control the granularity of the Employee Monthly Snapshot data"

11.2.5.1.1 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 Business Intelligence 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. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
2. Display the Tasks tab, and select the Task PLP_PayrollAggregate_Load.
3. Add a parameter under the parameters tab with the Name $$GRAIN.

4. Depending upon the aggregation requirement define the value as one among 'DAY', 'WEEK', 'MONTH', 'QUARTER' or 'YEAR' and Choose the value to be Static.
5. Save the Task.

11.2.5.1.2 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 the 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 11.2.5.1.3, "How to control the rolling period for storing Employee Daily Snapshot data"
- Section 11.2.5.1.4, "How to control the granularity of the Employee Monthly Snapshot data"

11.2.5.1.3 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 11.2.5.1.2, "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, display the Design view, 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'.

11.2.5.1.4 How to control the granularity of the Employee Monthly Snapshot data

For more information about granularity in the Employee Snapshot data, see Section 11.2.5.1.2, "About Configuring the Employee Snapshot Fact tables".

To configure the granularity of the Employee Snapshot data:

1. Log in to the DAC repository, display the Design view, 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.

11.2.5.2 Configuration Steps for Oracle HR Analytics For Oracle EBS

This section contains configuration steps that apply to Oracle EBS.

Not applicable to Oracle Business Intelligence Applications Version 7.9.5.

11.2.5.3 Configuration Steps for Oracle HR Analytics For PeopleSoft

This section contains configuration steps that apply to PeopleSoft, and contains the following topics:

- Section 11.2.5.3.1, "How to Tune Performance for the Payroll table for PeopleSoft HCM HR Analytics"

11.2.5.3.1 How to Tune Performance for the Payroll table for PeopleSoft 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 the 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.

11.2.5.4 Configuration Steps for Oracle HR Analytics For Universal

This section contains configuration steps that apply to Universal.

Not applicable to Oracle Business Intelligence Applications Release 7.9.5.

12

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 12.1, "Overview of Oracle CRM"
- Section 12.2, "Configuration Required Before A Full Load"
- Section 12.3, "Configuring Usage Accelerator"

12.1 Overview of Oracle CRM

Oracle Sales Analytics includes a wide range of tasks, analytics and engagement tactics that maximize the value of the customer relationship and contribute to sustainable revenue growth. CRM encompasses everything companies use to manage customer relationships, including capture and analysis of customer information and analytics to leverage that data toward better sales performance. Oracle Sales Analytics now includes Usage Accelerator for Siebel Sales Applications.

12.2 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 12.2.1, "How to Deploy Stored Procedures"
- Section 12.2.2, "Configuration Steps for Controlling Your Data Set"

12.2.1 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.

12.2.2 Configuration Steps for Controlling Your Data Set

This sections contains additional configuration steps for Oracle Sales Analytics, and contains the following topics:

- Section 12.2.2.1, "How to configure the snapshot frequency in Pipeline fact"

12.2.2.1 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 'GRAIN' you need to modify to configure for this.

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. In the DAC Client, display the Design view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Select the appropriate container for your OLTP source system.
3. Display the Tasks tab.
4. Locate the SIL_PipelineFact task, display the Parameters tab, and create a parameter called $$GRAIN with an appropriate parameter value.

12.3 Configuring Usage Accelerator

This section explains how to configure Usage Accelerator, and contains the following topics:

- Section 12.3.1, "What is Usage Accelerator?"
- Section 12.3.2, "About the Usage Accelerator Execution Plan"
- Section 12.3.3, "Using Usage Accelerator with Siebel 7.5.3 Applications"

12.3.1 What is Usage Accelerator?

Usage Accelerator enables 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, and drive alignment and increase the overall ROI of CRM implementations.

12.3.1.1 Configuring Usage Accelerator

This section provides an overview to configuring Usage Accelerator.

12.3.1.1.1 Tracking of Targeted Users In order to optimize the storage space required for the application data, Usage Accelerator enables you to choose which users you want to measure. 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.

12.3.1.1.2 Other Siebel Required Responsibilities Usage Accelerator introduces six new responsibilities. Three of the responsibilities will be used by all organizations except for Siebel Financial Services customers. Siebel Financial Services customers use three responsibilities that are targeted specifically for their user groups. The responsibilities determine which dashboards a user will see when they use Usage Accelerator. These responsibilities need to be assigned to users in the operational Siebel CRM application. The responsibilities are:

Table 12–1 User Accelerator Responsibilities

Usage Accelerator Responsibilities	Usage Accelerator Responsibilities (for Siebel Financial Services)
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

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 I.7, "ETL Issues".

The following table illustrates the dashboards and data-level security that are assigned to each Usage Accelerator responsibility. For more information about security in Oracle Business Intelligence Applications, see Chapter 6, "Integrated Security for Oracle Business Intelligence Applications".

Table 12–2 Dashboards and data-level security that are assigned to each Usage Accelerator responsibility.

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
Usage Accelerator - Sales Rep	Primary Position Data Level Security	Score Card	Individual Scorecard
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Account Coverage (Team)

Table 12–2 (Cont.) Dashboards and data-level security that are assigned to each Usage Accelerator responsibility.

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Contact Coverage (Team)
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Opportunity Coverage (Team)
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Financial Account Coverage (Team) - FINS Only
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Account Completeness (Team)
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Contact Completeness (Team)
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Opportunity Updates (Team)
Usage Accelerator - Sales Manager	No Position Based Security	Score Card	Team Scorecard
Usage Accelerator - Sales Manager	No Position Based Security	Score Card	Individual Scorecard
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Account Coverage (Team)
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Contact Coverage (Team)
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Opportunity Coverage (Team)
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Financial Account Coverage (Team) – FINS Only
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Account Completeness (Team)
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Contact Completeness (Team)
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Opportunity Updates (Team)
Usage Accelerator – FINS Sales Manager	No Position Based Security	Coverage	Account Coverage
Usage Accelerator – FINS Sales Manager	No Position Based Security	Coverage	Contact Coverage
Usage Accelerator – FINS Sales Manager	No Position Based Security	Coverage	Opportunity Coverage
Usage Accelerator – FINS Sales Manager	No Position Based Security	Coverage	Financial Account Coverage - FINS Only
Usage Accelerator – FINS Sales Manager	No Position Based Security	Completeness	Account Completeness

Table 12–2 (Cont.) Dashboards and data-level security that are assigned to each Usage Accelerator responsibility.

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
Usage Accelerator – FINS Sales Manager	No Position Based Security	Completeness	Contact Completeness
Usage Accelerator – FINS Sales Manager	No Position Based Security	Opportunity Updates	Opportunity Updates
Usage Accelerator – FINS Sales Manager	No Position Based Security	User Adoption	Active Users
Usage Accelerator – FINS Sales Manager	No Position Based Security	User Adoption	Application Usage - Excluded for FINS
Usage Accelerator – FINS Sales Manager	No Position Based Security	User Adoption	Application Usage - FINS Only
Usage Accelerator - Sales Executive	No Position Based Security	Scorecard	Team Scorecard
Usage Accelerator - Sales Executive	No Position Based Security	Scorecard	Individual Scorecard
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Account Coverage (Team)
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Contact Coverage (Team)
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Opportunity Coverage (Team)
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Financial Account Coverage (Team) – Fins Only
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Account Completeness (Team)
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Contact Completeness (Team)
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Opportunity Updates (Team)
Usage Accelerator - Sales Executive	No Position Based Security	Coverage	Account Coverage
Usage Accelerator - Sales Executive	No Position Based Security	Coverage	Contact Coverage
Usage Accelerator - Sales Executive	No Position Based Security	Coverage	Opportunity Coverage
Usage Accelerator - Sales Executive	No Position Based Security	Coverage	Financial Account Coverage – FINS Only
Usage Accelerator - Sales Executive	No Position Based Security	Completeness	Account Completeness
Usage Accelerator - Sales Executive	No Position Based Security	Completeness	Contact Completeness
Usage Accelerator - Sales Executive	No Position Based Security	Opportunity Updates	Opportunity Updates

Table 12–2 (Cont.) Dashboards and data-level security that are assigned to each Usage Accelerator responsibility.

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
Usage Accelerator - Sales Executive	No Position Based Security	User Adoption	Active Users
Usage Accelerator - Sales Executive	No Position Based Security	User Adoption	Application Usage - Excluded for FINS
Usage Accelerator - Sales Executive	No Position Based Security	User Adoption	Application Usage - FINS Only

12.3.1.2 About Deployment Date

Usage Accelerator module needs to be calibrated with a Deployment Date variable that determines the time that you want the Analysis to commence. This date will default to the date you create in the data warehouse, but you can change the date by changing the UA_DEPLOY_DT variable. Please note that if you set the Deployment Date earlier than the current calendar date, metrics that rely on the Last Updated Date (e.g. # of Updated Records, # of Logins, # of Sync's, etc.) of a record may be inaccurate. For this reason we recommend setting the Deployment Date to the current Calendar Date.

12.3.1.3 About Metric Definitions and Variables

This section explains Usage Accelerator metric definitions and variables.

12.3.1.3.1 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 12–3 User Adoption Metrics

Metric Category	Description	Variables
Active User	Identifies users that have not logged in or synced in the last two rolling 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

Table 12–3 (Cont.) User Adoption Metrics

Metric Category	Description	Variables
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

12.3.1.3.2 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 12–4 Opportunity Updates Metrics

Metric Category	Description	Variables
Opportunity Updates	Identifies Opportunities that have not been modified in the time period specified or have a close date that has passed.	Opportunity Updated Needed is currently set to 90 Days. Variable Name is: OPT_ OUTDATED_NUMDAYS

12.3.1.3.3 Completeness Metrics 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 12–5 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

12.3.1.3.4 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 12–6 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. 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.

12.3.2 About the Usage Accelerator Execution Plan

If you want to create a new execution plan or modify an existing execution plan to include a Usage Accelerator Subject area, then it should include a preceding task 'Create View For Usage Accelerator'. This is already done for the subject areas delivered out of the box. 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*'.

12.3.3 Using Usage Accelerator with Siebel 7.5.3 Applications

This section explains how to install and use Usage Accelerator with Siebel 7.5.3 Applications.

Note: This section only applies if you are planning to use Usage Accelerator with Siebel CRM 7.5.3 applications.

12.3.3.1 Usage Accelerator Application Integration with Siebel 7.5.3 CRM Application

The Usage Accelerator installation process consists of installing Usage Accelerator and importing seed data into the transactional database. You install Usage Accelerator using the standard Oracle Business Intelligence installer. The installation options that are available to you depend on the license key you enter.

To integrate Usage Accelerator

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 12–7 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 12–8 Import commands

Database Platform	Run Import Command
Oracle	$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
DB2UDB	$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
MSSQL	$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

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 Usage Accelerator reports.

12.3.3.2 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 12–9 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.

Table 12–9 (Cont.) Usage Accelerator Analytics Adapter Files

File	Description
seed_753_db2_UA.inp seed_753_mssql_UA.inp seed_753_ora_UA.inp	Import file used to control the import of the seed_753_UA.dat file to a particular database platform.

13

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 13.1, "Overview of Oracle Contact Center Telephony Analytics"
- Section 13.2.1, "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 7, "Configuring Common Areas and Dimensions").

13.1 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

13.2 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 13.1, "Overview of Oracle Contact Center Telephony Analytics"
- Section 13.2.1, "About the Dimension Key Resolution Process for Universal Source"
- Section 13.2.2, "How to Load the Dimension Table"

- Section 13.2.3, "How to Load the Fact Table"
- Section 13.2.4, "Setting Up The DATASOURCE_NUM_ID Parameter For The Universal Adapter"
- Section 13.2.5, "About Configuring the Event Type Columns"
- Section 13.2.7, "About Configuring Contact Status Type Column"
- Section 13.2.12, "How to Configure Dates and Times"
- Section 13.2.8, "Setting Up the Representative Activities Table"
- Section 13.2.9, "Setting Up the Contact Center Performance Table"
- Section 13.2.10, "Setting Up the Benchmarks and Targets Table"
- Section 13.2.13, "Configuring Flags for Oracle Contact Center Telephony Analytics"
- Section 13.2.14, "How to Exclude Representative Data for Post-Load Processing"

13.2.1 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.

13.2.2 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.

13.2.3 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.

13.2.4 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.

13.2.5 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.

13.2.5.1 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 REPACTIVITY 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.

13.2.5.2 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.

13.2.5.3 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.

13.2.5.4 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.

13.2.6 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.

13.2.7 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.

13.2.8 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 13–1 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.

13.2.9 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 \OracleBI\dwrep\Informatica\`SrcFiles` folder.
2. Type in your data to load the Contact Center Performance table.
3. Save and close the file.

