



Oracle® Business Intelligence Applications Installation and Configuration Guide

Version 7.9

February 2007

Part Number: B31979-01

Copyright © 2007, Oracle. All rights reserved.

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

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

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

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

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

Contents

Chapter 1: What's New in This Release

Chapter 2: Overview of Oracle Business Intelligence Product Licensing

Oracle Business Intelligence Product Licensing 19

Chapter 3: Overview of Oracle Business Analytics Warehouse

Oracle Business Analytics Warehouse Overview 23

Oracle Business Analytics Warehouse Architecture 24

Oracle Business Analytics Warehouse Architecture Components 25

Chapter 4: Preinstallation and Predeployment Considerations for Oracle BI Applications

Oracle Business Analytics Warehouse Deployment Configuration 28

Operating System, Driver, and Connectivity Requirements for Oracle Business Analytics Warehouse 29

General Guidelines for Setting Up Oracle Business Analytics Warehouse 30

IBM DB2 UDB-Specific Database Guidelines for Oracle Business Analytics Warehouse 32

IBM DB2 UDB zOS and OS/390 and Z/os-Specific Database Guidelines for Oracle Business Analytics Warehouse 33

Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse 34

Additional Suggestions for Optimizing Oracle Performance in Oracle Business Analytics Warehouse 43

SQL Server-Specific Database Guidelines for Oracle Business Analytics Warehouse 45

Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse 47

General Guidelines for Teradata Deployments 47

Best Practices For Teradata Deployments 48

Deploying Oracle Business Analytics Warehouse in Latin-1 General, Unicode and Non-English Environments	53
Code Page Overview	54
Source and Data Warehouse Code Pages for Latin-1 General (7-bit ASCII) to Latin-1 General (7-bit ASCII)	54
Source and Data Warehouse Code Pages for Latin-1 General (8-bit ASCII) to Latin-1 General (8-bit ASCII)	56
Source and Data Warehouse Code Pages for Unicode to Unicode	58
Source and Data Warehouse Code Pages for Code Page to Unicode	59
Source and Data Warehouse Code Pages for Code Page to Code Page	61
Setting the NLS_LANG Environment Variable for Oracle Databases	63
Setting the DB2CODEPAGE Environment Variable for DB2 Databases	64
Miscellaneous Information About Oracle Business Intelligence Applications Deployments	64
Preconfigured Mapping for the SA System Subject Area	65
Using Initialization Blocks	66
Performance Improvements for Change Capture for Incremental ETL Run	67

Chapter 5: Installing and Configuring Oracle BI Applications on Windows

Installation and Configuration Process Task List for Windows	69
Installing the Java SDK	71
Running the Oracle Business Intelligence Applications Installation Wizard (Windows)	72
Running the Informatica PowerCenter Installation Wizard (Windows)	75
Setting Up the Informatica Server and Informatica Repository Server	79
Process Of Setting Up The Informatica Server	79
Process Of Setting Up The Informatica Repository Server	83
Process of Copying Source Files and Lookup Files	84
Process of Starting the Informatica Services	84
Process of Creating ODBC Database Connections	85
Setting Environment Variables on Windows	86
Installing the Informatica Patch for Teradata	88
Setting Informatica Data Code Pages Validation	88
Process of Copying Hibernate Libraries	88
Configuring the DAC Client	89
Process of Configuring the DAC Config.bat File	89
Installing JDBC Drivers for DAC Database Connectivity	90

Creating the Required Databases	90
Creating the DAC Metadata Repository	91
Configuring the DAC Metadata Repository	95
Setting Up DAC System Properties	96
Specifying Transactional and Data Warehouse Data Sources in the DAC	100
Configuring Email Recipients in the DAC Client	103
Activating Join Indices for Teradata Databases	104
Configuring the Oracle Business Analytics Warehouse Database	104
Process of Creating the SSE Role	105
Process of Creating Data Warehouse Tables	105
Process of Creating Data Warehouse Tables on a Teradata Database	109
Registering Informatica Servers in the DAC Client	110
Process of Configuring and Managing the Informatica Repository	113
Process of Registering the Informatica Repository Server	114
Process of Loading the Pre-Built Repository Into Informatica	115
Process of Configuring the Informatica Repository in Workflow Manager	118
Process of Stopping and Starting the Informatica Repository (Optional)	121
Process of Backing Up and Restoring the Informatica Repository (Optional)	122
Setting the Code Page File Property for Unicode Environments on Teradata Databases	124
Setting Up the HOSTS File for Teradata Installations on Windows	124
Verifying the Informatica Repository Configuration	125
Setting Up The Time Dimension	126
Process of Setting Up The Fiscal Calendar	127
Configuring the DAC Server	129
Process of Configuring the Connection Between the DAC Server and DAC Repository	129
Process of Configuring Email in the DAC Server (Optional)	130
Process of Setting DAC Source System Parameters	131
Process of Starting the DAC Server	137
Starting the Informatica Server	138
Updating Siebel Transactional Database Schema Definitions	138
Process of Applying Siebel (CRM) and Oracle E-Business Suite Schema Changes For Oracle, DB2/UDB, MSSQL	139
Process of Applying CRM Schema Changes For DB2 on 390 or z/os	140
Process of Verifying the Siebel (CRM) and Oracle E-Business Suite Schema Changes	141
About Delete Triggers	141
Creating Stored Procedures for DB2-UDB	142

Teradata-Specific Installation and Configuration Checklist (Windows)	143

Chapter 6: Installing and Configuring Oracle BI Applications on UNIX

Overview of Installing and Configuring Oracle BI Applications on UNIX	145
Installation and Configuration Process Task List for UNIX	146
Installing Oracle BI Applications on UNIX	150
Installing and Configuring the Informatica Repository Server on UNIX	151
Installing and Configuring the Informatica Server on UNIX	153
Updating the Informatica License Files on UNIX	160
Installing the Informatica Patch on UNIX	161
Creating Database Connections for Teradata Installations	161
Setting Environment Variables for the Informatica Server on UNIX	162
Setting Up the HOSTS File for Teradata Installations on UNIX	163
Starting the Informatica Repository Server on UNIX	163
Stopping the Informatica Repository Server on UNIX	164
Configuring the DAC Server on UNIX	164
Starting the Informatica Server on UNIX	167
Stopping the Informatica Server on UNIX	168
Starting the DAC Server on UNIX	168
Stopping the DAC Server on UNIX	169
Teradata-Specific Installation and Configuration Checklist for UNIX	169

Chapter 7: Customizing the Oracle Business Analytics Warehouse

Oracle Business Analytics Warehouse Customization Scenarios	171
Type I Customizations: Adding Columns to Existing Fact or Dimension Tables	172
About Extending Mappings	173
Impact of Customization on Upgrade	174
Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse	177
Scenario for Extending the Oracle Business Analytics Warehouse	178
Other Types of Customizations Requiring Special Handling	185
Modifying Type II SCD Triggers	185
Adding Dimensions to Existing Facts	186
Adding Date Dimensions to Existing Facts	189

Adding Currencies to an Existing Table	189
Type II Customizations: Adding Additional Tables	190
About Creating New Dimension or Fact Tables	190
Using Custom Folders in the Oracle Business Analytics Warehouse	191
Creating Custom Informatica Workflows	192
Important Notes About Customizing the Oracle Business Analytics Warehouse	193
Including a Source Table for Change Capture Process	195
Adding a New Dimension in the Oracle Business Analytics Warehouse	195
Adding a New Fact Table in the Oracle Business Analytics Warehouse	196
Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse	197
Type III Customizations: Adding New Data as a Whole Row into a Standard Dimension Table	197

Chapter 8: Configuring the Oracle BI Repository for Oracle Business Analytics Warehouse

Modifying the Oracle BI Repository for Siebel version 6.3 and version 7.5 applications	199
Configuring the Oracle BI Repository Connections	200
Setting up Date Specific Metrics	202
Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse	203
Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse	204
About Oracle BI Repository Variables	204
Presentation Services Configuration Variables	206
About User Authentication	207
About the Security or Visibility Configuration	208
About the Group Variable	211
About the Period Ago Keys for Oracle Business Analytics Warehouse	212
About Configuring Usage Tracking for Oracle Business Analytics Warehouse	213
About the Incremental Deployment of the Oracle BI Applications Repository	213

Chapter 9: Configuring Common Components of the Oracle Business Analytics Warehouse

Configuring Extracts	215
Extracting Additional Data	216

Setting Up the Delimiter for Source Files	217
Configuring a Source Adapter Mapplet	218
Configuring Loads	219
Filtering and Deleting Records	219
About Primary Extract and Delete Mappings Process	219
About Working with Primary Extract and Delete Mappings	221
Configuring Slowly Changing Dimensions	222
About Identifying Historically Significant Attributes	223
Type I and Type II Slowly Changing Dimensions	223
About Document, Local, and Global Currencies	227
Process for Configuring Currencies	227
Extracting Exchange Rates from a Custom Table	228
Configuring Global Currencies and Exchange Rate Types	228
Handling European Monetary Union (EMU) Currencies	228
Using the Flat File to Supply EMU Exchange Rates	229
About the EXP_CURR_CONVERSION_TRANSFORM Expression Transformation to Supply EMU Exchange Rates	230
About Stored Lookups	234
Codes Lookup	234
Configuring Extension Column Code Description Lookups	236
About Resolving Dimension Keys	237
Resolving the Dimension Key Using Lookup	237
About Domain Values	238
About the Domain Value Conversion Process	238
About the Importance of Domain Values	241
About Extending the Domain Value Set	242
Configuring the Domain Value Set with CSV Worksheet Files	243
Configuring the Domain Value Set Using PowerCenter Designer	244
Configuring Conformed Dimensions	245
Configuring Conformed Dimensions for Universal Source	245

Chapter 10:Configuring Oracle Business Analytics Warehouse for Oracle 11i

Configuring the Group Currency Code for Oracle 11i	247
About Adding Oracle Flex Fields to the Data Model	247
Mapping Source Customer Hierarchies to the Customers Dimension Table	248

Configuring Product Categories	250
Modifying the DATASOURCE_NUM_ID	251
Modifying the Last Extract Date	252

Chapter 11: Checklist for Configuring Oracle Business Analytics Warehouse Applications

Checklist for Initializing Oracle Business Analytics Warehouse	253
Checklist for Configuring Oracle BI Applications Consumer	253
Checklist for Configuring Oracle BI Applications Consumer	254
Checklist for Configuring Oracle Workforce Analytics	256
Checklist for Configuring Oracle Financial Analytics	257
Checklist for Configuring Supply Chain Analytics	259
Checklist for Configuring Supply Chain Analytics for Oracle 11i	259
Checklist for Configuring Supply Chain Analytics for Universal Source	260
Checklist for Configuring Supply Chain Analytics for Post-Load Processing	261

Chapter 12: Configuring Oracle BI Applications Consumer

Overview of Oracle Contact Center Telephony Analytics	263
About the Dimension Key Resolution Process for Universal Source	264
Loading the Dimension Table	264
Loading the Fact Table	264
About Configuring the Event Type Columns	265
About Configuring The Contact Reason Column	267
About Configuring Contact Status Type Column	268
Setting Up the Representative Activities Table	269
Setting Up the Contact Center Performance Table	271
Setting Up the Benchmarks and Targets Table	272
Logical Delete of Records From Fact Tables	273
Configuring Dates and Times	275
Configuring Flags for Oracle BI Applications Consumer	276
Excluding Representative Data for Post-Load Processing	277
Configuring Alternate Load Plan to Enable Direct Load to the Entry-Level Rep Activity Aggregates	278