13.2.10 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_bnchmrk_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 5, "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_bnchmrk_tgt.csv` file using a text editor in the \OracleBI\dwrep\Informatica\`SrcFiles` folder.
2. Type in your data to load the Benchmarks and Targets.
3. Save and close the file.

13.2.11 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 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 13–2 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 13–3 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 13–4 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 13–5 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

13.2.12 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 13–6 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.

13.2.13 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 13–7 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

13.2.14 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 Informatica 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.

13.2.15 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. In the DAC Client, in the Design view, select the Universal container

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Display the Subject Areas tab and query for the 'Service – Rep Activity' subject area.
3. With this subject area selected, in the detail pane, click on Configuration Tags.
4. Uncheck the configuration point 'ECC- Load into base fact and then aggregates'.
5. Check the configuration point 'ECC – Load directly into aggregate facts'.
6. Redesign the subject area 'Service – Rep Activity' and save.
7. 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 \OracleBI\dwrep\Informatica\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.

13.2.16 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. In the DAC Client, in the Design view, select the Universal container

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".

2. Display the Subject Areas tab, and query for the 'Service - Contact Center Performance' subject area
3. With this subject area selected, click on Configuration Tags

4. Uncheck the configuration point 'ECC- Load into base fact and then aggregates'
5. Check the configuration point 'ECC – Load directly into aggregate facts'
6. Redesign the subject area 'Service - Contact Center Performance and save.
7. 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 \OracleBI\dwrep\Informatica\SrcFiles folder.
2. Enter the data for Contact Center Performance at the hourly level.
3. Save and close the file.

> **Note:** The Configuration Tags 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.

13.2.17 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. In the DAC Client, in the Design view, select the Universal container.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client".
2. Display the Tasks tab, and locate the SDE_Universal task.
3. In the lower pane, display the Parameters tab.
4. 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 13–8 Expressions and Task Names

Table Name	INTEGRATION_ID Expression	Task Name
W_ACD_EVENT_F	CALLID_NUM||'~'||EVENT_SEQUENCE_NUM	SDE_Universal_ACDEventFact

Table 13–8 (Cont.) Expressions and Task Names

Table Name	INTEGRATION_ID Expression	Task Name
W_CNTCT_CNTR_PERF_F	CNTCT_NUM||'~'||CNTCT_REP_ID||'~'||CNTCT_START_DT	SDE_Universal_ContactCenterPerformanceFact
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_ContactCenterBenchmarkTargetFact
W_CNTCT_CNTR_PERF_REP_HOUR_A	CNTCT_CNTR_LOC_ID||'~'||CNTCT_REP_ID||'~'||CNTCT_DT	SDE_Universal_ContactCenterPerformanceRepHourAggregate
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

13.2.18 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-……..-MENUn where MENUm 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.

13.2.18.1 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 13–9 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

Table 13–9 (Cont.) Example call events in a navigation profile

CallId	Event Type	Event Sequence Number	Menu Id
20060104-C1	REP	4	
20060104-C1	IVR	5	MENU4
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 13–10

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.

13.2.18.2 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.

13.2.19 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' to opn the Column Properties dialog.
3. Go to Data format.
4. Check 'Override Default data format'.
5. Select Treat Number as 'Custom' and give Custom Numeric Format as "0#:##:##".

13.2.20 Configuration Steps for Controlling Your Data Set

Not applicable to Oracle Business Intelligence Applications Release 7.9.5.

14

Configuring the Oracle Business Intelligence Applications Repository

This chapter describes how to configure the Oracle BI Repository for the Oracle Business Analytics Warehouse.

It contains the following topics:

- Section 14.1, "Modifying the Oracle BI Repository for Siebel version 6.3 and version 7.5 applications"
- Section 14.2, "Configuring the Oracle BI Repository Connections"
- Section 14.3, "Setting up Date Specific Metrics"
- Section 14.4, "Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse"
- Section 14.5, "Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse"
- Section 14.7, "About Oracle BI Time Repository Variables"
- Section 14.8, "About User Authentication"
- Section 14.9, "About the Security or Visibility Configuration"
- Section 14.10, "About the Group Variable"
- Section 14.6, "About the Period Ago Keys for Oracle Business Analytics Warehouse"
- Section 14.11, "About Configuring Usage Tracking for Oracle Business Analytics Warehouse"
- Section 14.12, "About the Incremental Deployment of the Oracle BI Applications Repository"

> **Note:** When you access the EnterpriseBusinessAnalytics.rpd repository in Informatica tools, you must log on as the username and password Administrator\SADMIN.

14.1 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 Business Intelligence Applications Repository.

To modify the Oracle Business Intelligence 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

14.2 Configuring the Oracle BI Repository Connections

Oracle Business Intelligence 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 14.2.2, "How to Configure Oracle Business Intelligence Applications Repository Connections".
- For information about configuring repository variables, see Section 14.2.3, "How to Configure Oracle Business Intelligence Applications Repository Variables".

14.2.1 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 Business Intelligence 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.** Yo u need to configure the Oracle Business Analytics Warehouse Repository Initblocks Connection Pool to connect to the 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 14.2.1.1, "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.

14.2.1.1 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.

14.2.2 How to Configure Oracle Business Intelligence 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 Business Intelligence 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 EnterpriseBusinessAnalytics.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 Business Intelligence Applications Repository connection pools

1. Using the Oracle BI Administration Tool, open the EnterpriseBusinessAnalytics.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.

14.2.3 How to Configure Oracle Business Intelligence Applications Repository Variables

The section explains how to configure the Oracle BI repository variables.

To configure the Oracle Business Intelligence Applications Repository variables

1. Using the Oracle BI Administration Tool, open the EnterpriseBusinessAnalytics.rpd file in the $SAHome\OracleBI\Server\Repository folder.
2. On the Manage menu, click Variables.
3. In the Variables Manager dialog, 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 dialog.
5. On the Manage menu, click Security
6. In the Security Manager dialog, 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.

14.3 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 14–1 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 14–1 Invoice Fact Table Example

Date	Invoice Amount	Fulfill Amount	Payment Amount
January	4000	5000	4500

To implement date specific metrics

1. Using the Oracle BI Administration Tool, open the EnterpriseBusinessAnalytics.rpd.

   The EnterpriseBusinessAnalytics.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.

14.4 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. Using the Oracle BI Administration Tool, open the EnterpriseBusinessAnalytics.rpdEnterpriseBusinessAnalytics.rpd.

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 EnterpriseBusinessAnalytics.rpd file.

14.5 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. Using the Oracle BI Administration Tool, open the EnterpriseBusinessAnalytics.rpd.
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 EnterpriseBusinessAnalytics.rpd file.

14.6 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.

14.7 About Oracle BI Time Repository Variables

The Oracle BI repository is preconfigured with variables that are used for both reporting and internal usage.

Table 14–2 lists the Oracle BI repository date variables and their descriptions.

Table 14–2 Oracle BI Repository Date Variables

Variable Name	Description
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.

Table 14–2 (Cont.) Oracle BI Repository Date Variables

Variable Name	Description
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.
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.

14.8 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.

14.9 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 14–3 lists the groups in the Oracle Business Analytics Warehouse repository.

Table 14–3 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.
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 Procurement and Spend 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 Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics).

Table 14–3 (Cont.) Repository User Groups

Repository User Group	Description
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.
Purchasing Buyer	This user group is able to view Oracle's Procurement and Spend 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 Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to supplier performance.

Table 14–3 (Cont.) Repository User Groups

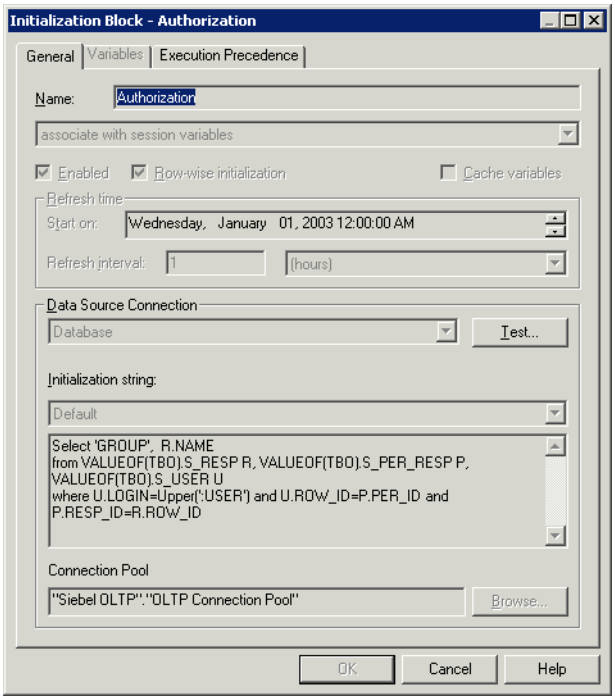
Repository User Group	Description
Supplier Performance Manager	This user group is able to view high-level content for Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to supplier performance.
Supply Chain Executive	This user group is able to view Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content.

14.10 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 EnterpriseBusinessAnalytics.rpd for the security filters to take effect.

If you are using the Oracle Business Intelligence 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 14–1 Screenshot of the Initialization Block - Authorization screen



14.11 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.

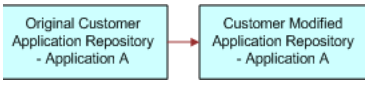
14.12 About the Incremental Deployment of the Oracle BI Applications Repository

Oracle Business Analytics Warehouse consist of various families of Oracle Business Intelligence 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.

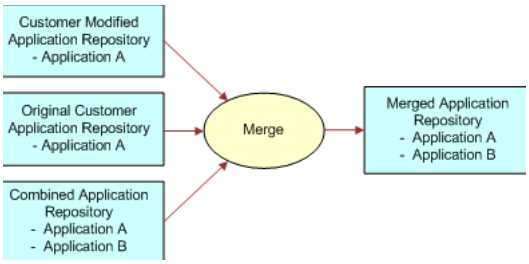
Figure 14–2 Oracle Business Analytics Warehouse environment



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 14–3 Merging with an Oracle Business Intelligence 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 B, "Configuring Universal Adapter for CRM Analytics"
- Appendix C, "Localizing Oracle Business Intelligence Deployments"
- Appendix D, "Integrating Interactive Dashboards and Operational Applications Data"
- Appendix E, "Configuring Metadata for Analytics Applications"
- Appendix F, "Oracle Business Analytics Warehouse for Life Sciences Data Considerations"
- Appendix G, "Pharma Disconnected Analytics Administration - Supplemental Information"
- Appendix H, "Using Oracle Business Analytics Warehouse Exception Reports"
- Appendix I, "Troubleshooting Oracle Business Intelligence Applications"

Note: For a high level road map for installation, configuration, and customization steps for Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications".

A

Supporting Tasks for DAC and Informatica PowerCenter

This section contains additional tasks relating to the DAC Client and DAC Server, and Informatica PowerCenter Client Tools, and contains the following topics:

- Section A.1, "How to Log Into the DAC Client"
- Section A.2, "How to Start and Stop the DAC Server"
- Section A.3, "How to Log Into Informatica PowerCenter Administration Console"
- Section A.4, "How to Start and Stop Informatica Services"
- Section A.5, "How to Stop and Start the Informatica Repository Service (Optional)"
- Section A.6, "About the DAC Server Scripts for Bash and C Shells"

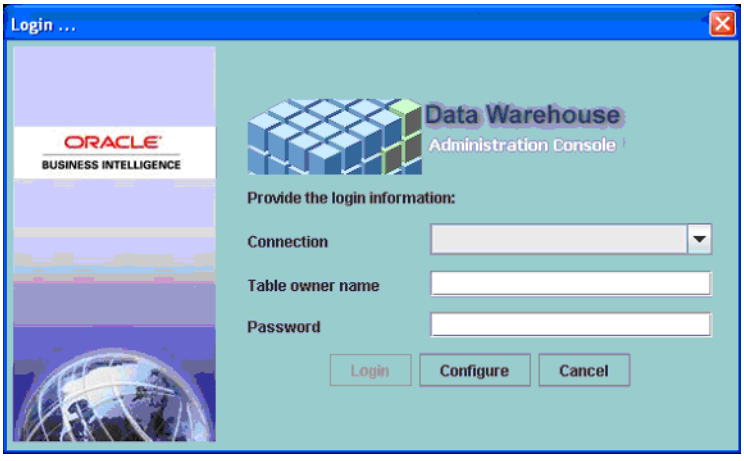
A.1 How to Log Into the DAC Client

You log into the DAC Client 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 4.8.3, "How to Create a DAC Connection".