Configuring Alternate Load Plan to Enable Direct Load to the Entry-Level Contact Performance Aggregates	278
Formation of Integration_Id of the Enterprise Contact Center Facts in the Warehouse	279
Setting Up the IVR Menu and IVR Navigation Profile Tables	280
Viewing Duration Metrics in hh:mi:ss Format in Oracle BI Answers	282

Chapter 13:Configuring Oracle's Supply Chain Analytics family of products

Overview of Oracle's Supply Chain Analytics family of products	283
Process of Configuring Supply Chain Analytics for Oracle 11i	286
Deploying Stored Procedures	286
Configuring the Bill of Materials Explosion for Oracle 11.5.9 and above version	287
Configuring the Bill of Materials Explosion Options	287
Configuring the Left Bound and Right Bound Calculation Option	292
Configuring Quantity Types for Product Transactions	294
About Configuring the Handling of Currency Types	294
Configuring the Country Region and State Region Name	295
Configuring the State Name	296
Configuring the Country Name	296
Configuring the Make-Buy Indicator	297
Extracting Particular Purchase Order Records	298
Configuring the DAC parameter for Purchase Cycle Line	298
Process of Aggregating Supply Chain Analytics Tables	299
About Configuring the Inventory Monthly Balance Table	299
Configuring the Inventory Monthly Balance Table	300
About Configuring the Product Transaction Aggregate Table	301
Configuring the Product Transaction Aggregate Table	303
About Configuring Supply Chain Analytics for Universal Source	305
Process of Configuring Supply Chain Analytics for a Universal Source	306
Configuring the Preferred Merchant Flag	306
Configuring the Customer Billable Indicator	306
Configuring the Receipts Indicator	307
Configuring Expense Payment Types	307
Configuring the Default Expense Distribution Percentage	308
Configuring Lookup Dates for Currency Conversion	309
About Configuring the Purchase Receipts Aggregate Table	309
Configuring the Purchase Receipts Aggregate Table	310
About Configuring the Purchase Cycle Lines Aggregate Table	311

Configuring the Purchase Cycle Lines Aggregate Table 312

Domain Values and CSV Worksheet Files for Supply Chain Analytics 313

Chapter 14:Configuring Oracle Financial Analytics

Overview of Oracle Financial Analytics 315

Process of Configuring Oracle Financial Analytics for Oracle 11i 316

About the Name of the Oracle 11i Folder in Informatica 317

Extracting Data Posted at the Detail-Level for Oracle 11i 317

Configuring How GL Balances Is Populated 318

Configuring the AP/AR Aging Tables 318

Mapping Oracle General Ledger and Profitability Analytics Account Numbers to Group Account Numbers 319

Configuring Oracle General Ledger and Profitability Analytics Transaction Extracts 321

Configuring General Ledger Cost Of Goods Extract 322

Configuring the General Ledger Account Hierarchies 322

Configuring General Ledger Account Hierarchies Using General Ledger Accounting Flexfield value sets definitions 323

Configuring General Ledger Account Hierarchies Using Financial Statement Generator (FSG) Report Definition 327

Configuring the General Ledger Balance ID 329

Configuring AP Balance ID for Oracle Payables Analytics 330

Configuring AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics 331

Configuring the AR Adjustments Extract for Oracle Receivables Analytics 331

Configuring the AR Schedules Extract 332

Configuring the AR Cash Receipt Application Extract for Oracle Receivables Analytics 333

Configuring the AR Credit-Memo Application Extract for Oracle's Siebel Receivables Analytics 333

About the Customer Costs Lines and Product Costs Lines Tables for Oracle's Siebel Profitability Analytics 334

Configuring the Customer Costs Lines and Product Costs Lines Tables for Oracle's Siebel Profitability Analytics 335

About PeopleSoft Trees in Oracle Financial Analytics 336

Chapter 15:Configuring Oracle Order Management and Fulfillment Analytics Option

Overview of Oracle Order Management and Fulfillment Analytics Option 337

Process of Configuring Oracle Order Management and Fulfillment Analytics Option for Oracle 11i 338

Configuring Sales Order Lines Data Storage 339

About Tracking Attribute Changes in Bookings 341

Tracking Multiple Attribute Changes in Bookings	342
Configuring Sales Schedule Lines Data Storage	343
Configuring Early and Late Tolerances for Shipments	345
Configuring Sales Invoice Lines Data Storage	346
Configuring Different Types of Backlog Calculations	346
Accounting for Negative Values in Orders, Invoices, and Picks	348
Process of Configuring the Order Lines Execution Plan and OTB ETL	349
Process of Aggregating Oracle Order Management and Fulfillment Analytics Option Tables	349
About Configuring the Sales Invoice Lines Aggregate Table	350
Configuring the Sales Invoice Lines Aggregate Table	351
About Configuring the Sales Order Lines Aggregate Table	351
Configuring the Sales Order Lines Aggregate Table	353
About Tracking Multiple Products for Oracle Order Management and Fulfillment Analytics Option	353
Adding Dates to the Order Cycle Time Table for Post-Load Processing	354
About Configuring the Backlog Period Date for Oracle Order Management and Fulfillment Analytics Option	356
Configuring the Backlog Period Date for Oracle Order Management and Fulfillment Analytics Option	358
About the Grain at Which Currency Amounts and Quantities Are Stored	359
About the Sales Order Dates	360
Domain Values and CSV Worksheet Files for Oracle Order Management and Fulfillment Analytics Option	360
Configuring Supply Chain Analytics for Oracle Order Management and Fulfillment Analytics Option	361
Configuring Oracle's Siebel Financial Analytics for Oracle Order Management and Fulfillment Analytics Option	362

Chapter 16:Configuring Oracle Workforce Analytics

Overview of Oracle Workforce Analytics	363
About Configuring Workforce Operations for Oracle 11i	364
Process of Configuring Workforce Operations for Oracle 11i	364
Configuring the Employees Dimension for U.S. Statutory Compliance	364
Configuring Sex Codes	369
Configuring Sex Descriptions	369
Configuring the Jobs Dimension for U.S. Statutory Compliance	370
Configuring Address Types for Workforce Profile	372

Configuring Phone Types for Workforce Profile	373
Process of Configuring Workforce Payroll for Oracle 11i	374
Modifying the Workforce Payroll Filters	374
Improving ETL Performance for Workforce Payroll	376
About Aggregating the Payroll Table for Oracle Workforce Analytics	377
Aggregating the Payroll Table for Oracle Workforce Analytics	378
Domain Values and CSV Worksheet Files for Oracle Workforce Analytics	379
Configuring Domain Values and CSV Worksheet Files for Oracle Workforce Analytics	384

Chapter 17: Using Oracle Business Analytics Warehouse Exception Reports With Siebel CRM Sources

Understanding Oracle Business Analytics Warehouse Exceptions	393
Executing Oracle Business Analytics Warehouse Exception Reports	396

Chapter 18: Troubleshooting Oracle Business Intelligence Applications

Oracle Business Analytics Warehouse Installation and Initial Configuration Issues	402
Informatica and Loading Issues	403
Unicode Issues with Informatica and the Oracle Business Analytics Warehouse	408
UNIX Installation Issues with the Oracle Business Analytics Warehouse	409
Oracle Business Analytics Warehouse Upgrade Issues	413
Oracle BI Interactive Dashboard Issues	415

Appendix A: Installing the Usage Accelerator Application

Usage Accelerator Application Integration with Siebel 7.5.3 CRM Application	417
Usage Accelerator-Analytics Adapter Files	420

Appendix B: Localizing Oracle Business Intelligence Deployments

Process of Maintaining Translation Tables for Oracle BI	421
Upgrading Oracle Business Intelligence Seed Data for Non-English Locales	422
Externalizing Metadata Strings	423
Adding Custom Translations to the W_LOCALIZED_STRING_G Table	424
About Translating Presentation Services Strings	425
About Installing Oracle Business Intelligence Localization Packs	425
Changing the Default Currency in Analytics Applications	425

Appendix C: Integrating Interactive Dashboards and Operational Applications Data

Importing Oracle's Siebel Industry Applications Seed Data	427
Completing the Initialization in the Siebel Operational Application	428
Updating the Siebel Operational Application	428
Reapplying Customized Style Sheets	429
Process of Configuring Oracle Business Intelligence with Oracle's Siebel Web Extension (SWE)	430
Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services	431
Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE	432
Process of Configuring Oracle BI Action Links	432
Testing Action Links	432
Process of Integrating Analytics and Oracle's Siebel Operational Applications	433
Accessing Optional Analytics Applications	433
Configuring the Browser Settings for Operational Applications	434
Process of Customizing Oracle BI Application Home Page and Dashboards	435
Customizing Oracle BI Content on the Siebel Operational Application Home Page	435
Determining the Oracle BI Report Path Argument	437
Adding Views for Custom Oracle BI Interactive Dashboards	438
Verifying NQHOST Setup for Pharma Disconnected Analytics Client	438
Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client	440
Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client	440
Viewing Information About Preconfigured Dashboards	441

Appendix D: Integrated Security for Oracle BI Applications

Types of Security in Oracle BI	443
Data-Level Security in CRM Analytics Applications	448
Implementing Data-Level Security in the Oracle BI Repository	452
Initialization Blocks and Security-Related Information in Oracle BI	453
Filters for Oracle BI Security Groups	454
Configuring Oracle BI Repository Table Joins for Security	455
Responsibilities and Data-Level Visibility in Sales Analytics Applications	456
Changing the Security Group from Position-Based to Organization-Based in Oracle BI	458
Metadata Object-Level Security in Oracle BI Applications	458
Oracle Partner Analytics Security Settings	460

Consumer Sector Analytics Security Settings	468
Communications, Media, and Energy (CME) Analytics Security Settings	468
Financial Services Analytics Security Settings	469
Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics Security Settings	472
Usage Accelerator Analytics Security Settings	474
Implementing Security With Oracle EBS	476
Background	477
About Unit-Based Security With Oracle EBS	477
About the Relationship Between Object Security and Data Security	482
Initialization Blocks	482

Appendix E: Configuring Metadata for Analytics Applications

What Are Oracle's Siebel Operational Applications?	485
Updating Finance Analytics Logical Table Sources	486
Externalizing Financial Services Metadata Translation Strings	487
Disabling the Finance Analytics Logical Table Sources	488
Developing and Deploying Predictive Scores	489
Analytics Metadata Requirements for Oracle's Siebel Industry Applications	489
Oracle's Siebel Consumer Sector Dashboards and Pages	490
Oracle's Siebel Consumer Sector Data Requirements	492
Oracle's CME family of products (Oracle Communications, Media and Energy Sales Analytics,	492
Oracle's Siebel CME Dashboards and Pages	494
Oracle's Siebel CME Data Requirements	496
Oracle Pharma Sales Analytics Dimensions	496
Dimensions Specific to Subject Areas in Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics	498

Appendix F: Oracle Business Analytics Warehouse for Life Sciences Data Considerations

What's New for Pharma Analytics in Version 7.9	501
Oracle Business Analytics Warehouse for Life Sciences Overview	501
Importing Syndicated Data into Oracle Business Analytics Warehouse	503
Syndicated Loading Definitions	503
Data Types Supported in the Oracle Business Analytics Warehouse	504

Loading Syndicated Data into the Oracle Business Analytics Warehouse	505
Syndicated Data Flat File Formats	510
Life Sciences Data Loading Issues with Oracle Business Analytics Warehouse	535
Incremental Updates in the Oracle Business Analytics Warehouse LS Dimension Tables	536

Appendix G: Pharma Disconnected Analytics Administration - Supplemental Information

About Pharma Disconnected Analytics	539
Sourcing Reports for Oracle's Pharma Disconnected Analytics	541

Index

1

What's New in This Release

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

The *Oracle Business Intelligence Applications Installation and Configuration Guide* contains instructions on installing, configuring, and customizing Oracle Business Intelligence Applications.

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

- On the Oracle Business Intelligence Applications CD-ROM.
- On the Oracle Technology Network at http://www.oracle.com/technology/documentation/bi_apps.html (to register for a free account on the Oracle Technology Network, go to <http://www.oracle.com/technology/about/index.html>).

What's New in Oracle Business Intelligence Applications Installation and Configuration Guide, Version 7.9

This guide is new for Release 7.9.

2

Overview of Oracle Business Intelligence Product Licensing

This chapter provides an overview of the Oracle Business Intelligence product licensing. It includes the following topic:

- [“Oracle Business Intelligence Product Licensing” on page 19.](#)

Oracle Business Intelligence Product Licensing

Oracle Business Intelligence has the following licensed products.

Oracle Business Intelligence Infrastructure

The Oracle Business Intelligence infrastructure is a set of servers, programs, and tools used to build Oracle Business Intelligence applications. Using the Oracle Business Intelligence infrastructure, you can develop your own integrated business intelligence application solution. The components that make up the Oracle Business Intelligence infrastructure are described in [Table 1](#).

Table 1. Oracle Business Intelligence Infrastructure Components

Oracle Business Intelligence Infrastructure Components
Oracle Business Intelligence Server
Oracle Business Intelligence Presentation Services
Oracle Business Intelligence Scheduler
Related component programs:
■ Oracle Business Intelligence Administration Tool
■ Oracle Business Intelligence Catalog Manager
■ Oracle Business Intelligence ODBC Client
■ Oracle Business Intelligence Client Tools

The following documentation is applicable to the Oracle Business Intelligence infrastructure:

- **Oracle Business Intelligence Infrastructure Installation and Configuration Guide.** This guide provides information about the installation and configuration of the Oracle Business Intelligence infrastructure.

- **Oracle Business Intelligence Presentation Services Administration Guide.** This guide provides post-installation configuration and administration procedures for Oracle BI Answers, Oracle BI Delivers, Oracle BI Interactive Dashboards, the Oracle BI Presentation Catalog, and Oracle BI Server.
- **Oracle Business Intelligence Server Administration Guide.** This guide provides information about administration of the Oracle BI Server, Oracle BI repository files, and Oracle BI Disconnected Analytics.

Oracle Business Intelligence Applications

Oracle Business Intelligence Applications are prebuilt business intelligence solutions.

You must install the Oracle Business Intelligence infrastructure before you install Oracle Business Intelligence Applications.

Oracle Business Intelligence applications support Siebel (CRM) and Oracle E-Business Suite applications, such as Oracle's Siebel Sales, Oracle's Siebel Service, and Oracle's Siebel Marketing. If you already own one of the Siebel (CRM) or Oracle E-Business Suite 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 [Table 2](#).

Table 2. Oracle Business Intelligence Applications Components

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

For more information about managing your data using the Oracle Business Intelligence Data Warehouse Administration Console, see *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

3

Overview of Oracle Business Analytics Warehouse

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

- [Oracle Business Analytics Warehouse Overview on page 23](#)
- [Oracle Business Analytics Warehouse Architecture on page 24](#)

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 SAP Applications, Oracle's Siebel, E-Business Suite and Peoplesoft Applications.

The Oracle Business Analytics Warehouse includes the following:

- A data integration engine that combines data from the transactional database to build a data warehouse.
- 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.
- Optional prebuilt data extractors to incorporate data from external applications into the Oracle Business Analytics Warehouse (licensed separately).
- A set of ETL (extract-transform-load) processes that takes data from SAP, Oracle's Siebel, E-Business Suite and Peoplesoft Applications transactional data (OLTP) and creates the Oracle Business Analytics Warehouse tables.

For more information about ETL processing, see *Oracle Business Analytics Warehouse Data Model Reference*.

- 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.

Figure 1 provides an overview of the Oracle Business Analytics Warehouse.

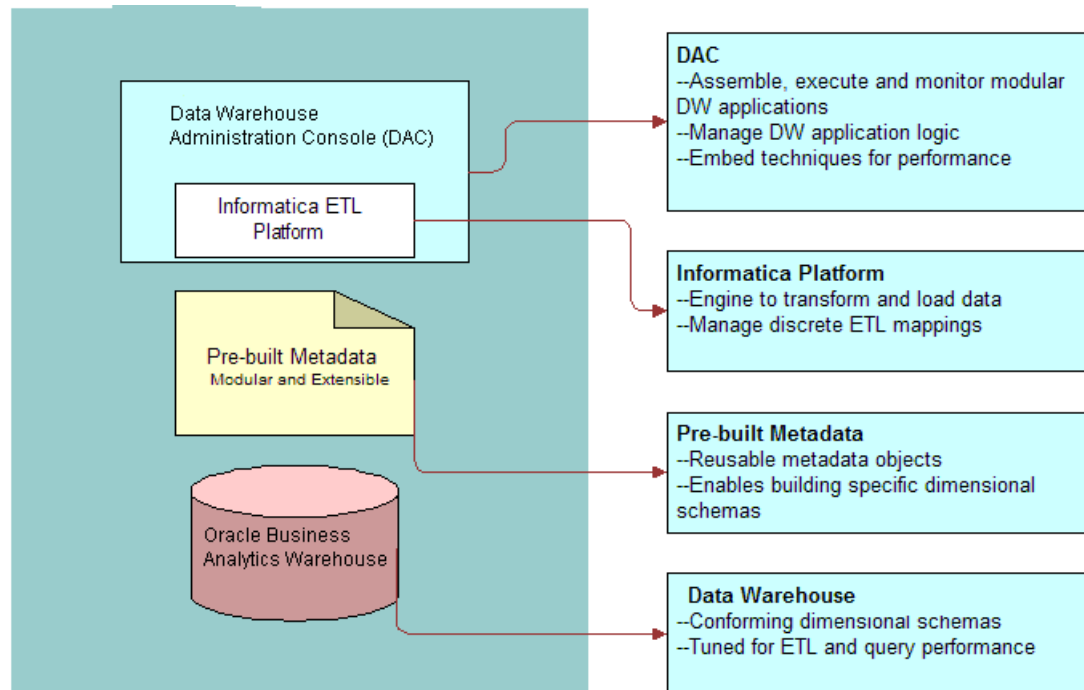


Figure 1. Oracle Business Analytics Warehouse Overview

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 the transactional database into the Oracle Business Analytics Warehouse.

Figure 2 illustrates the Oracle Business Analytics Warehouse architecture.

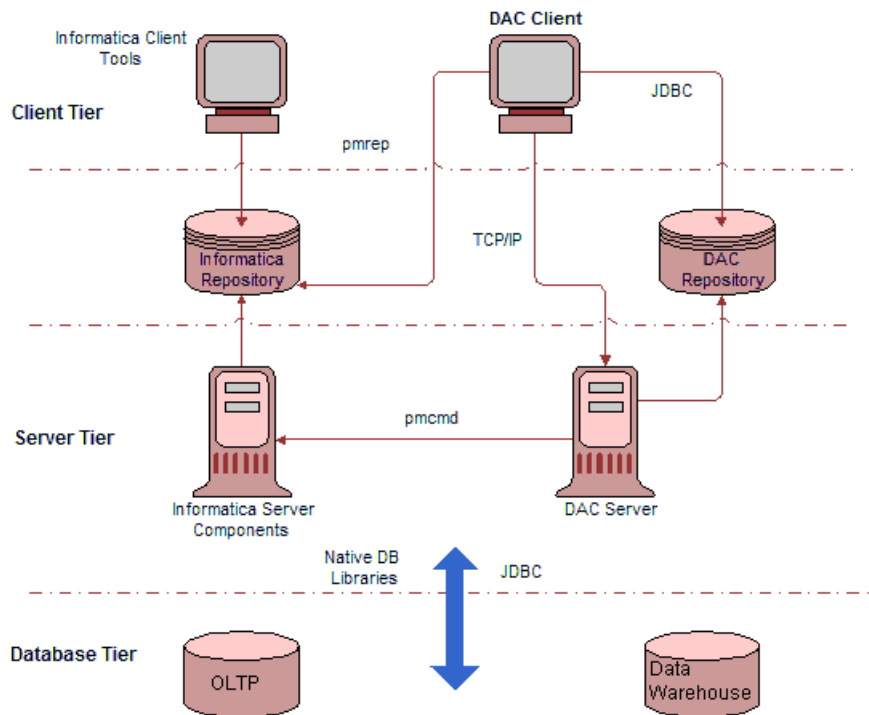


Figure 2. Oracle Business Analytics Warehouse Architecture

Oracle Business Analytics Warehouse Architecture Components

The Oracle Business Analytics Warehouse architecture comprises the following components:

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

- **Informatica Repository.** Stores the metadata related to Informatica workflows.
- **Informatica client utilities.** Tools that allow you to create and manage the Informatica repository.

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

4

Preinstallation and Predeployment Considerations for Oracle BI Applications

This chapter provides information about preparing to install and deploy Oracle Business Intelligence Applications. You should review this information before you begin the installation and deployment process.

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

This chapter contains the following topics:

- [Oracle Business Analytics Warehouse Deployment Configuration on page 28](#)
- [Operating System, Driver, and Connectivity Requirements for Oracle Business Analytics Warehouse on page 29](#)
- [General Guidelines for Setting Up Oracle Business Analytics Warehouse on page 30](#)
- [IBM DB2 UDB-Specific Database Guidelines for Oracle Business Analytics Warehouse on page 32](#)
- [IBM DB2 UDB zOS and OS/390 and Z/os-Specific Database Guidelines for Oracle Business Analytics Warehouse on page 33](#)
- [Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse on page 34](#)
- [Additional Suggestions for Optimizing Oracle Performance in Oracle Business Analytics Warehouse on page 43](#)
- [SQL Server-Specific Database Guidelines for Oracle Business Analytics Warehouse on page 45](#)
- [Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse on page 47](#)
- [Deploying Oracle Business Analytics Warehouse in Latin-1 General, Unicode and Non-English Environments on page 53](#)
- [Miscellaneous Information About Oracle Business Intelligence Applications Deployments on page 64](#)

Oracle Business Analytics Warehouse Deployment Configuration

Figure 3 shows the recommended Oracle Business Analytics Warehouse deployment configuration.

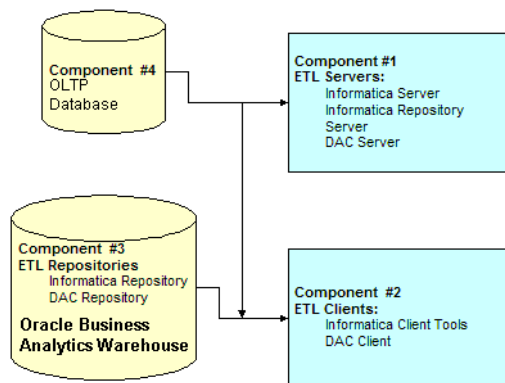


Figure 3. Oracle Business Analytics Warehouse Deployment Configuration

- Component 1 hosts all the ETL servers, that is, the Informatica Server, Informatica Repository Server, and the DAC server.
NOTE: You can install the Informatica Server on other machines as well to increase performance. The other ETL servers can also be hosted on other machines.
- Component 2 hosts all the ETL clients, that is, the Informatica client tools and the DAC client.
- Components 3 and 4 are database instances that can be hosted on one or more machines. The hardware requirements are entirely dependent on your usage and performance requirements. It is highly recommended that each of these components be defined in their own database instance to allow for instantiating parameters that are optimized for each component's usage.