To log into the DAC Client

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, select a connection for the required DAC Repository from the **Connection** drop down list.
3. In the Table owner name field, enter the database user name for the DAC Repository database.
4. In the Password field, enter the database password for the DAC Repository database.
5. Click Login to launch the DAC.

   The DAC client launches and connects to the DAC Repository.

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

A.2 How to Start and Stop the DAC Server

This section explains how to start and stop the DAC Server.

To start or stop the DAC Server on Windows

1. If you installed the DAC Server with the Oracle Business Intelligence Applications installation, choose the Windows Start menu, then Programs, then Oracle Business Intelligence, then Oracle DAC, and then Start DAC Server or Stop DAC Server.
2. If you installed the DAC Server by copying the \DAC\ directory to a machine, run the \DAC\startserver.bat script or \DAC\stopserver.bat script.

To start the DAC Server on UNIX and Linux

1. In csh-related shells, issue the following command:

   ```
   ./startserver.csh
   ```
   On AIX, use startserver_aix.csh.
2. In bash-related shells, issue the following command:

   ```
   ./startserver.sh
   ```
   On AIX, use startserver_aix.sh.

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. **Note**: To stop DAC Server running as a background process, use stopserver.sh or stopserver.csh.

To stop the DAC Server on UNIX and Linux

1. In csh-related shells, issue the following command:

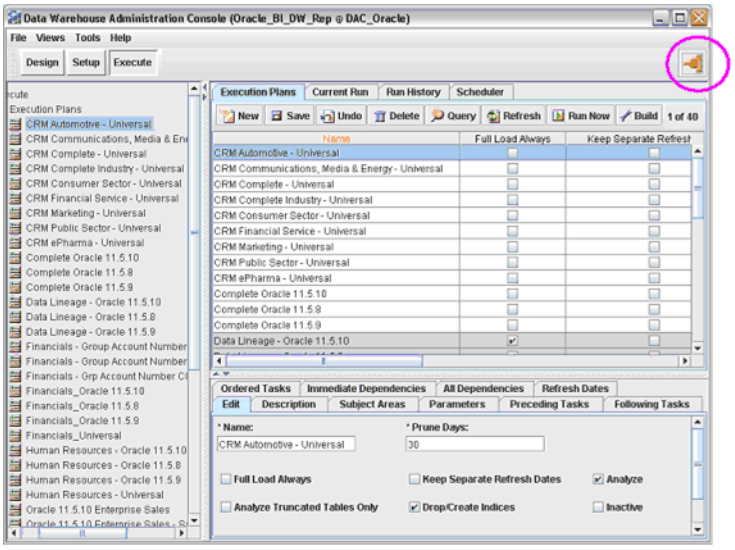
   ```
   ./stopserver.csh
   ```
2. In bash-related shells, issue the following command:

   ```
   ./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.

Tip: When you start the DAC Server, look at 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.



A.3 How to Log Into Informatica PowerCenter Administration Console

Informatica PowerCenter Administration Console is installed on the machine that hosts the gateway node for the PowerCenter domain. For a single machine install of PowerCenter Services as described in this chapter, PowerCenter Administration Console is installed along with PowerCenter Services.

To log into PowerCenter Administration Console

1. In Microsoft Internet Explorer or Mozilla Firefox, access the following URL:

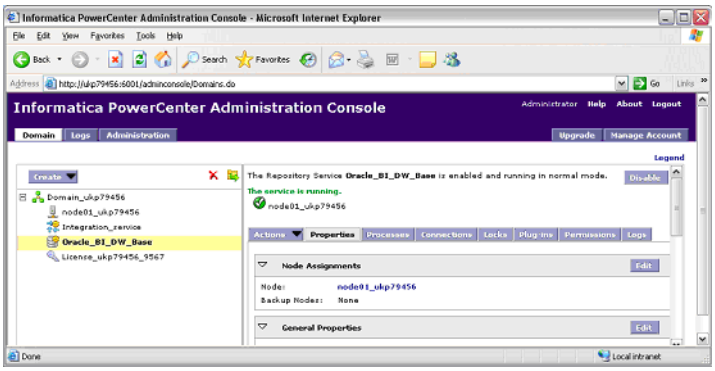
   http://<*gateway host*>:<*domain port*>

   Where:

   - <*gateway host*> is the name of the machine on which the gateway node has been configured; for a single-machine installation of PowerCenter Services it is the name of the machine on which PowerCenter Services has been installed.
   - <*domain port*> is the port number for the gateway. The default port number is 6001.

2. In the login page, enter the domain username and password that was specified when you created the domain during installation of PowerCenter Services.

   For more information about logging into PowerCenter Administration Console, refer to the Appendix in the *Informatica PowerCenter Installation and Configuration Guide*.



A.4 How to Start and Stop 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.

On Windows

1. Start and stop the Informatica Services from the Start Menu > Programs > Informatica PowerCenter 8.1.1 > Services > Start Informatica Services or Stop Informatica Services.

   You can also start and stop Informatica Services from the Windows Services panel.

On UNIX or Linux

1. Use the infaservice command to start or stop Informatica Services, as follows:

   ```
   Infaservice [startup | shutdown]
   ```

   The infaservice command is located in $Informatica/PowerCenter8.1.1/server/tomcat/bin.

The Informatica Services may also be stopped on a node by shutting down the node using PowerCenter Administration Console.

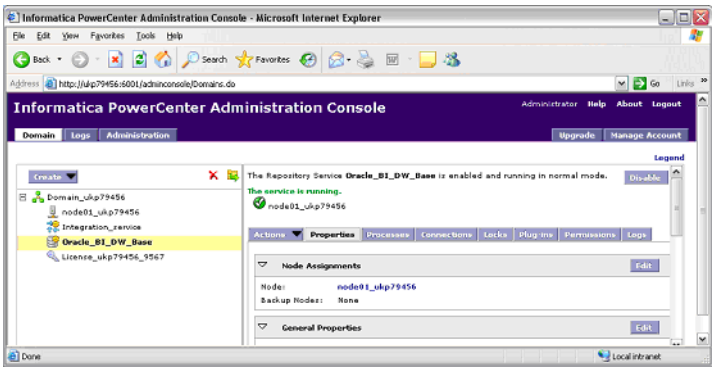
For more information about stopping and starting Informatica PowerCenter Services, refer to the Appendix in the *Informatica PowerCenter Installation and Configuration Guide*.

A.5 How to Stop and Start the Informatica Repository Service (Optional)

This section explains how to stop and start the Informatica Repository Service. For example, you might need to re-start the service after making changes to the service properties.

To stop and start the Informatica repository

1. In Informatica PowerCenter Administration Console, in the Domain tab, select the Repository Service (for example, Oracle_BI_DW_Base).



2. Do one of the following:
   - To stop the Informatica Repository Service, click Disable.
   - To start the Informatica Repository Service, click Enable.

A.6 About the DAC Server Scripts for Bash and C Shells

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 A–1 lists the available shell scripts and their usage. These files contain comments that provide information about how to configure the scripts.

Table A–1 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 for the DAC Server.
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 or Linux, 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: The files listed in Table A–1 need to have read, write, and execute permissions. If the files do not have these permissions, modify them using the chmod command.

When you copy across these files, use a MS-DOS to UNIX conversion tool, convert the script files 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.

On some shells, export commands in dac_env.sh and config.sh might not work correctly. In this case, try breaking the commands in two. For example, change the command from:

```
export JAVA_HOME=/opt/java1.5
```

To:

```
JAVA_HOME=/opt/java1.5
export JAVA_HOME
```

Follow these procedures to start and stop 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.

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
   ```

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.

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.

A.7 How to Configure the DAC Metadata Repository

This section explains how to configure the DAC metadata repository, and includes the following topics:

- Section A.7.1, "How to Set Up DAC System Properties"
- Section A.7.2, "How to Activate Join Indices for Teradata Databases"

A.7.1 How to Set Up DAC System Properties

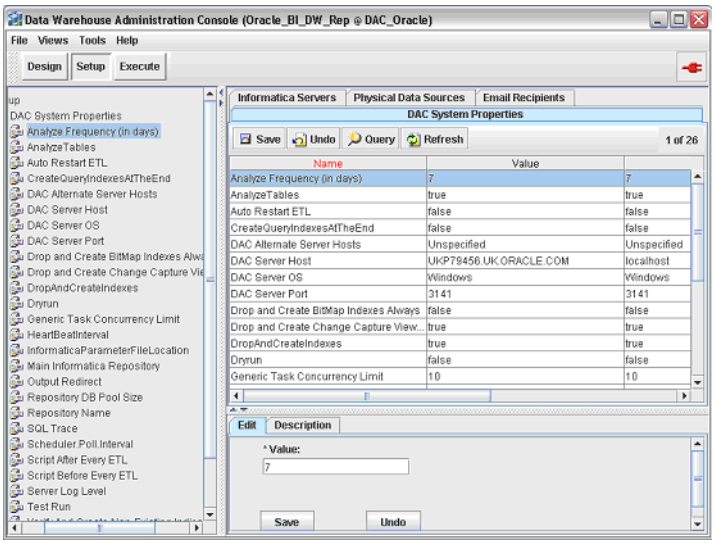
Follow this procedure to set up the DAC system properties, which determine the behavior of the DAC server. If you do not change a property value, Oracle Business Intelligence Applications uses the property value that is automatically set during the Oracle Business Intelligence Applications installation.

To set up the DAC system properties

1. In the DAC Client, display the Setup view.

   For more information about logging into the DAC, see Section A.1, "How to Log Into the DAC Client").

2. Display the DAC System Properties tab.



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).

A.7.2 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.

B

Configuring Universal Adapter for CRM Analytics

This appendix provides instructions for Configuring Universal Adapter for CRM Analytics, and contains the following topics:

- Section B.1, "What is Universal Adapter for CRM Analytics?"
- Section B.2, "Data Format Requirements"
- Section B.3, "About The Sample Source Files Provided"
- Section B.4, "How to Create Data Files for Universal Adapter for CRM Analytics"
- Section B.5, "List of Sample Source Files for Universal Adapter for CRM Analytics"

B.1 What is Universal Adapter for CRM Analytics?

Universal Adapter for CRM Analytics enables you to extract data from CRM transactional applications for analysis, when pre-packaged business adapters for these sources are not available. Unlike the pre-packaged business adapters that can only be used for particular source systems, Universal Adapter for CRM Analytics can be used for any source systems as long as the source data can be presented in the specified flat file format.

The CRM source data could also reside in custom applications or legacy systems. Universal Adapter for CRM Analytics is shipped as part of any of the Oracle Business Intelligence Applications CRM Analytics product and is not separately priced.

B.2 Data Format Requirements

To use Universal Adapter for CRM Analytics, source data should be presented in flat files according to the following specifications:

- Data should be in comma delimited format (*.csv files).
- All date values should be in the format of YYYYMMDDHH24MISS. For example, 20071231140300 should be used for December 31, 2007, 2:03 pm.
- If a string data value contains one or more commas, then it should be double quoted.
- Data should start from line six of each file. The first five lines of each file will be skipped during ETL process.

B.3 About The Sample Source Files Provided

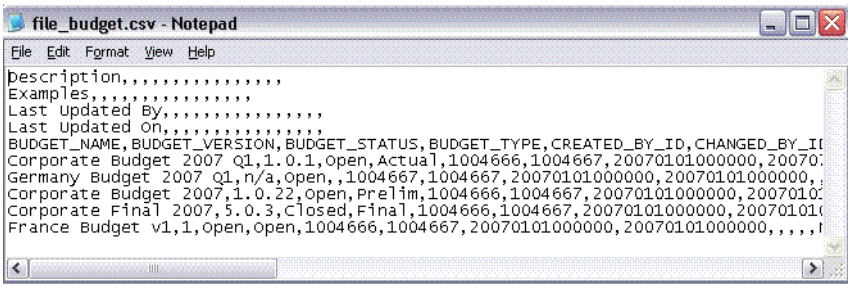
Oracle Business Intelligence Applications provides a sample source file for each supported source file type. Typically, each sample source files contain sample data rows. Sample source files are located in:

```
<OracleBI>\dwrep\Informatica\Sample Universal Source Files
```

For a list of sample source files, see Section B.5, "List of Sample Source Files for Universal Adapter for CRM Analytics".

The sample source file file_budget.csv is shown in the figure below:

Figure B–1 Sample Universal Adapter Source File



B.4 How to Create Data Files for Universal Adapter for CRM Analytics

To create data files, Oracle recommends that you use the sample source files provided (for a list of sample source files, see Section B.5, "List of Sample Source Files for Universal Adapter for CRM Analytics").