NOTE: For more detailed information about system requirements, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

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

Table 3 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, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

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

Component	Operating System	Software	Connectivity and Driver
1 ETL Servers	■ Solaris ■ AIX ■ HP ■ Windows ■ Linux	■ Informatica Server ■ Informatica Repository Server ■ DAC Server	■ Java SDK ■ JDBC drivers ■ Javamail
2 ETL Clients	■ Windows	■ Informatica client tools ■ DAC client	■ Java SDK ■ JDBC drivers ■ ODBC drivers
3 (A) Oracle Business Analytics Warehouse	■ Solaris ■ AIX ■ HP ■ Windows ■ OS/390, Z/os	Database software with Oracle Business Analytics Warehouse	Not applicable
3 (B) ETL Repositories	■ Solaris ■ AIX ■ HP ■ Windows	Database software with ETL repositories	Not applicable

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 is 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 prejoined, denormalized 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.
- Siebel 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*.

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 bufferpools 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 [“Process of Creating Data Warehouse Tables” on page 105](#), 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 4](#).

Table 4. 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

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

Table 5 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 5. Recommended DB2 Parameter Settings

Parameter	DB2 UDB V7	DB2 UDB V8	Notes
SHEAPTHRES	400000	400000	
ASLHEAPSZ	15	15	
RQRIOBLK	65535	65535	
QUERY_HEAP_SZ	16384	16384	
JAVA_HEAP_SZ	2048	2048	
MAXAGENTS	400	400	
NUM_INITAGENTS	10	10	
NUM_POOLAGENTS	200	200	
INTRA_PARALLEL	YES	YES	
FCM_NUM_BUFFERS	12288	12288	
SHEAPTHRES_SHR	N/A	=SHEAPTHRES	
DBHEAP	16384	16384	
CATALOGCACHE_SZ	5558	5558	
LOGBUFSZ	2048	2048	
UTIL_HEAP_SZ	10000	10000	
NUM_ESTORE_SEGS	16	NIL	Estore 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	

Table 5. Recommended DB2 Parameter Settings

Parameter	DB2 UDB V7	DB2 UDB V8	Notes
STMHEAP	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

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 midtier 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 6](#).

Table 6. 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.

Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse

[Table 7](#) provides parameter guidelines based on the rule-based optimizer for Oracle 8i and the cost-based optimizer for Oracle 9i and Oracle 10g. Use these guidelines as a starting point. You will need to make changes based on your specific database sizes, data shape, server size (CPU and memory), and type of storage. The database administrator should make changes to the settings based on performance monitoring and tuning considerations.

Table 7. Recommended Variable Settings for Oracle Databases

Parameter	Oracle 8i	Oracle 9i	Oracle 10g	Notes
always_anti_join	HASH	N/A	N/A	
cursor_sharing	Exact	Exact	Exact	Specifies the kind of SQL statements can share the same cursors.
cursor_space_for_time	FALSE	FALSE	FALSE	Enabling CURSOR_SPACE_FOR_TIME results in holding pins on cursors and their associated frames/buffers for longer periods of time. Cursor_space_for_time requires at least a 50% increase in the size of the shared pool because of the frames/buffers. If the Stats pack reports show that the waits for library cache latch gets is significant, and the latch gets are due to pin requests, then cursor space for time can be used to improve performance.

Table 7. Recommended Variable Settings for Oracle Databases

Parameter	Oracle 8i	Oracle 9i	Oracle 10g	Notes
db_block_lru_latches	1 or (# of CPUs / 2), whichever is greater	N/A	N/A	Specifies the upper bound of the number of LRU latch sets. Set this parameter to a value equal to the desired number of LRU latch sets. It is advisable to set this parameter if you want to use multiple database writers via the init.ora. There must be at least one latch per DB Writer gatherer process.
db_block_size	32768	32768	32768	
db_cache_size	Use db_block_buffers parameter.	The value depends on physical memory available, as well as other memory consuming processes running on the server.	Use sga_target parameter.	Specifies the size of the DEFAULT buffer pool for buffers with the primary block size. It has a direct impact on query performance. A low value might result in slow query performance, and a high value might deprive other server processes of the memory. For example, the DB_CACHE_SIZE value for an Oracle Analytics Warehouse Database running alone on an 8Gb RAM server can be set to 2Gb.
db_file_direct_io_count	64	N/A	N/A	
db_file_multiblock_read_count	32	32	32	Specifies the maximum number of blocks read in one I/O operation during a sequential scan. The optimizer is more likely to choose a full table scan over an index if the value of this parameter is high. The maximum value is always less than the operating system's maximum I/O size expressed as Oracle blocks ((max I/O size)/ DB_BLOCK_SIZE). If you set this parameter to a value greater than the maximum, Oracle uses the maximum.

Table 7. Recommended Variable Settings for Oracle Databases

Parameter	Oracle 8i	Oracle 9i	Oracle 10g	Notes
db_files	512	512	512	Specifies the maximum number of data files that can be opened for a database.
db_writer_processes	1 or (# of CPUs / 4), whichever is greater.	1 or (# of CPUs / 4), whichever is greater.	1 or (# of CPUs / 4), whichever is greater.	Specifies the initial number of database writer processes for an instance. It cannot be greater than DB_BLOCK_LRU_LATCHES parameter. DB_WRITER_PROCESSES is useful for systems that modify data heavily.
Default pctincrease	0	0	0	This parameter is specified at tablespace / object level, not in init.ora.
dml_locks	1000	1000	1000	
hash_area_size	2 * sort_area_size	Use pga_aggregate_target parameter.	Use pga_aggregate_target parameter.	Specifies the upper limit for a single hash operation. Setting WORKAREA_SIZE_POLICY = AUTO and defining PGA_AGGREGATE_TARGET takes care of HASH and SORT areas. NOTE: PGA_AGGREGATE_TARGET does not work for MTS / Shared server configuration in 9i.
hash_join_enabled	TRUE	TRUE	N/A	Specifies whether the optimizer should consider using a hash join as a join method.

Table 7. Recommended Variable Settings for Oracle Databases

Parameter	Oracle 8i	Oracle 9i	Oracle 10g	Notes
hash_multiblock_io_count	The value range is between 1 and 65,536 / DB_BLOCK_SIZE.	N/A	N/A	Specifies the number of sequential blocks that a hash join reads and writes in one IO. The maximum value for HASH_MULTIBLOCK_IO_COUNT varies by operating system. It is always less than the operating system's maximum I/O size expressed as Oracle blocks (max_IO_size/DB_BLOCK_SIZE). Make sure that you are not using MTS. For Oracle 8i you need not set or change the value of this parameter, because Oracle computes the value individually for every query.
Initial extent	20 Mb	20 Mb	20 Mb	This parameter is specified at tablespace / object level, not in init.ora
java_pool_size	20M	20M	Covered by SGA_TARGET	Specifies the size (in bytes) of the Java pool from which the Java memory manager allocates most Java state during runtime execution.
large_pool_size	16M	16M	Covered by SGA_TARGET	
log_buffer	Default is 128 * Number of CPUs.	Default is 128 * Number of CPUs.	Default is 128 * Number of CPUs.	Specifies the amount of memory (in bytes) that Oracle uses when buffering redo entries to a redo log file. Larger values for LOG_BUFFER reduce redo log file I/O, particularly if transactions are long or numerous. In a busy system, a value 65536 or higher is reasonable.
log_checkpoint_interval		100000	100000	
log_checkpoint_timeout		1800	1800	

Table 7. Recommended Variable Settings for Oracle Databases

Parameter	Oracle 8i	Oracle 9i	Oracle 10g	Notes
Next Extent	20 Mb	20 Mb	20 Mb	This parameter is specified at tablespace / object level, not in init.ora.
open_cursors	1000	1000	1000	Specifies the maximum number of open cursors (handles to private SQL areas) that a session can have at once.
optimizer_index_caching	unset	Refer to "Notes" column.	Refer to "Notes" column.	Specifies the behavior of cost-based optimization to favor nested loops joins and IN-list iterators. For 8i: use Unset as the value for this parameter to avoid nested loop index joins (as favored by the optimizer). For 9i and 10g: DBA can modify the optimizer's assumptions about index caching for nested loops joins and IN-list iterators by setting this parameter to a value between 0 and 100 to indicate the percentage of the index blocks the optimizer should assume are in the cache. Setting this parameter to a higher value makes nested loops joins and IN-list iterators look less expensive to the optimizer.

Table 7. Recommended Variable Settings for Oracle Databases

Parameter	Oracle 8i	Oracle 9i	Oracle 10g	Notes
optimizer_index_cost_adj	Choose appropriate value based on data shape and performance monitoring.	Choose appropriate value based on data shape and performance monitoring.	Choose appropriate value based on data shape and performance monitoring.	Specifies how you tune the performance of a system in cases in which the optimizer may be choosing too few or too many index access paths. For example, setting it to 50 percent will make the index access path look half as expensive as normal. The legal range of values for this parameter is 1 to 10000 percent. The default for this parameter is 100 percent, which makes the optimizer cost index access paths at the regular cost. See also Additional Suggestions for Optimizing Oracle Performance in Oracle Business Analytics Warehouse on page 43 .
optimizer_max_permutations				Specifies the number of permutations of the tables that the optimizer will consider in queries with joins. Such a restriction ensures that the parse time for the query stays within acceptable limits. The default value is: 2000 if Parameter OPTIMIZER_FEATURES_ENABLE >= 9.0.0 and 80000 if Parameter OPTIMIZER_FEATURES_ENABLE < 9.0.0
optimizer_mode	ALL_ROWS	CHOOSE	CHOOSE	

Table 7. Recommended Variable Settings for Oracle Databases

Parameter	Oracle 8i	Oracle 9i	Oracle 10g	Notes
pga_aggregate_target	N/A	The value depends on physical memory available, as well as other memory consuming processes running on the server.	The value depends on physical memory available, as well as other memory consuming processes running on the server.	Specifies the amount of target aggregate PGA memory available to all server processes attached to the instance. Setting WORKAREA_SIZE_POLICY = AUTO and defining PGA_AGGREGATE_TARGET takes care of HASH and SORT areas. NOTE: PGA_AGGREGATE_TARGET does not work for MTS / Shared server configuration in 9i. For example, PGA_AGGREGATE_TARGET value for an Oracle Analytics Warehouse Database running alone on an 8Gb RAM server can be set to 2Gb.
processes	510	510	510	
query_rewrite_enabled		TRUE	TRUE	
query_rewrite_integrity	USE_STALE	TRUSTED	TRUSTED	
rollback_segments				See also undo_retention.
session_cached_cursors	500	500	500	
sga_max_size	N/A	The value cannot be less than the sum of all SGA components (pools, buffer and *_cache_size).	The value cannot be less than the sum of SGA_TARGET and all manual pools and buffers.	Specifies the maximum size of SGA for the lifetime of the instance.

Table 7. Recommended Variable Settings for Oracle Databases

Parameter	Oracle 8i	Oracle 9i	Oracle 10g	Notes
sga_target	N/A	N/A	The value depends on physical memory available, as well as other memory consuming processes running on the server.	Specifies the size following SGA components: DB_CACHE_SIZE, SHARED_POOL_SIZE, LARGE_POOL_SIZE and JAVA_POOL_SIZE. The STATISTICS_LEVEL should be set to TYPICAL or ALL. Applicable to Oracle 10g only.
shared_pool_reserved_size	10% of shared_pool_size	10% of shared_pool_size	10% of shared_pool_size	Specifies (in bytes) the shared pool space that is reserved for large contiguous requests for shared pool memory. The value cannot exceed 50% of shared_pool_size. If the value is not set, then its default is 5%.
shared_pool_size	500Mb	500Mb	Covered by SGA_TARGET	Specifies (in bytes) the size of the shared pool. The shared pool contains shared cursors, stored procedures, control structures, and other structures.
sort_area_retained_size	This value is instance specific. DBA has to define its value basing on the volume of concurrent sort operations and available physical RAM.	Use pga_aggregate_target parameter	Use pga_aggregate_target parameter	Specifies the amount of memory which that is allocated to store sorted data after sort operation is over. The data that does not fit into SORT_AREA_RETAINED_SIZE memory would be written to a temporary tablespace. This memory is released back to the UGA (not to the operating system) after the last row is fetched from the sort space.