To create data files for Universal Adapter for CRM Analytics:

1. Copy the sample source files (file_xxx.csv) from the `<OracleBI>\dwrep\Informatica\Sample Universal Source Files` folder to the `<Informatica PowerCenter>\Server\SrcFiles` folder.
2. Delete the sample data rows from every sample source file.

   You must have all of the empty files in the <Informatica PowerCenter>\Server\SrcFiles folder first for ETL execution plans to run without interruptions. Some ETL execution plans might contain tasks that you do not need but that would fail and cause interruption in your ETL run if source files expected by the tasks cannot be found in the `<Informatica PowerCenter>\Server\SrcFiles` folder.
3. Populate the files that you need in the `<Informatica PowerCenter>\Server\SrcFiles` folder, making sure that the data starts at line six.

   For example, to load budget data, you might populate the file_budget.csv file.
4. In the DAC Client, create an execution plan to load the data from these files.

B.5 List of Sample Source Files for Universal Adapter for CRM Analytics

This section lists sample source files for Universal Adapter for CRM Analytics, and contains the following topics:

- Section B.5.1, "List of Source Files for Seed Data and Common Dimensions"

- Section B.5.2, "List of Source Files for Specific Subject Areas"

B.5.1 List of Source Files for Seed Data and Common Dimensions

This section lists source files for seed data and common dimensions. These source files are mandatory and should be populated when using the CRM Universal Adapter.

Table B–1 Seed Data and Common Dimension Source Files

CRM Subject Area	Associated Source Files
Seed Data	■ file_codes_common ■ file_costlst_g ■ file_dual_g ■ file_exch_rate ■ file_lov_d ■ file_lst_of_val_g
Common Dimension	■ file_employee ■ file_geo_ds ■ file_int_org ■ file_int_org_dh ■ file_org ■ file_person ■ file_position_ds ■ file_product ■ file_user

B.5.2 List of Source Files for Specific Subject Areas

This section lists CRM subject areas and their related source files. Populate these source files selectively, depending on your business needs.

Table B–2 Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Forecasting	■ file_dtl_fcst_fs ■ file_fcst_ser_ds ■ file_industry_ds ■ file_opty_ds ■ file_source ■ file_sum_fcst_fs ■ file_terr

Table B–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Marketing	■ file_camp_hist_fs ■ file_camp_opty_fs ■ file_geo_region_ds ■ file_household_ds ■ file_household_fs ■ file_industry_ds ■ file_ld_wave_ds ■ file_mktg_cost_fs ■ file_mktg_goal_fs ■ file_offer_ds ■ file_ofr_prod_fs ■ file_opty_ds ■ file_period_ds ■ file_person_fs ■ file_regn_ds ■ file_resp_prod_fs ■ file_response_ds ■ file_response_fs ■ file_segment_ds ■ file_source ■ file_terr ■ file_vendor_ds ■ file_wave_ds

Table B–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Sales	■ file_agree_ds ■ file_entlmnt_ds ■ file_geo_region_ds ■ file_industry_ds ■ file_invoice_fs ■ file_offer_ds ■ file_oiprdattr_ds ■ file_opty_cmpt_fs ■ file_opty_con_fs ■ file_opty_ds ■ file_opty_stg_fs ■ file_order_ds ■ file_orderitem_fs ■ file_person_fs ■ file_prdattrnm_ds ■ file_qiprdattr_ds ■ file_quote_ds ■ file_quoteitem_fs ■ file_regn_ds ■ file_revn_fs ■ file_segment_ds ■ file_source ■ file_srvreq_ds ■ file_sstage_ds ■ file_terr
CRM - Service	■ file_activity_fs ■ file_agree_ds ■ file_agreeitem_fs ■ file_assess_ds ■ file_asset_ds ■ file_asset_fs ■ file_claimelmnt_ds ■ file_entlmnt_ds ■ file_opty_ds ■ file_quote_ds ■ file_region_ds ■ file_source ■ file_srvreq_ds ■ file_srvreq_fs ■ file_survey_fs

Table B–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Automotive Vehicle Sales	■ file_asset_ds ■ file_household_ds ■ file_vehicle_fs
CRM - Comm, Media and Energy - Quotes and Orders	■ file_agree_ds ■ file_entlmnt_ds ■ file_industry_ds ■ file_offer_ds ■ file_oiprdattr_ds ■ file_opty_ds ■ file_order_ds ■ file_orderitem_fs ■ file_prilstitem_ds ■ file_qiprdattr_ds ■ file_quote_ds ■ file_quoteitem_fs ■ file_regn_ds ■ file_segment_ds ■ file_source ■ file_srvreq_ds ■ file_sstage_ds ■ file_terr
CRM - Consumer Sector Fund	■ file_fndhdr_fs ■ file_fndtxn_fs ■ file_fund_ds ■ file_fund_fs ■ file_plan_promo_ds ■ file_terr
CRM - Consumer Sector - Performance	■ file_performance_fs
CRM - Consumer Sector - Promotion	■ file_plan_promo_ds ■ file_promotion_fs
CRM - Consumer Sector - Retail Audit	■ file_invloc_ds ■ file_plan_promo_ds ■ file_rtlaudit_fs

Table B–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - ePharma - Account Call	■ file_accnt_rnk_ds ■ file_affl_aa_hs ■ file_affl_ac_hs ■ file_affl_ca_hs ■ file_affl_cc_hs ■ file_area_ds ■ file_call_accnt_fs ■ file_call_attnd_fs ■ file_con_rnk_ds ■ file_postn_con_ds
CRM - ePharma - Contact Call	■ file_affl_ac_hs ■ file_affl_cc_hs ■ file_area_ds ■ file_call_con_fs ■ file_con_rnk_ds ■ file_postn_con_ds
CRM - ePharma - Direct Sales	■ file_accnt_rnk_ds ■ file_affl_aa_hs ■ file_affl_ca_hs ■ file_alignmt_dhs ■ file_area_ds ■ file_ins_plan_ds ■ file_syndd_ds_fs
CRM - ePharma - Indirect Sales	■ file_accnt_rnk_ds ■ file_affl_aa_hs ■ file_affl_ca_hs ■ file_alignmt_dhs ■ file_area_ds ■ file_ins_plan_ds ■ file_syndd_ids_fs
CRM - ePharma - Medical Education	■ file_area_ds ■ file_con_rnk_ds ■ file_med_att_tmp ■ file_med_cst_tmp ■ file_med_ed_fs ■ file_med_evt_ds ■ file_med_plan_ds ■ file_postn_con_ds

Table B–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - ePharma - Objective	■ file_accnt_rnk_ds ■ file_affl_aa_hs ■ file_affl_ac_hs ■ file_affl_ca_hs ■ file_affl_cc_hs ■ file_alignmt_dhs ■ file_area_ds ■ file_call_accnt_fs ■ file_call_con_fs ■ file_con_rnk_ds ■ file_ins_plan_ds ■ file_objective_fs ■ file_plan_promo_ds ■ file_postn_con_ds ■ file_syndd_ds_fs ■ file_syndd_ids_fs ■ file_syndd_rx_fs
CRM - ePharma - Physician Plan Prescription	■ file_affl_aa_hs ■ file_affl_ac_hs ■ file_affl_ca_hs ■ file_affl_cc_hs ■ file_alignmt_dhs ■ file_area_ds ■ file_con_rnk_ds ■ file_postn_con_ds ■ file_syndd_xpt_fs
CRM - ePharma - Physician Prescription	■ file_affl_ac_hs ■ file_affl_cc_hs ■ file_alignmt_dhs ■ file_area_ds ■ file_con_rnk_ds ■ file_ins_plan_ds ■ file_postn_con_ds ■ file_syndd_rx_fs
CRM - ePharma - Plan Prescription	■ file_affl_aa_hs ■ file_affl_ca_hs ■ file_alignmt_dhs ■ file_area_ds ■ file_syndd_pt_fs

Table B–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - ePharma - Weekly Physician Prescription	■ file_affl_ac_hs ■ file_affl_cc_hs ■ file_alignmt_dhs ■ file_area_ds ■ file_con_rnk_ds ■ file_ins_plan_ds ■ file_postn_con_ds ■ file_syndd_w_fs
CRM - ePharma - Profile Rank	■ file_accnt_rnk_ds ■ file_affl_aa_hs ■ file_affl_ac_hs ■ file_affl_ca_hs ■ file_affl_cc_hs ■ file_area_ds ■ file_con_rnk_ds ■ file_per_rank_fs ■ file_postn_con_ds
CRM - Financial Service - Accounts and Application	■ file_agree_ds ■ file_asset_ds ■ file_asset_fs ■ file_balhstry_fs ■ file_fn_holding_ds ■ file_fn_holding_fs ■ file_industry_ds ■ file_ins_claim_ds ■ file_offer_ds ■ file_opty_ds ■ file_region_ds ■ file_regn_ds ■ file_revn_fs ■ file_segment_ds ■ file_source ■ file_src_offr_hs ■ file_srvreq_ds ■ file_srvreq_fs ■ file_sstage_ds ■ file_terr

Table B–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Financial Service - Accounts and Insurances	▪ file_asset_ds ▪ file_asset_fs ▪ file_claimelmnt_ds ▪ file_industry_ds ▪ file_ins_claim_ds ▪ file_ins_claim_fs ▪ file_terr
CRM - Financial Service - Balance History	▪ file_asset_ds ▪ file_balhstry_fs ▪ file_industry_ds
CRM - Public Sector - Benefit	▪ file_case_ds ▪ file_order_ds ▪ file_orderitem_fs
CRM - Public Sector - Case	▪ file_activity_fs ▪ file_case_ds ▪ file_case_fs ▪ file_incident_ds ▪ file_lead_ds ▪ file_period_ds ▪ file_region_ds ▪ file_srvreq_ds ▪ file_srvreq_fs
CRM - Public Sector - Incident	▪ file_case_ds ▪ file_incident_ds ▪ file_incident_fs ▪ file_period_ds
CRM - Public Sector - Lead	▪ file_case_ds ▪ file_lead_ds ▪ file_lead_fs ▪ file_period_ds

C

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 C.1, "Process of Maintaining Translation Tables for Oracle BI"
- Section C.2, "About Translating Presentation Services Strings"
- Section C.3, "Changing the Default Currency in Analytics Applications"

C.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 C.1.1, "Upgrading Oracle Business Intelligence Seed Data for Non-English Locales"
- Section C.1.2, "Externalizing Customer Metadata Strings"
- Section C.1.3, "Adding Custom Translations to the W_LOCALIZED_STRING_G Table"

C.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).

C.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.

C.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 C–1.

Table C–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. Using the Oracle BI Administration Tool in offline mode, open EnterpriseBusinessAnalytics.rpd.
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 dialog 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.

C.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).

C.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 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.

C.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.

C.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.

D

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 D.1, "Importing Oracle's Siebel Industry Applications Seed Data"
- Section D.2, "Completing the Initialization in the Siebel Operational Application"

D.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.

D.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 D.2.1, "Updating the Siebel Operational Application"
  - Section D.2.2, "Reapplying Customized Style Sheets"
  - Section D.2.3, "How to Configure Oracle Business Intelligence with Oracle's Siebel Web Extension (SWE)"
    - Section D.2.4, "Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services"
    - Section D.2.5, "Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE"
  - Section D.2.6, "How to Configure Oracle BI Action Links"
    - Section D.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 D.2.8, "Process of Integrating Analytics and Oracle's Siebel Operational Applications"
    - Section D.2.9, "Accessing Optional Analytics Applications"
    - Section D.2.10, "Configuring the Browser Settings for Operational Applications"
  - Section D.2.11, "Process of Customizing Oracle BI Application Home Page and Dashboards"
    - Section D.2.12, "Customizing Oracle BI Content on the Siebel Operational Application Home Page"
    - Section D.2.13, "Determining the Oracle BI Report Path Argument"
    - Section D.2.14, "Adding Views for Custom Oracle BI Interactive Dashboards"
    - Section D.2.15, "Verifying NQHOST Setup for Pharma Disconnected Analytics Client"
    - Section D.2.16, "Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client"
    - Section D.2.17, "Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client"

D.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.

D.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 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.

D.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 Business Intelligence 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 Business Intelligence 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 D.2.4, "Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services"
- Section D.2.5, "Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE"

D.2.4 Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services

This task is part of the Section D.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.

D.2.5 Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE

This task is part of the Section D.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 D.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.

D.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 D.2.7, "Testing Action Links"

D.2.7 Testing Action Links

This task is part of the Section D.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.

D.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 D.2.9, "Accessing Optional Analytics Applications"
- Section D.2.10, "Configuring the Browser Settings for Operational Applications"
- Section D.2.11, "Process of Customizing Oracle BI Application Home Page and Dashboards"

D.2.9 Accessing Optional Analytics Applications

This task is part of Section D.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 D–1 shows the additional options for Oracle Business Intelligence.