Table 7. Recommended Variable Settings for Oracle Databases

Parameter	Oracle 8i	Oracle 9i	Oracle 10g	Notes
sort_area_size	This value is instance specific. DBA has to define its value basing on the volume of concurrent sort operations and available physical RAM.	Use pga_aggregate_target parameter	Use pga_aggregate_target parameter	Specifies the upper limit for a single sort operation. In a database with many concurrent sort operations the server could use as much as number of concurrent sorts * sort_area_size bytes of RAM. Depending on the database load, specifically sort operations, you need to determine the parameter value that will not consume all physical memory at the peak loads and at the same time allocate reasonable memory for performing sort operations. Setting WORKAREA_SIZE_POLICY = AUTO and defining PGA_AGGREGATE_TARGET takes care of HASH and SORT areas. NOTE: PGA_AGGREGATE_TARGET does not work for MTS / Shared server configuration in 9i.
sort_multiblock_read_count	The value is instance specific. Suggested value is 2.	N/A	N/A	Specifies the number of database blocks to read each time a sort performs a read from a temporary segment. Any increase in I/O throughput obtained by increasing SORT_MULTIBLOCK_READ_COUNT needs to be balanced with a possible increase in total amount of I/O performed due to an increase in the number of merge passes.
Star_transformation_enabled	TRUE	TRUE	TRUE	
Statistics_level		Typical	Typical	

Table 7. Recommended Variable Settings for Oracle Databases

Parameter	Oracle 8i	Oracle 9i	Oracle 10g	Notes
timed_statistics	TRUE	TRUE	TRUE	On most platforms, enabling timed statistics has minimal effect on performance. It can be enabled/disabled dynamically at both the system and session level.
Undo_management	N/A	Auto	Auto	
Undo_retention	N/A	The suggested value range is from 30000 to 90000.	The suggested value range is from 30000 to 90000.	Specifies (in seconds) the amount of committed undo information to retain in the database. To calculate the value: 1. Identify the longest query that will run in the instance and set the undo_retention to that query execution time. 2. Set autoextend for UNDO tablespace. 3. After the initial load is complete, you might want to reduce UNDO_RETENTION to a smaller value.
Undo_tablespace	N/A	Undo	Undo	
Workarea_size_policy	N/A	Auto	Auto	

Additional Suggestions for Optimizing Oracle Performance in Oracle Business Analytics Warehouse

- 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.
 - In addition, you may also configure the Oracle star-join transformation. This requires non-enforced foreign keys in Oracle and the creation of necessary bitmap indices. This task is optional. It may not be necessary, as ongoing tuning may reach the desired performance goals.
- 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))
```

SQL Server-Specific Database Guidelines for Oracle Business Analytics Warehouse

This section provides guidelines for SQL Server database usage.

NOTE: SQL Server databases with binary sort order or case-sensitive dictionary sort order are supported. Case-insensitive dictionary sort order is not supported.

This section includes the following topics:

- [Setting the ANSI NULL Option on page 45](#)
- [Modifying the DB Library Options Setting on page 45](#)
- [Recommended SQL Server Database Parameters on page 46](#)

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.

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.

Recommended SQL Server Database Parameters

If necessary, SQL Server components can be configured to optimize performance, as shown in [Table 8](#).

Table 8. 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	
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 multithreaded 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.

Table 8. Recommended Variable Settings for SQL Server Databases

Parameter	Recommended Setting	Notes
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.

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:

- [General Guidelines for Teradata Deployments on page 47](#)
- [Best Practices For Teradata Deployments on page 48](#)

General Guidelines for Teradata Deployments

The following requirements apply to Teradata database usage:

- Install the FastLoad and TPump Teradata utilities on the machine on which the Informatica Server is installed.
- When installing Informatica PowerCenter, make sure there are no spaces in the Informatica Server directory name or the directory path. The default directory contains spaces that you must remove manually.
- When you install Oracle BI Applications in a Teradata environment, you must load the pre-built Informatica repository file \dwrep\Oracle_BI_DW_Teradata.rep. For more information, see [Process of Loading the Pre-Built Repository Into Informatica on page 115](#).
- 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.
- In DAC ETL preferences update the values of Teradata required hints. Please refer to the DAC ETL preference section.
- The following static parameters are applicable only when the target data warehouse database is Teradata. To avoid overflows if the arithmetic calculation involves any decimal datatype more than 18,3 (precision, scale), add the following static parameters:
 - "as decimal(18,3))" for \$\$Hint_Tera_Post_Cast
 - "Cast (" for \$\$Hint_Tera_Pre_Cast

Best Practices For Teradata Deployments

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

- [LEFT OUTER JOIN Issue on page 48](#)
- [Database Statistics on page 50](#)
- [Group By Vs Distinct on page 50](#)
- [Pruning Of Mapping on page 50](#)
- [Loader Configurations on page 51](#)

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

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_ID 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.

Database Statistics

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.

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.

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.

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.

Loader Configurations

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

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

- Fastload
- Mload
- T pump

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

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

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

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

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

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

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

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

Fastload

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

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

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

If a table is locked (for example, (for W_ORG_DS), use the following script to release the lock:

```
LOGON SDCNCR1/Siebel_qa1,sqa1;
BEGIN LOADING Siebel_qa1.W_ORG_DS
ERRORFILES Siebel_qa1.ET_W_ORG_DS,siebel_qa1.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.

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.

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 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.

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

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

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

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

- **Latin-1 General (7-bit ASCII) to Latin-1 General (7-bit ASCII).** 7-bit ASCII is the character set used for English. See [Source and Data Warehouse Code Pages for Latin-1 General \(7-bit ASCII\) to Latin-1 General \(7-bit ASCII\) on page 54](#).
- **Latin-1 General (8-bit ASCII) to Latin-1 General (8-bit ASCII).** 8-bit ASCII is the character set used for Western European languages that use accented Roman characters. See [Source and Data Warehouse Code Pages for Latin-1 General \(8-bit ASCII\) to Latin-1 General \(8-bit ASCII\) on page 56](#).
NOTE: You must run the Informatica Server in Unicode mode if your source data contains multibyte or ISO 8859-1 (8-bit ASCII) data. For instructions on setting the data movement mode to Unicode, see [To set up the Informatica Server in Informatica PowerCenter on page 79](#).
- **Unicode to Unicode.** See [Source and Data Warehouse Code Pages for Latin-1 General \(8-bit ASCII\) to Latin-1 General \(8-bit ASCII\) on page 56](#).
- **Code page (multi- or single-byte) to Unicode.** See [Source and Data Warehouse Code Pages for Code Page to Unicode on page 59](#).
- **Code page to code page (where the code pages are the same).** See [Source and Data Warehouse Code Pages for Code Page to Code Page on page 61](#).

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

- **NLS_LANG (for Oracle).** For instructions, see [Setting the NLS_LANG Environment Variable for Oracle Databases on page 63](#).
- **DB2CODEPAGE (for DB2).** For instructions, see [Setting the DB2CODEPAGE Environment Variable for DB2 Databases on page 64](#).

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

- **Informatica Server Data Movement.** For instructions, see [To set up the Informatica Server in Informatica PowerCenter on page 79](#).

Code Page Overview

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

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

Choosing Code Pages

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

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

Code Page Compatibility

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

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

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

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

Informatica Server and Repository Server Running on Windows with OS ENU

Table 9 provides the code pages for the Informatica Server and Repository Server running on Windows with OS ENU.

Table 9. Code Pages for Informatica Server and Repository Server on Windows with OS ENU

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

Informatica Server and Repository Server Running on UNIX with OS ENU

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

- **NLS_LANG (for Oracle).** For instructions, see [Setting the NLS_LANG Environment Variable for Oracle Databases on page 63](#).
- **DB2CODEPAGE (for DB2).** For instructions, see [Setting the DB2CODEPAGE Environment Variable for DB2 Databases on page 64](#).

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

- **Informatica Server Data Movement.** For instructions, see [To set up the Informatica Server in Informatica PowerCenter on page 79](#).

Table 10 provides the code pages for the Informatica Server and Repository Server running on UNIX with OS ENU.

Table 10. Code Pages for Informatica Server and Repository Server on UNIX with OS ENU

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

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

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

- **NLS_LANG (for Oracle).** For instructions, see [Setting the NLS_LANG Environment Variable for Oracle Databases on page 63](#).
- **DB2CODEPAGE (for DB2).** For instructions, see [Setting the DB2CODEPAGE Environment Variable for DB2 Databases on page 64](#).

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

- **Informatica Server Data Movement.** For instructions, see [To set up the Informatica Server in Informatica PowerCenter on page 79](#).

Table 11 provides the code pages for the Informatica Server running on UNIX, and the Repository Server on Windows, both with OS ENU.

Table 11. Code Pages for Informatica Server on UNIX, Repository Server on Windows with OS ENU

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

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

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

NOTE: You must run the Informatica Server in UNICODE mode if your source data contains multibyte or ISO 8859-1 (8-bit ASCII) data. For instructions on setting the data movement mode to UNICODE, see [To set up the Informatica Server in Informatica PowerCenter on page 79](#).

Informatica Server and Repository Server Running on Windows with OS ENU

Table 12 provides the code pages for the Informatica Server and Repository Server running on Windows with OS ENU.

Table 12. Code Pages for Informatica Server and Repository Server on Windows with OS ENU

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

Table 12. Code Pages for Informatica Server and Repository Server on Windows with OS ENU

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

Informatica Server and Repository Server Running on UNIX with OS ENU

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

- **NLS_LANG (for Oracle).** For instructions, see [Setting the NLS_LANG Environment Variable for Oracle Databases on page 63](#).
- **DB2CODEPAGE (for DB2).** For instructions, see [Setting the DB2CODEPAGE Environment Variable for DB2 Databases on page 64](#).

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

- **Informatica Server Data Movement.** For instructions, see [To set up the Informatica Server in Informatica PowerCenter on page 79](#).

Table 13 provides the code pages for the Informatica Server and Repository Server running on UNIX with OS ENU.

Table 13. Code Pages for Informatica Server and Repository Server on UNIX with OS ENU

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

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

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

- **NLS_LANG (for Oracle).** For instructions, see [Setting the NLS_LANG Environment Variable for Oracle Databases on page 63](#).
- **DB2CODEPAGE (for DB2).** For instructions, see [Setting the DB2CODEPAGE Environment Variable for DB2 Databases on page 64](#).

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

- **Informatica Server Data Movement.** For instructions, see [To set up the Informatica Server in Informatica PowerCenter on page 79](#).

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

Table 14. Code Pages for Informatica Server on UNIX, Repository Server on Windows with OS ENU

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

Source and Data Warehouse Code Pages for Unicode to Unicode

For a list of the code pages that are supported for source and data warehouse databases, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

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

- **NLS_LANG (for Oracle).** For instructions, see [Setting the NLS_LANG Environment Variable for Oracle Databases on page 63](#).
- **DB2CODEPAGE (for DB2).** For instructions, see [Setting the DB2CODEPAGE Environment Variable for DB2 Databases on page 64](#).