Table D–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.

D.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 D.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.

D.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 D.2.12, "Customizing Oracle BI Content on the Siebel Operational Application Home Page"
- Section D.2.13, "Determining the Oracle BI Report Path Argument"
- Section D.2.14, "Adding Views for Custom Oracle BI Interactive Dashboards"
- Section D.2.15, "Verifying NQHOST Setup for Pharma Disconnected Analytics Client"
- Section D.2.16, "Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client"
- Section D.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 Business Intelligence Applications, see Appendix G, "Pharma Disconnected Analytics Administration - Supplemental Information".

D.2.12 Customizing Oracle BI Content on the Siebel Operational Application Home Page

This task is part of the Section D.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 D.2.15, "Verifying NQHOST Setup for Pharma Disconnected Analytics Client".

The figure below shows example Inline and IFrame symbolic URL arguments.

Figure D–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.

4. 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

5. 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 D.2.13, "Determining the Oracle BI Report Path Argument" to determine the path to the Analytics report.

D.2.13 Determining the Oracle BI Report Path Argument

This task is part of the Section D.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

1. Log on to your Oracle BI Presentation Services as an Administrator.
2. In the Siebel operational application, navigate to Answers, then Oracle BI Presentation Services Administration.
3. Select Manage Analytics Catalog, and then navigate to your report.
4. 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	/shared/Sales/Pipeline/Overview/Top 10 Deals	Y	2
Syndicate	Constant	Siebel	Y	3
nQUser	Command	UseSiebelLoginId	Y	4
nQPassword	Command	UseSiebelLoginPassword	Y	5
PostRequest	Command	PostRequest	Y	6

D.2.14 Adding Views for Custom Oracle BI Interactive Dashboards

This task is part of the Section D.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.

D.2.15 Verifying NQHOST Setup for Pharma Disconnected Analytics Client

This task is part of the Section D.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 Delivers for SiebelDelivers	Y	1
nQUser	Command	UseSiebelLoginId	Y	2
nQPassword	Command	UseSiebelLoginPassword	Y	3

D.2.16 Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client

This task is part of the Section D.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.

D.2.17 Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client

This task is part of the Section D.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.

D.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.

D.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.

D.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

E

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 E.1, "What Are Oracle's Siebel Operational Applications?"
- Section E.2, "Updating Financial Analytics Logical Table Sources"
- Section E.3, "Externalizing Financial Services Metadata Translation Strings"
- Section E.4, "Disabling the Financial Analytics Logical Table Sources"
- Section E.5, "Developing and Deploying Predictive Scores"
- Section E.6, "Analytics Metadata Requirements for Oracle's Siebel Industry Applications"

E.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.

E.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. Using the Oracle BI Administration Tool, open the Analytics Repository (EnterpriseBusinessAnalytics.rpd).
2. Go to the Business Model and Mapping dialog (the logical layer dialog) 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 dialog 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 dialog 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.

E.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. Using the Oracle BI Administration Tool, open the EnterpriseBusinessAnalytics.rpd.
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 dialog.
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.

E.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. Using the Oracle BI Administration Tool, open the Analytics Repository (EnterpriseBusinessAnalytics.rpd).
2. Go to Business Model and Mapping dialog (the logical layer dialog) 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 dialog 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 dialog.

   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.

E.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.

E.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 E.6.1, "Oracle's Siebel Consumer Sector Dashboards and Pages"
- Section E.6.2, "Oracle's Siebel Consumer Sector Data Requirements"
- Section E.6.3, "Oracle's CME family of products (Oracle Communications, Media and Energy Sales Analytics"
- Section E.6.4, "Oracle's Siebel CME Dashboards and Pages"
- Section E.6.5, "Oracle's Siebel CME Data Requirements"
- Section E.6.6, "Oracle Pharma Sales Analytics Dimensions"
- Section E.6.7, "Dimensions Specific to Subject Areas in Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics"

E.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 E–1.

Table E–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 E–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?

E.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 E–2. This includes the fund, promotion, performance and retail audit schema.

Table E–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	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. 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.
Retail Audit	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. 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.

E.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.

E.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 E–3. These responsibilities also determine what subject areas the user may access.

Table E–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.

E.6.5 Oracle's Siebel CME Data Requirements

The data requirements for each of the Communications, Media and Energy dashboards are detailed in Table E–4.

Table E–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 Section E.5, "Developing and Deploying Predictive Scores" 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.

E.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*.

E.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.

E.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 E–5.

Table E–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.

E.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 E–6.

Table E–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.

E.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 E–7.

Table E–7 Product Costs For Customer Products

Product Type	Field to be Populated
Sample	Sample Cost
Detail	Avg. Promo Cost

Table E–7 (Cont.) Product Costs For Customer Products

Product Type	Field to be Populated
Promotional Item Cost	Sample Cost

E.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*.

E.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.

E.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 E–8 lists the categories that need to have the Usage Type field populated in the Catalog Admin Category Detail list.

Table E–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

E.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.

E.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.

E.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.

F

Oracle Business Analytics Warehouse for Life Sciences Data Considerations

This chapter contains the following topics:

- Section F.1, "What's New for Pharma Analytics in Version 7.9"
- Section F.2, "Oracle Business Analytics Warehouse for Life Sciences Overview"
- Section F.3, "Importing Syndicated Data into Oracle Business Analytics Warehouse"
- Section F.4, "Syndicated Data Flat File Formats"
- Section F.5, "Life Sciences Data Loading Issues with Oracle Business Analytics Warehouse"
- Section F.6, "Incremental Updates in the Oracle Business Analytics Warehouse LS Dimension Tables"

F.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 the 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 the DAC's Design area.

F.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.

F.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 F.3.1, "Syndicated Loading Definitions"
- Section F.3.2, "Data Types Supported in the Oracle Business Analytics Warehouse"
- Section F.3.3, "Loading Syndicated Data into the Oracle Business Analytics Warehouse"

F.3.1 Syndicated Loading Definitions

Table F–1 provides terms and definitions related to syndicated loading.

Table F–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 x 24 bottles = 24 units).

F.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 F–2.

Table F–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 F–2 (Cont.) Data Types Supported in Oracle Business Analytics Warehouse

Data Type	Description
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	**Incentives.** 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	**Modified.** 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.

F.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 F.3.3.1, "Loading From a Flat File Source Using Syndicated Data Staging Tables and Cross-Referencing Tables"
- Section F.3.3.2, "Loading From a Flat File Source Using Syndicated Market Staging Tables"

F.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 F–4 to Table F–19 in section Section F.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 F–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 F–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.

F.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.

F.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 F–4 to Table F–19 in section Section F.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.

F.4 Syndicated Data Flat File Formats

This section includes descriptions of the following flat files used in the syndicated data load process:

- Section F.4.1, "Syndicated Data Flat File for Direct Sales Data"
- Section F.4.2, "Syndicated Data Flat File for Indirect Sales Data"
- Section F.4.3, "Syndicated Data Flat File for Prescription Data"
- Section F.4.4, "Syndicated Data Flat File for Plan Level Rx Data"
- Section F.4.5, "Syndicated Data Flat File for Physician Plan Level Rx Data"
- Section F.4.6, "Syndicated Data Flat File for Weekly Early View Data"
- Section F.4.7, "Syndicated Market Flat File for Indirect Sales Data"
- Section F.4.8, "Syndicated Market Flat File for Prescription Data"
- Section F.4.9, "Syndicated Market Flat File for Plan Level Rx Data"
- Section F.4.10, "Syndicated Market Flat File for Physician Plan Level Rx Data"
- Section F.4.11, "Syndicated Market Flat File for Weekly Early View Data"
- Section F.4.12, "Flat File Cross-Reference Data Population Rules"

F.4.1 Syndicated Data Flat File for Direct Sales Data

Table F–4 provides information about the syndicated data flat file for Direct Sales data.

Table F–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. **Note:** 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. **Note:** 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 F–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. **Note:** 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. **Note:** 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)		

F.4.2 Syndicated Data Flat File for Indirect Sales Data

Table F–5 provides information about the syndicated data flat file for Indirect Sales data.

Table F–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 F–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. **Note:** 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. **Note:** 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. **Note:** 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. **Note:** 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. **Note:** 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 F–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)		

F.4.3 Syndicated Data Flat File for Prescription Data

Table F–6 provides information about the syndicated data flat file for Prescription data.

Table F–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. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CONTACT_ID	VARCHAR2(30)		External contact ID. **Note:** 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. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.

Table F–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. **Note:** 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. **Note:** 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)		

F.4.4 Syndicated Data Flat File for Plan Level Rx Data

Table F–7 provides information about the syndicated data flat file for Plan Level Rx data.

Table F–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. **Note:** 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. **Note:** 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. **Note:** 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 F–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)		

F.4.5 Syndicated Data Flat File for Physician Plan Level Rx Data

Table F–8 provides information about the syndicated data flat file for Physician Plan Level Rx data.

Table F–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. **Note:** 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. **Note:** 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. **Note:** Corresponding Siebel S_CONTACT.ROW_ID should be provided in cross-reference file.

Table F–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. **Note:** 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. **Note:** 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)		

F.4.6 Syndicated Data Flat File for Weekly Early View Data

Table F–9 provides information about the syndicated data flat file for Weekly Early View data.

Table F–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. **Note:** 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. **Note:** 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. **Note:** 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. **Note:** 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 F–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)		

F.4.7 Syndicated Market Flat File for Indirect Sales Data

Table F–10 provides information about the syndicated market flat file for Indirect Sales data.

Table F–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. **Note:** 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. **Note:** 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 F–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. **Note:** Corresponding Siebel S_ REGION.ROW_ID should be provided in cross-reference file.
MARKET_ID	VARCHAR2(30)		External market ID for the product. **Note:** 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. **Note:** 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)		

F.4.8 Syndicated Market Flat File for Prescription Data

Table F–11 provides information about the syndicated market flat file for Prescription data.

Table F–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. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CONTACT_ID	VARCHAR2(30)		External contact ID. **Note:** 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. **Note:** 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. **Note:** 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. **Note:** Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.

Table F–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)		

F.4.9 Syndicated Market Flat File for Plan Level Rx Data

Table F–12 provides information about the syndicated market flat file for Plan Level Rx data.

Table F–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. **Note:** 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. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.

Table F–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. **Note:** 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 Rxof 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)		

F.4.10 Syndicated Market Flat File for Physician Plan Level Rx Data

Table F–13 provides information about the syndicated market flat file for Physician Plan Level Rx data.

Table F–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 F–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. **Note:** 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. **Note:** 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. **Note:** 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. **Note:** 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 F–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. **Note:** 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)		

F.4.11 Syndicated Market Flat File for Weekly Early View Data

Table F–14 provides information about the syndicated market flat file for Weekly Early View data.

Table F–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. **Note:** Corresponding Siebel S_CONTACT.ROW_ID should be provided in cross-reference file.

Table F–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. **Note:** 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. **Note:** 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)		

F.4.12 Flat File Cross-Reference Data Population Rules

The following tables provide information about flat file cross-reference data population rules:

- Table F–15, " Flat File Cross-Reference Data Population Rules for Area (Brick) Data"

- Table F–16, " Flat File Cross-Reference Data Population Rules for Account and Plan Data"
- Table F–17, " Flat File Cross-Reference Data Population Rules for Payer Type Data"
- Table F–18, " Flat File Cross-Reference Data Population Rules for Contact Data"
- Table F–19, " Flat File Cross-Reference Data Population Rules for Product and Market Data"

Table F–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. **Note:** 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. **Note:** The same value should already be loaded in Siebel S_REGION.NAME.
EXT_PROVINCE	VARCHAR2(50)		Area province. **Note:** The same value should already be loaded in Siebel S_REGION.PROVINCE.
EXT_STATE	VARCHAR2(50)		Area state. **Note:** 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. **Note:** 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. **Note:** This value should come from Siebel S_REGION.ROW_ID.

Table F–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. **Note:** The same value should be used in the DATASOURCE_NUM_ID field of the syndicated data and syndicated market flat files.

Table F–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. **Note:** 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. **Note:** 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. **Note:** This value should come from Siebel S_REGION.ROW_ID.

Table F–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. **Note:** 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. **Note:** 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. **Note:** 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. **Note:** The same value should already be loaded in Siebel S_INS_PLAN.NAME.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel payer ID. **Note:** This value should come from Siebel S_REGION.ROW_ID.

Table F–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. **Note:** 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. **Note:** 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. **Note:** 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. **Note:** The same value should already be loaded in Siebel S_CONTACT.CSN.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel contact ID. **Note:** This value should come from Siebel S_CONTACT.ROW_ID.

Table F–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. **Note:** 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. **Note:** 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. **Note:** The same value should already be loaded in Siebel S_PROD_INT.
EXT_PROD_NAME	VARCHAR2(50)		Product and market name. **Note:** The same value should already be loaded in Siebel PROD_INT.NAME.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel product ID. **Note:** This value should come from Siebel S_PROD_INT.ROW_ID.

F.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 Client, 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'.

F.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.

F.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.

F.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.

F.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 though the setting of the Alignment Version flag in the DAC console. In Pharma Analytics 7.9 this flag is now set in the Source System Parameters screen within the 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.

G

Pharma Disconnected Analytics Administration - Supplemental Information

This appendix describes additional configuration steps for Oracle's Pharma Disconnected Analytics, and contains the following sections:

- Section G.1, "About Pharma Disconnected Analytics"
- Section G.2, "Sourcing Reports for Oracle's Pharma Disconnected Analytics"

G.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*.

G.2 Sourcing Reports for Oracle's Pharma Disconnected Analytics

Table G–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 G–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 G–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.

H

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 H.1, "Understanding Oracle Business Analytics Warehouse Exceptions"
- Section H.2, "Executing Oracle Business Analytics Warehouse Exception Reports"

H.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.

H.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 H–1.

Table H–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

H.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 H–2.

Table H–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.

H.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.

H.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.

H.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.

H.1.6 Circular Hierarchy Exceptions

Circular Hierarchies arise when the parent-child relationship has circular references. See Table H–3 for an example.

Table H–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 H–4 for an example.

Table H–4 Circular Hierarchy Example 2

Child	Parent
A1	A2

Table H–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.

H.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 Client, 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 H.2.1, "Cleansing Data".

H.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.)

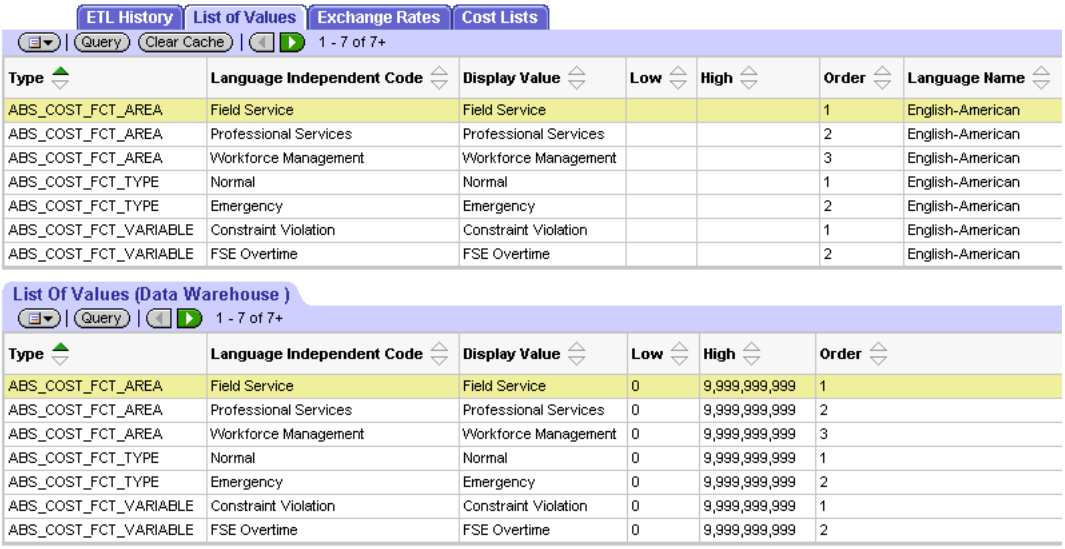
4. Repair the problem, using the text in the ETL Exception Explanation form as a guide.

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.

H.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 H–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
ABS_COST_FCT_AREA	Field Service	Field Service	0	9,999,999,999	1
ABS_COST_FCT_AREA	Professional Services	Professional Services	0	9,999,999,999	2
ABS_COST_FCT_AREA	Workforce Management	Workforce Management	0	9,999,999,999	3
ABS_COST_FCT_TYPE	Normal	Normal	0	9,999,999,999	1
ABS_COST_FCT_TYPE	Emergency	Emergency	0	9,999,999,999	2
ABS_COST_FCT_VARIABLE	Constraint Violation	Constraint Violation	0	9,999,999,999	1
ABS_COST_FCT_VARIABLE	FSE Overtime	FSE Overtime	0	9,999,999,999	2

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 4.11, "How to Install and Configure the DAC Server").

H.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.

H.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.

H.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.

H.2.6 Additional Exceptions

Additional exceptions include:

- 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 I.1, "Oracle Business Analytics Warehouse Installation and Initial Configuration Issues"
- Section I.2, "Informatica and Loading Issues"
- Section I.3, "Unicode Issues with Informatica and the Oracle Business Analytics Warehouse"
- Section I.4, "UNIX Installation Issues with the Oracle Business Analytics Warehouse"
- Section I.5, "Oracle Business Analytics Warehouse Upgrade Issues"
- Section I.6, "Oracle BI Interactive Dashboard Issues"
- Section I.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)

I.1 Oracle Business Analytics Warehouse Installation and Initial Configuration Issues

Table I–1 provides information about problems and solutions related to the installation and initial configuration of the Oracle Business Analytics Warehouse.

Table I–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 Section 3.3, "IBM DB2 UDB-Specific Database Guidelines for Oracle Business Analytics Warehouse".
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 Appendix C.3, "Changing the Default Currency in Analytics Applications".

I.2 Informatica and Loading Issues

Table I–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 I–2 Informatica and Loading Issues

Symptom/Error Message	Probable Cause/Solution
Double-clicking the workflow yields an error message in Informatica PowerCenter Workflow Manager: '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 in Informatica PowerCenter 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 I–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.	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: 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 Upon completion of executing these commands, continue executing the workflow processes to load the Oracle Business Analytics Warehouse.
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.	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. The Extended MAPI Error. MAPILogonEx failed[2147746065] error indicates that the logon is not configured correctly. Check the following: 1. Choose Services, then Informatica, then Logon, and make sure the login (domain\username) and password are correct. 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. 3. Choose Programs, then Informatica Server, then Informatica Server Setup, then Miscellaneous, and make sure the MS Exchange profile is correct.
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 I–2 (Cont.) Informatica and Loading Issues

Symptom/Error Message	Probable Cause/Solution
Oracle 8i sessions running in bulk mode fail and return an error message similar to: WRITER_1_1_1> CMN_1022 Database driver error... CMN_1022 [ORA-00600: internal error code, arguments: [kpodpmop_01], [2], [], [], [], [],[], [] Or MAPPING> TE_7022 TShmWriter: Initialized MAPPING> Sat Jan 26 13:54:45 2002 MAPPING> TE_7001 Internal error: Failed to allocate a target slot. Sat MAPPING> Jan 26 13:54:45 2002 TE_ 7017 Failed to Initialize Server MAPPING> Transformation BLK_ALL_DATATYPES1 Sat Jan 26 13:54:45 2002MAPPING> TM_6006 Error initializing DTM for session...MAPPING> TM_6020 Session [s_BULK_LONG] completed at [Sat Jan 26 13:54:45 2002]	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 Informatica PowerCenter Workflow Manager, navigate to the Targets dialog, and change the Target Load type to Normal.
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: 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	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. To correct the problem, do the following: **1.** Download the file IBM01140.ucs from ftp://ftp.software.ibm.com/ps/products/db2/fixes/english/siebel/siebel7/Conversion_Files to the /sqllib/conv directory. **2.** Make a copy of the file and rename it to IMB05348.ucs.
[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 = 42806This issue comes up when you use 'Unicode' data movement in Informatica.	The target table that the mapping is inserting into is a non-unicode table. Make sure the table is created as an Unicode table.

Table I–2 (Cont.) Informatica and Loading Issues

Symptom/Error Message	Probable Cause/Solution

The ELT run fails 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].`
	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.

I.3 Unicode Issues with Informatica and the Oracle Business Analytics Warehouse

Table I–3 provides information about problems and solutions related to Unicode issues with Informatica and the Oracle Business Analytics Warehouse.

Table I–3 Unicode Issues with Informatica and Oracle Business Analytics Warehouse

Symptom/Error Message	Probable Cause/Solution
Task fails and generates the error message: 'TRANSF_1_1_1_1> TE_7073 Aggregate Error: Expecting keys to be ascending.' Tasks known to fail are SDE_DtlForecastFact and SDE_CostList	In Informatica PowerCenter Designer, open the corresponding mapping and in the Aggregator transformation, remove the check from the Sortedinput check box.
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 Informatica PowerCenter 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.	Share the folder under a new name with no double-byte characters, and use the network share as the value for $PMRootDir.

Table I–3 (Cont.) Unicode Issues with Informatica and Oracle Business Analytics

Symptom/Error Message	Probable Cause/Solution
The database connection code page is incorrect for Unicode/Non-Latin code pages.	The database connection code page has to be set manually in Informatica PowerCenter Workflow Manager. 1. In Informatica PowerCenter Workflow Manager, connect to the Oracle_BI_DW_Base using Administrator as the user. 2. Click Menu Connection, then Select Relational. 3. Select the appropriate OLTP connection (DB2_OLTP for DB2, ORA_OLTP for Oracle, MSSQL_OLTP or ODBC_OLTP for MSSQL). 4. For the code page select the UTF-8 encoding of Unicode or the appropriate code page, and then click OK. 5. Select the OLAP connection (DB2_OLAP for DB2, ORA_OLAP for Oracle, MSSQL_OLAP or ODBC_OLTP for MSSQL). 6. For the code page select the UTF-8 encoding of Unicode or the appropriate code page, and then click OK.
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 `locale` at the UNIX command prompt. Make sure the correct locales are installed on the UNIX machine. Set the environment variable PM_CODEPAGENAME correctly (For example set it to MS932 for Japanese).

Symptom/Error Message: 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.'	Set the LC_ALL setting. For example:
	`LANG=en_US`
	`export LANG`
	`LC_ALL=en_US`
	`export LC_ALL`

I.4 UNIX Installation Issues with the Oracle Business Analytics Warehouse

Table I–4 provides information about issues related to installing the Oracle Business Analytics Warehouse on UNIX.

Table I–4 UNIX Installation Issues with Oracle Business Analytics Warehouse

Symptom/Error Message	Probable Cause/Solution

When restoring the Informatica Repository, the following error message appears in the dwdb.log:
ERROR: OBJM_54543 [Tue Mar 25 17:47:16 2003]: (164 | 2484) DataBase error:
ORA-01461: can bind a LONG value only for insert into a LONG column
	The database is Unicode and the environment variable is needed.
	1. For Unicode UTF8, set the environment variable to:
	`American_America.UTF8`
	2. For Unicode AL32UTF8, set the environment variable to:
	`American_America.AL32UTF8`

When starting pmserver on UNIX, the following error message appears: INFO: LM_36039 [Thu Mar 27 11:40:29 2003]: (6772 | 1) The maximum number of sessions that can run simultaneously is [10]. 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]: [] FATAL ERROR: SF_34004 [Thu Mar 27 11:40:29 2003]: (6772 | 1) Server initialization failed. INFO: SF_34014 [Thu Mar 27 11:40:29 2003]: (6772 | 1) Server shut down.	In the pmserver.cfg file, change MaxSessions to `15` and change LMSharedMem to `3000000`. **Note:** For each 10 count increase of MaxSessions, increase the LMSharedMem by 2000000 bytes.
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: sdcb50a125{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}

I.5 Oracle Business Analytics Warehouse Upgrade Issues

Table I–5 provides information about problems and solutions related to issues with upgrading the Oracle Business Analytics Warehouse.

Table I–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.	Before upgrading, all leading and trailing spaces need to be removed from the data.
	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: `set(col)=rtrim(col)`
	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:
	`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'}`

I.6 Oracle BI Interactive Dashboard Issues

Table I–6 provides information about problems and solutions related to Oracle BI Interactive Dashboards.

Table I–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,kllcqas:kllsltba) at OCI call OCIStmtFetch. [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: ▪ Increase pga_aggregate_target. ▪ Decrease sort_area_size and/or hash_area_size. ▪ Move to multi-threaded server (a.k.a. MTS or shared servers).

I.7 ETL Issues

Table I–7 provides information about problems and solutions related to ETL in Oracle Business Intelligence Applications.

Table I–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.