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

- **Informatica Server Data Movement.** For instructions, see [To set up the Informatica Server in Informatica PowerCenter on page 79](#).
- **SiebelUnicodeDB.** For instructions on Windows, see [Setting the Siebel UnicodeDB Environment Variable on Windows on page 87](#). For instructions on UNIX, see [To set Siebel UnicodeDB on the Informatica Server if source and target databases are Unicode on UNIX on page 163](#).

Informatica Server and Repository Server Running on Windows with OS ENU

Table 15 provides the code pages for the Informatica Server and Repository Server running on Windows with OS ENU.

Table 15. Code Pages for Informatica Server and Repository Server on Windows with OS ENU

Component Code Page	Code Page
Source code page	UTF-8 encoding of Unicode
Data warehouse code page	UTF-8 encoding of Unicode

Table 15. Code Pages for Informatica Server and Repository Server on Windows with OS ENU

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

Informatica Server and Repository Server Running on UNIX with OS ENU

Table 16 provides the code pages for the Informatica Server and Repository Server running on UNIX with OS ENU.

Table 16. Code Pages for Informatica Server and Repository Server on UNIX with OS ENU

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

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

Table 17 provides the code pages for the Informatica Server running on UNIX, the Repository Server on Windows with OS ENU.

Table 17. Code Pages for Informatica Server on UNIX, Repository Server on Windows with OS ENU

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

Source and Data Warehouse Code Pages for Code Page to Unicode

For a list of the code pages that are supported for source and data warehouse databases, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

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

- NLS_LANG (for Oracle). For instructions, see [To set the NLS_LANG environment variable for Oracle databases on page 63](#).
- DB2CODEPAGE (for DB2). For instructions, see [To set the DB2CODEPAGE environment variable for DB2 databases on page 64](#).

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

- Informatica Server Data Movement. For instructions, see [To set up the Informatica Server in Informatica PowerCenter on page 79](#).
- SiebelUnicodeDB. For instructions on Windows, see [Setting the Siebel UnicodeDB Environment Variable on Windows on page 87](#). For instructions on UNIX, see [To set Siebel UnicodeDB on the Informatica Server if source and target databases are Unicode on UNIX on page 163](#).
- If the Informatica Server is running on Unix, then you also need to set the PMREPCODEPAGE environment variable appropriately. For example: PMREPCODEPAGE=MS932.

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

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

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

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

Table 18. Informatica Server and Repository Server on Windows with OS <LANG>

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

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

Table 19 provides the code pages for the Informatica Server and Repository Server running on UNIX with OS <LANG>.

Table 19. Code Pages for Informatica Server and Repository Server on UNIX with OS <LANG>

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

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

Table 20 provides the code pages for the Informatica Server running on UNIX and the Repository Server running on Windows, both with OS <LANG>.

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

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

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

For a list of the code pages that are supported for source and data warehouse databases, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

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

- NLS_LANG (for Oracle). For instructions, see [To set the NLS_LANG environment variable for Oracle databases on page 63](#).
- DB2CODEPAGE (for DB2). For instructions, see [To set the DB2CODEPAGE environment variable for DB2 databases on page 64](#).

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

- Informatica Server Data Movement. For instructions, see [To set up the Informatica Server in Informatica PowerCenter on page 79](#).
- SiebelUnicodeDB. For instructions on Windows, see [Setting the Siebel UnicodeDB Environment Variable on Windows on page 87](#). For instructions on UNIX, see [To set Siebel UnicodeDB on the Informatica Server if source and target databases are Unicode on UNIX on page 163](#).
- If the Informatica Server is running on UNIX, then you also need to set the PMREPCODEPAGE environment variable appropriately. For example: PMREPCODEPAGE=MS932.

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

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

[Table 21](#) provides the code pages for the Informatica Server and the Repository Server running on Windows, both with OS <LANG>.

Table 21. Code Pages for Informatica Server and Repository Server on Windows, OS <LANG>

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

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

[Table 22](#) provides the code pages for the Informatica Server and the Repository Server running on UNIX, both with OS <LANG>.

Table 22. Code Pages for Informatica Server and Repository Server on Windows, OS <LANG>

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

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

Table 23 provides the code pages for the Informatica Server running on UNIX and the Repository Server running on Windows, both with OS <LANG>.

Table 23. Code Pages for Informatica Server on UNIX, Repository Server on Windows, OS <LANG>

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

Setting the NLS_LANG Environment Variable for Oracle Databases

Follow this procedure to set the NLS_LANG environment variable for Oracle databases.

NOTE: You need to set the NLS_LANG environment variable on the machine that runs the Informatica server, and on the machine that runs the Oracle client.

To set the NLS_LANG environment variable for Oracle databases

- Determine the NLS_LANG value.
 - In the data warehouse database, run the command

```
SELECT * FROM V$NLS_PARAMETERS
```
 - Make a note of the NLS_LANG value, which is in the format [NLS_LANGUAGE]_[NLS_TERRITORY].[NLS_CHARACTERSET].
For example: American_America.UTF8
- For Windows:
 - Navigate to Control Panel > System and click the Advanced tab. Click Environment Variables.
 - In System variables section, click New.
 - In the Variable Name field, enter NLS_LANG.
 - 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.
- For UNIX, set the variable as shown below:

```
setenv NLS_LANG <NLS_LANG>
```

For example: `setenv NLS_LANG American_America.UTF8.`

If your data is 7-bit or 8-bit ASCII and the Informatica Server is running on UNIX, then set

```
NLS_LANG <NLS_LANGUAGE>_<NLS_TERRITORY>.WE8ISO8859P1
```

CAUTION: Make sure you set the NLS_LANG variable correctly, as stated in this procedure, or your data will not display correctly.

- 4 Reboot the machine after creating the variable.

Setting the DB2CODEPAGE Environment Variable for DB2 Databases

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

To set the DB2CODEPAGE environment variable for DB2 databases

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

```
SELECT CODEPAGE FROM SYSCAT.DATATYPES WHERE TYPENAME = 'VARCHAR'
```
 - b Make a note of the result.
For example: 1208
- 2 For Windows:
 - a Navigate to Control Panel > System and click the Advanced tab. Click Environment Variables.
 - b In System variables section, click New.
 - c In the Variable Name field, DB2CODEPAGE.
 - d In the Variable Value field, enter the value that was returned in [Step 1](#).
- 3 For UNIX, set the variable as shown below:

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

For example: `setenv 1208.`
- 4 Reboot the machine after creating the variable.

Miscellaneous Information About Oracle Business Intelligence Applications Deployments

This section contains the following topics:

- [Preconfigured Mapping for the SA System Subject Area on page 65](#)
- [Using Initialization Blocks on page 66](#)
- [Performance Improvements for Change Capture for Incremental ETL Run on page 67](#)

Preconfigured Mapping for the SA System Subject Area

For Oracle's Siebel Applications customers, [Table 24 on page 65](#) 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 24. 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

Table 24. Preconfigured Mappings for the User Table in the SA System Subject Area

Logical Column	Physical Table	Expression	Comments
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	

Using Initialization Blocks

Table 25 on page 66 lists some of the initialization blocks common to all Oracle Business Intelligence applications and their purposes. Initialization blocks that are specific to each Oracle BI applications area are not listed here.

Table 25. 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 25. 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.
Orgs 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.

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.

Implementing Transactional Database (OLTP) Indexes

To implement indexes, you use SQL files that are available in the \OracleBI\dwrep folder. The table below describes the SQL files that are appropriate for the specified applications.

Table 26. SQL files for 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.

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
)
```

Based on the SQL above, the following indexes are created on the S_CONTACT table by the SQL generation scripts:

Table 27.

Index	Index Column
S_CONTACT_W1	LAST_UPD, ROW_ID_MODIFICATION_NUM
S_CONTACT_W11	LAST_UPD

5

Installing and Configuring Oracle BI Applications on Windows

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

- [Installing the Java SDK on page 71](#)
- [Running the Oracle Business Intelligence Applications Installation Wizard \(Windows\) on page 72](#)
- [Running the Informatica PowerCenter Installation Wizard \(Windows\) on page 75](#)
- [Installing the Informatica Patch for Teradata on page 88](#)
- [Setting Informatica Data Code Pages Validation on page 88](#)
- [Process of Copying Hibernate Libraries on page 88](#)
- [Configuring the DAC Client on page 89](#)
- [Creating the Required Databases on page 90](#)
- [Creating the DAC Metadata Repository on page 91](#)
- [Configuring the DAC Metadata Repository on page 95](#)
- [Configuring the Oracle Business Analytics Warehouse Database on page 104](#)
- [Registering Informatica Servers in the DAC Client on page 110](#)
- [Process of Configuring and Managing the Informatica Repository on page 113](#)
- [Setting the Code Page File Property for Unicode Environments on Teradata Databases on page 124](#)
- [Setting Up the HOSTS File for Teradata Installations on Windows on page 124](#)
- [Verifying the Informatica Repository Configuration on page 125](#)
- [Setting Up The Time Dimension on page 126](#)
- [Configuring the DAC Server on page 129](#)
- [Starting the Informatica Server on page 138](#)
- [Updating Siebel Transactional Database Schema Definitions on page 138](#)
- [Creating Stored Procedures for DB2-UDB on page 142](#)
- [Teradata-Specific Installation and Configuration Checklist \(Windows\) on page 143](#)

Installation and Configuration Process Task List for Windows

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

- 1** [Installing the Java SDK on page 71](#) (if not already installed as part of the Oracle BI infrastructure installation).
- 2** [Running the Oracle Business Intelligence Applications Installation Wizard \(Windows\) on page 72.](#)
- 3** [Running the Informatica PowerCenter Installation Wizard \(Windows\) on page 75.](#)
- 4** [Setting Up the Informatica Server and Informatica Repository Server on page 79](#)
 - a** [“Process Of Setting Up The Informatica Server” on page 79](#)
 - b** [“Process Of Setting Up The Informatica Repository Server” on page 83](#)
 - c** [“Process of Starting the Informatica Services” on page 84](#)
- 5** [Setting Environment Variables on Windows on page 86](#)
- 6** [Installing the Informatica Patch for Teradata on page 88.](#) (Teradata only)
- 7** [Setting Informatica Data Code Pages Validation on page 88.](#)
- 8** [Process of Copying Hibernate Libraries on page 88](#)
- 9** [Configuring the DAC Client on page 89.](#)
 - a** [Process of Configuring the DAC Config.bat File on page 89.](#)
 - b** [Installing JDBC Drivers for DAC Database Connectivity on page 90.](#)
- 10** [Creating the Required Databases on page 90](#)
- 11** [Creating the DAC Metadata Repository on page 91.](#)
- 12** [Configuring the DAC Metadata Repository on page 95.](#)
 - a** [Setting Up DAC System Properties on page 96.](#)
 - b** [Specifying Transactional and Data Warehouse Data Sources in the DAC on page 100](#)
 - c** [Configuring Email Recipients in the DAC Client on page 103](#)
 - d** [Configuring Email Recipients in the DAC Client on page 103.](#)
 - e** [Activating Join Indices for Teradata Databases on page 104.](#)
- 13** [Configuring the Oracle Business Analytics Warehouse Database on page 104.](#)
 - a** [Process of Creating the SSE Role on page 105.](#)
 - b** [Process of Creating Data Warehouse Tables on page 105.](#)
 - c** [Process of Creating Data Warehouse Tables on a Teradata Database on page 109](#)
- 14** [Registering Informatica Servers in the DAC Client on page 110.](#)
- 15** [Process of Configuring and Managing the Informatica Repository on page 113.](#)
 - a** [Process of Registering the Informatica Repository Server on page 114](#)
 - b** [Process of Loading the Pre-Built Repository Into Informatica on page 115.](#)
 - c** [Process of Configuring the Informatica Repository in Workflow Manager on page 118](#)
 - d** [Process of Stopping and Starting the Informatica Repository \(Optional\) on page 121.](#)

- e Process of Backing Up and Restoring the Informatica Repository (Optional) on page 122.
- 16 Setting the Code Page File Property for Unicode Environments on Teradata Databases on page 124
- 17 Setting Up the HOSTS File for Teradata Installations on Windows on page 124.
- 18 Verifying the Informatica Repository Configuration on page 125.
- 19 Starting the Informatica Server on page 138.
- NOTE:** You must perform the following steps for each source database type.
- 20 Setting Up The Time Dimension on page 126
- 21 Configuring the DAC Server on page 129.
 - a Process of Configuring the Connection Between the DAC Server and DAC Repository on page 129
 - b Process of Configuring Email in the DAC Server (Optional) on page 130
 - c Process of Setting DAC Source System Parameters on page 131
 - d Process of Starting the DAC Server on page 137
- 22 Updating Siebel Transactional Database Schema Definitions on page 138.
 - a Process of Applying Siebel (CRM) and Oracle E-Business Suite Schema Changes For Oracle, DB2/ UDB, MSSQL on page 139.
 - b Process of Applying CRM Schema Changes For DB2 on 390 or z/os on page 140
 - c Process of Verifying the Siebel (CRM) and Oracle E-Business Suite Schema Changes on page 141.
 - d About Delete Triggers on page 141.

Installing the Java SDK

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

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

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

To install Java SDK

- 1 Download the Java SDK from the Sun Web site, and install it on the machine where both the DAC server and DAC client are installed.
 - a For example, (on Windows): <DRIVE>:\j2sdk1.5

NOTE: Make sure that there are no spaces in the directory path.

Running the Oracle Business Intelligence Applications Installation Wizard (Windows)

NOTE: Before you install Oracle Business Intelligence Applications on a machine, you must first install the Oracle Business Intelligence infrastructure on that machine. For instructions on installing the Oracle Business Intelligence infrastructure, see *Oracle Business Intelligence Infrastructure Installation and Configuration Guide*.

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

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

To install Oracle BI Applications on Windows

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



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

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

Oracle Business Intelligence Applications (7.9.0.0)

Please select the installation location for the installed Oracle BI 10.1.3.2 (Browse or Enter):

C:\OracleBI

Browse

Please select the data location for the installed Oracle BI 10.1.3.2 (Browse or Enter):

C:\OracleBIData

Browse

Java SDK 1.5 and greater is required. Please specify Java SDK installation location (JAVA_HOME) below.

C:\jdk1.5.0_06

Browse

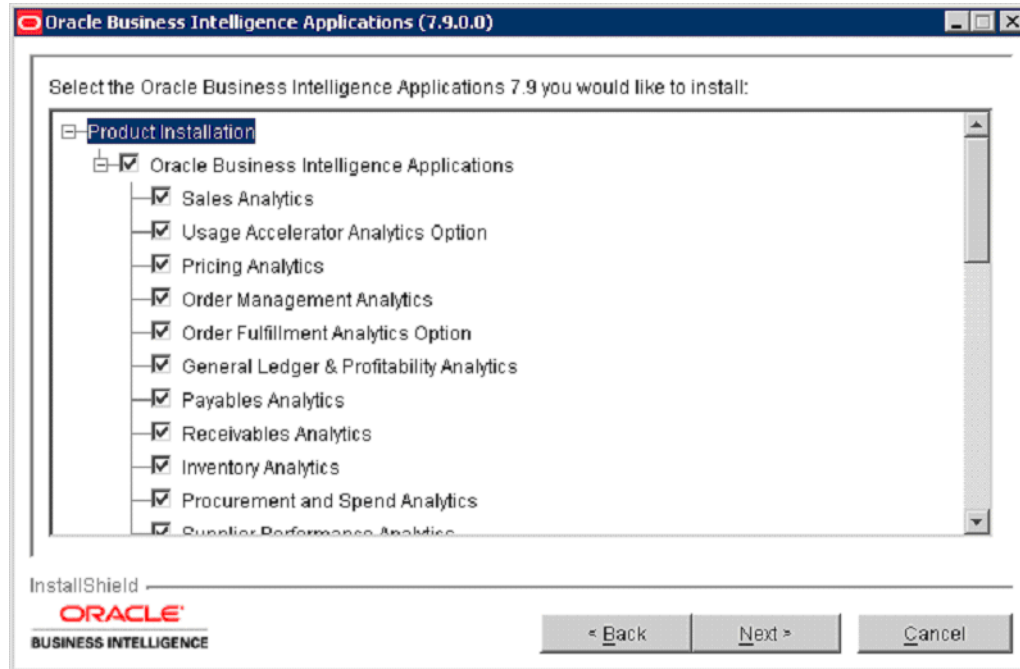
InstallShield

ORACLE
BUSINESS INTELLIGENCE

< Back Next > Cancel

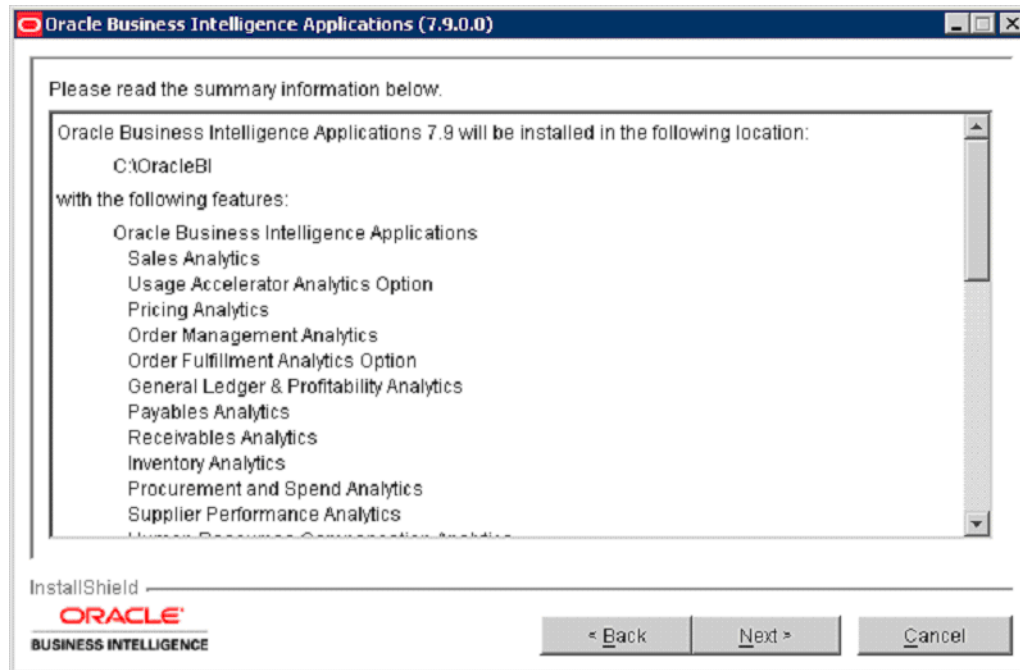
- 3 Enter or browse for the location for the Oracle BI infrastructure (for example, <DRIVE>:\OracleBI\), the Oracle BI infrastructure data (for example, <DRIVE>:\OracleBIData\), and the Java SDK (for example, <DRIVE>:\jdk1.5).
- 4 Click Next to display the Source Systems Selection page.

- 5 At the Source Systems Selection page, select the Business Applications (or containers) that you want to install, then click Next.



Once you have installed Business Applications, you can deploy one or more of them selectively by specifying a container when you import the required DAC metadata (for more information, see [To import metadata into the DAC repository on page 93](#)).

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



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

NOTE: Even when the progress bar on the installer reports 100% complete, you must wait until the Finish button is displayed, which might take a long time (that is, up to 30 minutes).

- 7 Click Finish.

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

Running the Informatica PowerCenter Installation Wizard (Windows)

Follow this procedure to install Informatica PowerCenter.

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

To install Informatica PowerCenter on Windows

1 Access the installation files on the installation CD-ROM, and run the program launch.exe.

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

- 2 The installation wizard window appears and prompts you through each screen, as shown in the following table.

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

Screen	Your Action	Notes
Welcome	Click Next.	
Customer Information	Enter the user name, company name, and Product license key. NOTE: Do not enter the Connectivity license key or Options license key during the product installation.	Informatica provides three different license keys: ■ Product. Allows you to install the Informatica Server and Repository Server. Enter during the product installation and the procedures “Setting Up the Informatica Server and Informatica Repository Server” on page 79 and “Process of Loading the Pre-Built Repository Into Informatica” on page 115. ■ Connectivity. Allows you to access sources and targets. Enter during the product installation and the procedures “Setting Up the Informatica Server and Informatica Repository Server” on page 79 and “Process of Loading the Pre-Built Repository Into Informatica” on page 115. ■ Options. Allows you to access specific options, such as Team-Based Development, Server Grid, and Partitioning. Enter during the product installation and the procedures “Setting Up the Informatica Server and Informatica Repository Server” on page 79 and “Process of Loading the Pre-Built Repository Into Informatica” on page 115.
License Agreement	Select “I accept...” and click Next.	

Screen	Your Action	Notes
Select Components and Destination Folder	■ Select the following components: ■ InformaticaPowerCenter 7.1.4 OEM for Oracle - Client ■ InformaticaPowerCenter 7.1.4 OEM for Oracle - Server ■ InformaticaPowerCenter 7.1.4 OEM for Oracle - Repository Server ■ Browse for and select the appropriate destination folder, and then click Next.	
Program Folder	Select a program folder or accept the default, and then click Next.	NOTE: If you are using a Teradata database, make sure there are no spaces in the Informatica Server directory name or the directory path. The default directory contains spaces that you must remove manually.
Review Settings	Review the selected components and destination directory information, and then click Next.	The Informatica components are now installed on the machine.