J

About the Versioned Informatica Repository

This section explains the versioned Informatica Repository that is included in Oracle Business Intelligence Applications, and contains the following topics:

- Section J.1, "Summary of Versioning in the Informatica Repository"
- Section J.2, "Customization of Repository Objects Using Check Out and Check In"

J.1 Summary of Versioning in the Informatica Repository

Oracle Business Intelligence Applications 7.9.5 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 later, these Informatica repository files are now versioned. Because of this change, you need the Team based Development option. This option is included in the OEM license provided to you by Oracle. 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 Business Intelligence 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.

J.2 Customization of Repository Objects Using Check Out and Check In

When working with an Informatica repository that is versioned, 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.

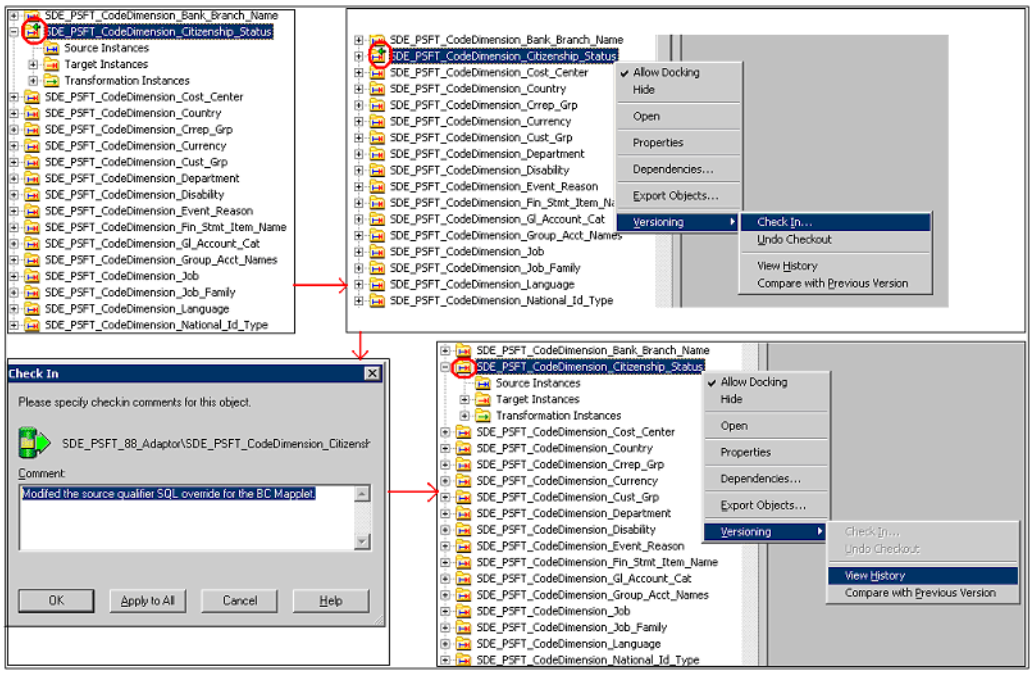
  The green arrow is still displayed to indicate this is still a checked out object.

- 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 now also compare versions and view history for this object.

  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 about working with versioned objects.



Index

Symbols

$$PERIOD
  about, 10-19
  configuring, 10-19
$$WH_DATASOURCE_NUM_ID
  about, 10-18
  configuring, 10-19

A

action links
  about, D-6
  testing, D-6
alignments dimension, configuring, E-10
ANSI NULL option, setting, 3-8
Applications
  Oracle Contact Center Telephony Analytics, 8-1, 12-1, 13-1
  Oracle Financial Analytics, 9-1
  Oracle HR, 11-1
  Oracle Order Management and Fulfillment Analytics, 10-1
  Oracle Sales Analytics, 12-1
  Oracle Supply Chain Analytics, 8-1, 12-1
AR Aging calculations
  including UnApplied Payments, 9-13
attributes
  historically significant attributes, identifying, 5-32

B

browsers
  settings, configuring, D-7
bulkload mode, error while creating a custom session, I-4

C

calculations
  including UnApplied Payments in AR Aging calculations, 9-13
Catalog Manager
  using, 2-6
charts
  local seed data, importing, C-2
  translation tables, process of maintaining, C-1
  W_LOCALIZED_STRING_G table, adding custom translations, C-4
  Web catalog strings, about translating, C-5
circular hierarchy exceptions, about and example, H-3
code lookup
  about, 5-35
  code mappings, 5-36
  codes mapplets, 5-37
  sessions for lookups, configuring, 5-37
  W_CODES table, 5-36
code page
  environments, 4-7
Communications, Media, and Energy Analytics
  dashboards and pages, E-9
  data requirements, E-10
  order management analytics, about, E-7
  predictive scores, developing and deploying, E-4
  security settings, 6-40
complex products defined, E-7
configuration steps
  all source systems, 7-1
  controlling data set
    Oracle Contact Center Telephony Analytics, 13-1
    Oracle Financial Analytics, 9-1, 9-13
    Oracle Human Resources Analytics, 11-1
    Oracle Procurement and Spend Analytics, 8-1
    Oracle Sales Analytics, 12-1
    source independent, 7-1
    Supply Chain and Order Management Analytics, 10-1
  Oracle BI Repository, 14-1
  Oracle EBS-specific, 7-18
  PeopleSoft-specific, 7-28
  required before full load
    Oracle Contact Center Telephony Analytics, 13-1
    Oracle Financial Analytics, 9-1
    Oracle Human Resources Analytics, 11-1
    Oracle Procurement and Spend Analytics, 8-1
    Oracle Sales Analytics, 12-1
    source independent, 7-1
    Supply Chain and Order Management, 10-1
configuring, E-11

DAC client, 4-14
DAC metadata repository, A-7
email recipients in DAC client, 4-49
fiscal calendar, 7-5
Oracle Business Analytics Warehouse database, 4-34
SSE role, 4-9
conformed dimensions
universal source, configuring for, 5-44
Contact Center Telephony Analytics
dashboards and pages, E-5
data requirements, E-7
Cost List exceptions
See also exception reports
about and example, H-2
Cost lists, using to display, H-6
products without a cost list, H-3
CRM and EBS schema changes
verifying, 4-61
cross-module configuration, performing
codes lookup, 5-35
conformed dimensions, configuring, 5-44
dimension keys, resolving, 5-37
domain values, working with, 5-39
loads, configuring, 5-28
records, filtering and deleting, 5-28
slowly changing dimensions, changing, 5-31
stored lookups, configuring, 5-35
currency, diagnosing translation issues, H-5
Customer Status History Fact table
about, 10-18
configuring, 10-19

D

DAC
configuring metadata repository, A-7
Source System Parameters, 4-65
starting and stopping the server, A-2
DAC metadata repository
database, 4-9
DAC Server
starting and stopping manually, A-2
dashboards
CME dashboards and pages, E-9
consumer sector dashboards and pages, E-5
customizing, about, D-8
exposing dashboards and requests, D-14
information, about viewing information, D-14
locating dashboard and page names, D-14
viewing a list of, 2-6
views, adding, D-11
data import
syndicated data loading, overview, F-2
data loading
Oracle's Siebel Analytics for Life Sciences, issues when loading, F-30
troubleshooting, Informatica and data loading issues, I-2
Data Movement Mode, 4-7
data movement mode, 4-7
Data Source NUM ID
configuring $$WH_DATASOURCE_NUM_ID, 7-17
data warehouse
database connections, 4-51
Data Warehouse Administration Console (DAC)
about, 2-4
configuring on UNIX, A-5
configuring the client, 4-14
setting up system properties, A-7
database connections
transactional, creating, 4-51
database requirements, 4-9
DataMovementMode parameter, 4-7
DATASOURCE_NUM_ID
about, 7-17
configuring, 7-17
DB Library Options setting, modifying, 3-8
DB2 databases
DB2CODEPAGE, 4-11
environment variable, 4-11
Oracle Business Analytics Warehouse requirements, 3-4
Oracle Business Analytics Warehouse, cannot connect to, I-2
DB2-UDB
creating stored procedures, 4-62
deployment
topology example, 4-1
dimensions
alignments dimension, configuring, E-10
key resolution, using a lookup, 5-38
keys, resolving, 5-37
pharmaceuticals, configuring dimensions specific to, E-12
positions dimension, configuring, E-10
product costs, configuring, E-11
products dimension, configuring, E-11
documentation
Repository Documentation feature, 2-6
domain values
domain value conversion process, understanding, 5-39
domain value set, configuring, 5-43
domain value set, extending, 5-42
importance, understanding, 5-40
working with, 5-39

E

email
configuring Email in the DAC Server, 4-64
configuring email recipients in the DAC Client, 4-63
notification in Outlook error when sending post session email, I-4
environment variables
DB2CODEPAGE, 4-11
NLS_LANG, 4-10

ETL
error with W_UAEMP_TMP and W_UAPOS_TMP table, I-10
History view, H-6
improve performance using partitioning, 3-18
index errors, I-10
running with Non-English OLTP Data Source, 3-17
troubleshooting, 12-3, I-10
ETL failure, I-10
Event Types
configuring for PSFT and HR, 11-32
exception reports
additional exceptions, list of, H-6
circular hierarchy exceptions, about and example, H-3
cleansing the data, process and diagram, H-4
ETL History view, using to view the history of the ETL processes and statuses, H-6
exception report components, list of, H-1
exception, fixing, H-4
exchange rate exceptions, about, H-3
Exchange Rate view, using to diagnose currency translation issues, H-5
executing the report, cleansing steps, H-4
invalid hierarchy exceptions, about, H-3
List of Values exceptions, H-1
List of Values view, using to compare values with Oracle Business Analytics Warehouse, H-5
See also Cost List exceptions
system preferences
exchange rate exceptions, about, H-3
Exchange Rate view, using to diagnose currency issues, H-5
extracts, configuring
extracting additional data, about, 5-25
extracting data from new source table, 5-26
extracting new data from existing source table, 5-26

F

fiscal calendar
configuring, 7-5
fiscal month
configuring, 7-5
fiscal week
configuring, 7-5

H

Healthcare Analytics
security settings, 6-41
hierarchical relationships, invalid hierarchy exceptions, about, H-3
historically significant attributes
identifying, 5-32
home page
content, customizing, D-8
customizing, about, D-8
path argument, determining, D-10
Horizontal applications
indexing for performance, 3-16
HR Analytics
configuring Event Types for PSFT, 11-32

I

Incentive Compensation Analytics, turning on, D-7
indexing
S_.* tables, 3-16
to improve performance, 3-16
Informatica
default passwords, 4-19
fatal errors 11, I-6
fatal errors 3644-1 or 360454, I-6
username and password, 14-1
Informatica PowerCenter
requirements, 4-6
terminology changes in 8.1.1, 4-6
Informatica Repository
versioned, J-1
Informatica repository
database, 4-9
dropping, A-4
Informatica Server
password, 4-19
registering, 4-49
infrastructure requirements, 4-5
Initialization Blocks, 3-15
installation
database connectivity, 4-28
Java SDK, 4-14
JDBC drivers, 4-28
Oracle BI Applications installation process task list, 4-3
installation and configuration
task list, 4-3
installing
Oracle BI repository, setting up, 14-2
Insurance Analytics family of products
security settings, 6-41
IP address
virtual IP address, creating, D-5
virtual IP address, testing, D-5

J

Java JDK
requirements, 4-14
Java SDK
installation, 4-14
JDBC drivers
installation, 4-28

L

large fact tables
partitioning, 3-18
List of Values
exceptions, about and example, H-1

view, using to compare values with the Oracle Business Analytics Warehouse, H-5
loads, configuring
about, 5-28
Source Adapter mapplet. creating, 5-27

M

Mappings
SA System Subject Area, 3-14
metadata
externalizing strings, C-3
object level security, repository groups, 6-29
Web Catalog group security, 6-30
Web Catalog security, 6-30
metadata object level security
repository groups, 6-29
Microsoft Outlook
closes when finishing ETL process, probable cause, I-4
error when sending post session email, I-4
Migrating Seed Data from Siebel OLTP Source Systems, 4-66

N

NLS_LANG environment variable, 4-10
Non-English OLTP Data Source
running an ETL with, 3-17

O

OPB_OBJECT_LOCKS Informatica session deadlocked, probable cause, I-4
Operational Applications
dashboards, adding views, D-11
home page content, customizing, D-8
home page path argument, determining, D-10
Oracle
mappings hang, probable cause, I-3
Oracle BI Administration Tool
using to generate Repository Documentation, 2-6
Oracle BI Applications
action links, about, D-6
action links, testing, D-6
browser settings, configuring, D-7
completing initialization, D-2
dashboards, adding views, D-11
home page and dashboards, about customizing, D-8
home page content, customizing, D-8
Incentive Compensation Analytics, turning on, D-7
initialization, completing, D-2
installation and configuration process task list, 4-3
operational applications, about integrating within, D-6
Optional Analytics Applications, D-7
virtual IP address, creating, D-5
Web Extension, configuring to work with, D-4
Oracle BI Applications, configuring
home page, determining path argument, D-10
virtual IP address, testing, D-5
Oracle BI Charts
metadata strings, externalizing, C-3
Oracle BI seed data, updating for non-English locales, C-2
Oracle BI EE
requirements, 4-5
Oracle BI Presentation layer
process of maintaining translation tables, C-1
Oracle BI repository
externalizing metadata strings, C-3
repository groups, security, 6-29
Oracle BI SE One
requirements, 4-5
Oracle BI seed data
updating for non-English locales, C-2
Oracle BI Supply Chain Analytics
domain values and CSV worksheet files, 8-4
universal source, configuring expenses, 8-14
Oracle Business Analytics Warehouse
architecture components, 2-3
database, 4-9
database parameters and requirements, 3-2
database, configuring, 4-34
DB2 requirements, 3-4
operating system, driver, connectivity requirements, 3-2
SQL server-specific requirements, 3-7
tables, creating, 4-35
Teradata requirements, 3-10
Oracle Contact Center Telephony Analytics
Benchmarks and Targets table, setting up, 13-8
Contact Representative Snapshot table, setting up, 13-7
Contact Status Type columns, about configuring for, 13-6
dates and times, configuring, 13-11
dimension key resolution, 13-2
dimension table, loading, 13-2
Event Type columns, about configuring for, 13-3
fact table, loading, 13-3
overview, 13-1
Representative Activities table, setting up, 13-6
Oracle databases
NLS_LANG, 4-10
Oracle Financial Analytics
configuration steps before full load, 9-2
disabling logical table sources, E-3
externalizing metadata translation strings, E-3
logical table sources, disabling, E-3
logical table sources, updating, E-2
metadata translation strings, externalizing, E-3
overview, 9-1
updating logical table sources, E-2
Oracle HR Analytics, 11-33
domain values and CSV worksheet files, 11-3
overview, 11-1
Oracle Order Management and Fulfillment Analytics

options, 10-1
overview, 10-1
Oracle Partner Analytics
role-based analytics dashboard mapping, 6-35
Oracle Pharma Sales Analytics applications
security settings, 6-43
Oracle Procurement and Spend Analytics
configuration steps before full load, 8-3
overview, 8-1
Oracle's Siebel Industry Applications
Oracle Pharma Sales Analytics security settings, 6-43
Oracle's Siebel Financial Services security settings, 6-41
Oracle's Siebel Life Sciences
data loading issues, F-30
Oracle's Siebel Operational Applications
Oracle Partner Analytics security settings, 6-35
Oracle's Siebel Industry Applications
alignments dimension, configuring, E-10
Analytics applications, about metadata requirements, E-5
CME dashboards and CME Analytics applications dashboards and pages, E-9
CME data requirements, E-10
CME security settings, 6-40
consumer sector dashboards and pages, E-5
consumer sector data requirements, E-7
consumer sector security settings, 6-40
dimensions, configuring specific to pharmaceuticals, E-12
Pharma sales effectiveness, configuring subject area, E-12
product costs dimension, configuring, E-11
Usage Accelerator Analytics security settings, 6-45
Oracle's Siebel Industry applications
alignments dimension, configuring, E-10
Analytics applications, about metadata requirements, E-5
CME dashboards and pages, E-9
CME data requirements, E-10
consumer sector dashboards and pages, E-5
consumer sector data requirements, E-7
dimensions, configuring specific to pharmaceuticals, E-12
Pharma medical education effectiveness, configuring subject area, E-13
Pharma product categories, configuring subject area, E-12
Pharma promotional effectiveness, configuring subject area, E-12
Pharma Sales Analytics, about metadata requirements, E-10
Pharma sales effectiveness, configuring subject area, E-12
positions dimension, configuring, E-10
predictive scores, developing and deploying, E-4
product costs dimension, configuring, E-11
Oracle's Siebel Operational Applications
integration with data types, D-1
integration with Industry Applications, D-1
security
OracleBIAnalyticsApps.rpd
Informatica username and password, 14-1
order management analytics, about, E-7
Order Management and Fulfillment Analytics
multiple products, about tracking, 10-14
Outlook
closes when finishing ETL process, probable cause, I-4
error when sending post session email, I-4

P

page names, locating and dashboards, D-14
parent-child relationship, circular hierarchy exceptions, H-3
partitioning, 3-18
password
Informatica administrator, 4-19
OracleBIAnalyticsApps.rpd, 14-1
Payroll table, about aggregating, 11-33
PeopleSoft Trees, 9-20
PerfIndex_Horizontal.sql, 3-16
PerfIndex_Industry.sql, 3-16
performance
components configured to optimize performance, 3-8
implementing indexes to improve, 3-16
partitioning large fact tables, 3-18
Pharma objectives achievement
configuring subject area, E-13
medical education effectiveness, configuring subject area, E-13
Pharma product categories
configuring subject area, E-12
Pharma promotional effectiveness, configuring subject area, E-12
Pharma Sales Analytics
alignments dimension, configuring, E-10
dimensions, configuring specific to pharmaceuticals, E-12
medical education effectiveness, configuring subject area, E-13
metadata requirements, about, E-10
objectives achievement, configuring subject area, E-13
Pharma medical education effectiveness, configuring subject area, E-13
Pharma objectives achievement, configuring subject area, E-13
Pharma product categories, configuring subject area, E-12
Pharma promotional effectiveness, configuring subject area, E-12
Pharma Sales effectiveness, configuring subject area, E-12
positions dimension, configuring, E-10

product categories, configuring subject area, E-12
product costs dimension, configuring, E-11
products dimension, configuring, E-11
promotional effectiveness, configuring subject area, E-12
sales effectiveness, configuring subject area, E-12
security settings, 6-43
Pharma Sales Analytics applications
dimensions, configuring specific to pharmaceuticals, E-12
metadata requirements, about, E-10
Pharma medical education effectiveness, configuring subject area, E-13
Pharma objectives achievement, configuring subject area, E-13
Pharma product categories, configuring subject area, E-12
Pharma promotional effectiveness, configuring subject area, E-12
Pharma Sales effectiveness, configuring subject area, E-12
positions dimension, configuring, E-10
product costs dimension, configuring, E-11
products dimension, configuring, E-11
Pharma Sales Analytics, about metadata requirements, E-10
positions dimension, configuring, E-10
predictive analytics metadata
predictive scores, developing and deploying, E-4
predictive scores, developing and deploying, E-4
Presentation Catalog, administering
dashboards, about viewing information, D-14
dashboards, exposing and requests, D-14
dashboards, locating and page names, D-14
Presentation Services Catalog
viewing, 2-6
primary extract and delete mapping
delete and primary extract sessions, enabling, 5-31
source-archived records, deletion configuration, 5-30
working with, 5-30
PRM Analytics
data-level visibility, 6-39
portal-based analytics dashboard mapping, 6-37
subject area mapping, 6-38
subject area visibility, 6-39
product costs dimension, configuring, E-11
products dimension, configuring, E-11

Q

Query, I-2
Query Analyzer, can not open after installing ODBC driver, I-2
query performance
improve using partitioning, 3-18

R

RDBMS error message, probable cause, I-3
records
source-archived records, deletion configuration, 5-30
records, filtering and deleting
about, 5-28
primary extract and delete mappings, understanding, 5-29
primary extract and delete mappings, working with, 5-30
repository
Oracle BI repository, setting up, 14-2
versioned, J-1
Repository Documentation
generating, 2-6
repository, setting up, 14-2
requests
exposing dashboards and requests, D-14
requirements
database, 4-9
Informatica PowerCenter, 4-6
infrastructure, 4-5
Java SDK, 4-14
Oracle BI EE, 4-5
Oracle BI SE One, 4-5
responsibilities
in Usage Accelerator, 12-3, I-10

S

S_.* tables
indexing, 3-16
SA System Subject Area
mapping, 3-14
schema definitions
transactional database, updating, 4-59
SDEINC_RecordExtractStart session fails, probable cause, I-3
security
CME security settings, 6-40
CME settings, 6-40
Communications, Media, and Energy Analytics, 6-40
consumer sector security settings, 6-40
data-level security, about, 6-56
data-level security, implementing, 6-58
default security settings, 6-3
embedded analytics support for changing positions (Siebel App), 6-54
employee/position based security, about, 6-31
initialization blocks, 3-15
integrating data security with EBS, 6-3
integrating data security with Oracle's PeopleSoft Enterprise Apps, 6-19
metadata object level security (repository groups), about, 6-29
operational applications security
Oracle Financial Analytics settings, 6-41
Oracle Partner Analytics security settings, 6-35

Oracle Pharma Sales Analytics security settings, 6-43
Oracle's Siebel Usage Accelerator Analytics security settings, 6-45
primary position based security for CRM vertical applications, about, 6-34
security categories, about, 6-1
team based visibility, 6-47
team based visibility, DAC metadata, 6-52
types of security, about, 6-1
user responsibilities, checking, 6-2
Web Catalog security, 6-30
seed data
importing, C-2
seed data, importing, D-1
server components, installation requirements
See also server component software
setting up
DAC system properties, A-7
Siebel Financial Analytics
PeopleSoft Trees, 9-20
simple products defined, E-8
slowly changing dimensions, configuring
about, 5-31
effective dates, 5-34
historically significant attributes, identifying, 5-32
Type I changing dimension, 5-33
Type II slowly changing dimension, 5-33
Source Adapter mapplet
creating, 5-27
Source System Parameters
about, 4-65
setting, 4-65
Source System Parameters (also known as DAC ETL Preferences), 4-65
source-archived records
deletion configuration, 5-30
SQL Server
DB Library Options setting, modifying, 3-8
Oracle Business Analytics Warehouse requirements, 3-7
SSE role
creating, 4-9
requirements, 4-9
starting the DAC Server, A-2
stored lookups
configuring, 5-35
stored procedures for DB2-UDB
creating, 4-62
string translations
adding for metadata, C-4
SWE
See Web Extension
syndicated data
importing, overview, F-2
system preferences
See also exception reports

T

Task List
installation and configuration, 4-3
Teradata
activating join indexes, A-8
avoiding arithmetic overflows, 3-10
code page file property, setting, 4-58
configuration checklist, Windows, 4-63
creating and dropping tables, 4-38
data warehouse requirements, 3-10
database hints, 3-10
general guidelines, 3-10
Hosts file, setting up, 4-58
terminology
changes in Informatica PowerCenter 8.1.1, 4-6
topology
deploying Oracle BI Applications, 4-1
example, 4-1
Transactional database
connection error, I-2
Cost lists, displaying, H-6
importing locale seed data, C-2
translation tables
process of maintaining, C-1
troubleshooting
ETL error, I-6, I-10
ETL failure, 12-3, I-10
general, I-1
Informatica and data loading issues, I-2
Oracle Business Analytics Warehouse installation, I-2
Unicode, I-6, I-9, I-10
UNIX installation, I-7
W_UAEMP_TMP, I-10
W_UAPOS_TMP, I-10

U

UnApplied Payments
Using in AR Aging calculations, 9-13
Unicode
local seed data, importing, C-2
metadata strings, externalizing, C-3
Oracle BI seed data, updating for non-English locales, C-2
string translations, adding for metadata, C-4
translation tables, process of maintaining, C-1
W_LOCALIZED_STRING_G table, adding custom translations, C-4
Web catalog strings, about translating, C-5
Universal Adapter
conformed dimensions, configuring, 5-44
for CRM Analytics, B-1
Oracle BI Supply Chain Analytics, configuring expenses, 8-14
URL
dashboards, configuring for, D-11
home page, configuring for, D-8
Usage Accelerator
ETL failure, 12-3, I-10

Usage Accelerator Analytics security settings, 6-45
username
OracleBIAnalyticsApps.rpd, 14-1

V

verifying schema changes in Oracle's Siebel (CRM) and EBS, 4-61
versioned Informatica Repository, J-1
versioned objects
Informatica Repository, J-1
Vertical applications
indexing for performance, 3-16
views, adding to dashboards, D-11
virtual IP address
creating, D-5
testing, D-5

W

W_LOCALIZED_STRING_G table, adding custom translations, C-4
W_UAEMP_TMP
ETL errors, I-10
W_UAPOS_TMP
ETL errors, I-10
Web Catalog groups
Analytics applications, security in, 6-30
Web Catalog strings
about translating, C-5
Web Extension
configuring to work with, D-4
virtual IP address, creating, D-5
virtual IP address, testing, D-5
workflows
error message when executing on the Informatica Server Manager, probable cause, I-3
Server Manager error message, probable cause, I-3