Setup	Click OK.	This screen informs you that you must specify Productivity and Options license keys in later steps.
Setup Type	■ Remove the check marks from the Configure Informatica Server and Configure Informatica Repository Server check boxes. ■ (Optional) Select the View Readme check box to view the Read Me file. ■ Click Next.	You will configure the Informatica Server and Repository Server in later steps.
Wizard Complete	Click Finish.	

NOTE: Depending on how your Windows machine is set up, you might need to start the Informatica services manually (for more information, see [Process of Starting the Informatica Services on page 84](#)).

Setting Up the Informatica Server and Informatica Repository Server

After the Informatica Server is installed, you must configure it before you can start it, by doing the following:

- Setting up the Informatica Server in Informatica PowerCenter. For more information, see [Process Of Setting Up The Informatica Server on page 79](#).
- Setting up the Informatica Repository Server in Informatica PowerCenter. For more information, see [Process Of Setting Up The Informatica Repository Server on page 83](#).
- Copy source files and lookup files from the Oracle BI Applications installation folder to the Informatica folder. For more information, see [Process of Copying Source Files and Lookup Files on page 84](#).
- After you have created the Informatica repository (for more information, see [Process of Configuring and Managing the Informatica Repository on page 113](#)), start the Informatica Services. For more information, see [Process of Starting the Informatica Services on page 84](#).
- Create the required ODBC Database Connections. For more information, see [Process of Creating ODBC Database Connections on page 85](#).

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

TIP: Keep a record of the Informatica Server details that you specify here, and use the same details when you register the Informatica servers in the DAC (for more information, see [Registering Informatica Servers in the DAC Client on page 110](#)).

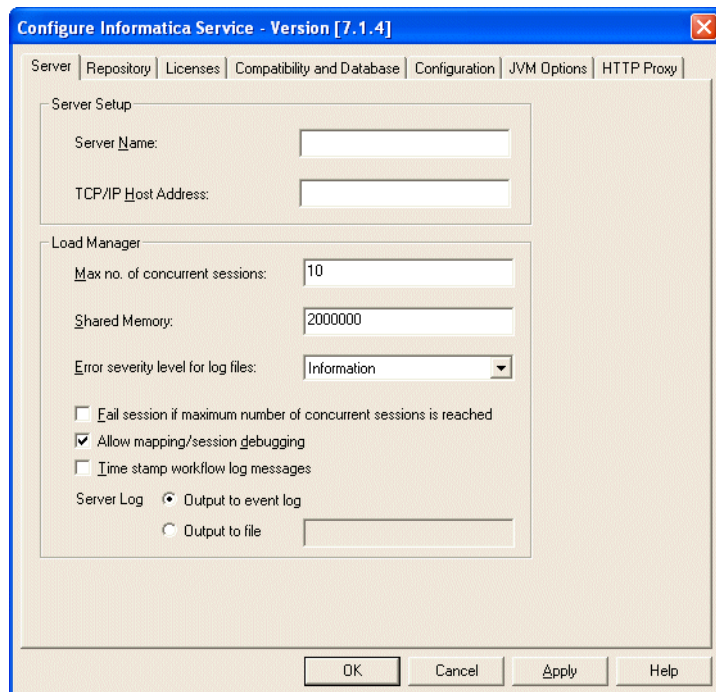
NOTE: Throughout this guide, the name Oracle_BI_DW_Base is used for the Informatica repository (the file \dwrep\Informatica\Repository\Oracle_BI_DW_Base.rep). However, you can use a different name if required, and have more than one version of this file if you have more than one development environment.

Process Of Setting Up The Informatica Server

To set up the Informatica Server in Informatica PowerCenter

- 1 Navigate to Programs > Informatica PowerCenter 7.1.4 > Informatica PowerCenter 7.1.4 OEM for Oracle - Server > Informatica Server Setup.

- 2 In the Informatica Server Setup Application - Options window, select Configure Informatica Service, and then click Continue.



- 3 In the Server tab, enter the following:

Field	Description
Server Name	A logical name for the Informatica Server. It is recommended that you enter ORACLE_BI_DW_SERVER.
TCP/IP Host Address	The IP address or name of the machine on which you plan to run the Informatica Server.
Max no. of concurrent sessions	From 1 to 15 sessions (default is 10).
Shared Memory	The default is 2,000,000. You should specify 200,000 for each session specified in the 'Max no. of concurrent sessions' field. For example, if you specify 5 in the 'Max no. of concurrent sessions' field, you should specify 1,000,000 in the Shared Memory field.

- 4 In the Repository tab, enter the following:

Field	Description
Repository Name	A logical name for the Informatica repository. It is recommended that you enter Oracle_BI_DW_Base. For Teradata databases, it is recommended that you enter Oracle_BI_DW_Teradata.
Repository User	The account used to access the repository. It is recommended that you use Administrator as the username.
Repository Password	The password you use to connect to the Repository Server from the Repository Manager. It is recommended that you use Administrator as the password.
Repository Server Host Name	The host machine name where the Informatica Repository Server is installed.
Repository Port Number	The port number the Repository Server uses to connect to repository client applications. By default, this value is 5001. It is recommended that you use the default value.

- 5 In the Licenses tab:

- a Leave the default License File Name as pm.lic.
- b If you have an option or connectivity license key, enter it in the “with Key” field, and click Update. Do this for each option and connectivity license key.

- 6 In the Compatibility and Database tab:

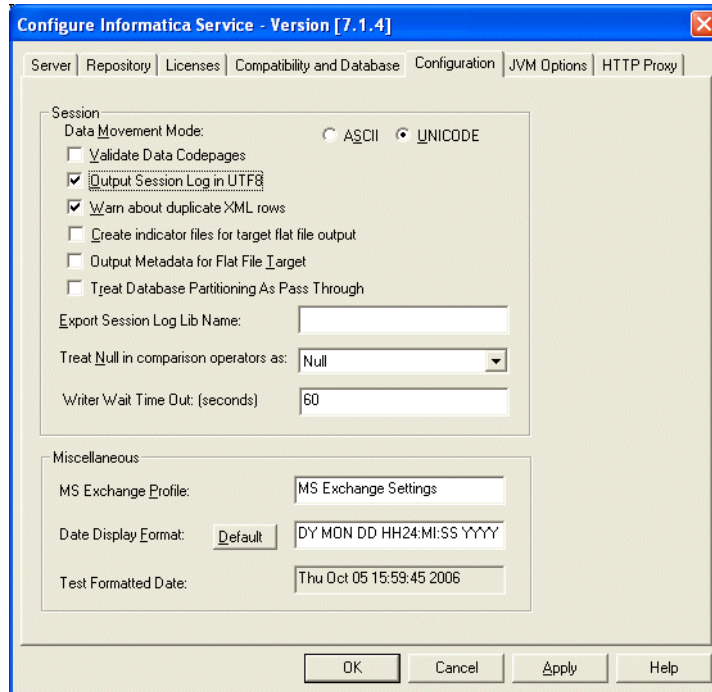
- a Make sure the “Treat CHAR as CHAR on read” option is selected.
- b Make sure the maximum number of database connections is 100.

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

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

- a Select the ASCII or UNICODE option.

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



- c** Leave the default values in the remaining fields.

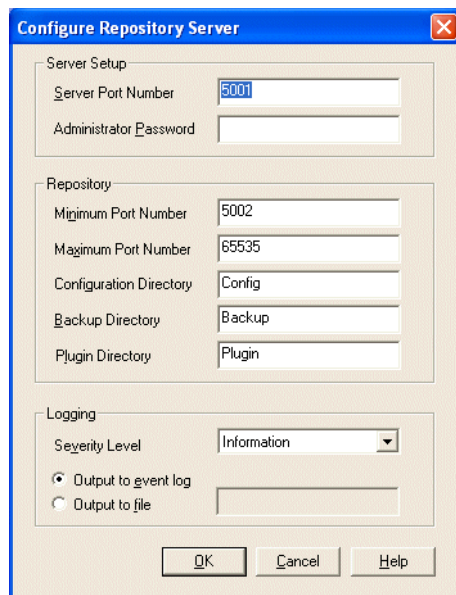
- 8** In the JVM Options tab, leave the default values in all fields.
- 9** Click OK in the Configure Informatica Service dialog box to complete the configuration of the Informatica Server service.

If you are using a Teradata database, you cannot start the Informatica Server until you have completed the procedures in the section [“Setting the Code Page File Property for Unicode Environments on Teradata Databases”](#) on page 124.

Process Of Setting Up The Informatica Repository Server

To set up the Informatica Repository Server in Informatica PowerCenter

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



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

Field	Description
Server Port Number	NOTE: The port number the Repository Server uses to connect to repository client applications. By default, this value is set to 5001. It is recommended that you use the default value. This value must match the value you entered in Setting Up the Informatica Server and Informatica Repository Server on page 79.
Administrator Password	The password you use to connect to the Repository Server from the Repository Manager. It is recommended that you use Administrator as the password. NOTE: This value must match the value you entered in Setting Up the Informatica Server and Informatica Repository Server on page 79.

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

Process of Copying Source Files and Lookup Files

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

To copy source files and lookup files:

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

Process of Starting the Informatica Services

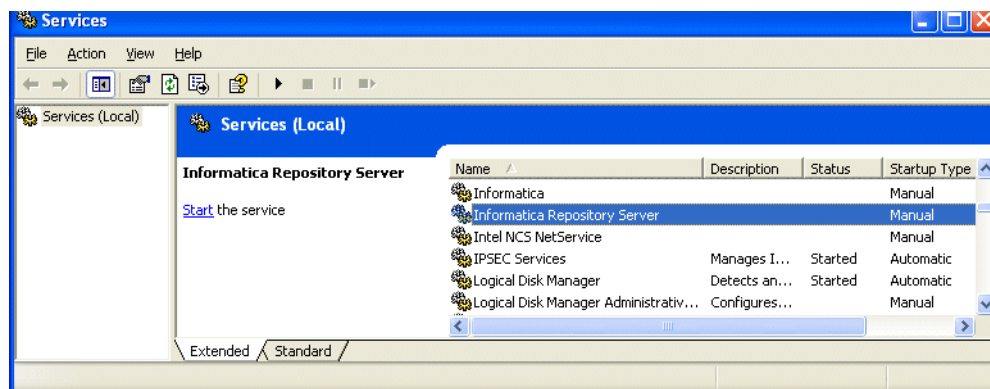
After you have installed Informatica PowerCenter, you might need to start the Informatica services manually. The Informatica services must be running to enable you to access the Oracle Business Analytics Warehouse using the Informatica tools.

You need to configure the Informatica repository before you start the Informatica services (for more information, see [Process of Configuring and Managing the Informatica Repository on page 113](#)).

To start the Informatica Services

- 1 Display the Windows Services dialog.

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



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

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

Process of Creating ODBC Database Connections

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

You also need ODBC connections when you create the data warehouse tables in the DAC in a later configuration task (for more information, see [Process of Creating Data Warehouse Tables on page 105](#)). In the case of Oracle database, use the Oracle Merant ODBC Driver that comes with the Oracle BI Applications install. For all other databases, you should use ODBC drivers supplied by your database vendor.

To create database connections for DB2 installations

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

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

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

To create database connections for Oracle installations

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

To create database connections for SQL Server installations

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

- an ODBC connection to the transactional database

NOTE: Select SQL Server as the ODBC driver.

- 2 Test the connections to make sure they work.

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

To create database connections for Teradata installations

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

Field	Enter
DateFormat	AAA
SessionMode	ANSI
NoScan	Yes

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

Setting Environment Variables on Windows

This section includes instructions for setting the following environment variables:

- [Setting the PATH Environment Variable for the Informatica Server and Informatica Repository Server on page 86](#)
- [Setting the Siebel UnicodeDB Environment Variable on Windows on page 87](#)
- [Setting the Bulk Load Registry for DB2/390 Databases on page 87](#)

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

Setting the PATH Environment Variable for the Informatica Server and Informatica Repository Server

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

To set the PATH environment variables for Informatica

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

Setting the Siebel UnicodeDB Environment Variable on Windows

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

To set Siebel UnicodeDB for Windows

- 1 In Windows, click Start > Run.
- 2 In the Open field, enter regedit.
- 3 Navigate to HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > PowerMart > Parameters > Configuration.
- 4 On the right window panel, right-click and select New > String Value.
- 5 Rename the new string value SiebelUnicodeDB.
- 6 Double-click SiebelUnicodeDB.
- 7 In the Value data field, enter the connection string for your transactional database followed by the connect string for your data warehouse database, using the following format:

[user_OLTP]@[connectString_OLTP] [user_OLAP]@[ConnectionString_OLAP].

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

- Always leave a space between the connection strings.
- For MSSQL, use the system DSN name for [connectString_OLTP] and [connectString_OLAP].

NOTE: You must enter the user names and connect strings in the same case as you used in the Workflow Manager > Connections > Relational settings.

Setting the Bulk Load Registry for DB2 / 390 Databases

DB2/390 databases do not support bulk loading; therefore, you need to disable bulk loading in the Windows registry:

HKEY_LOCAL_MACHINE\...\System\CurrentControlSet\Services\Powermart\Configuration area.

To disable bulk loading for DB2 / 390 databases

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

Installing the Informatica Patch for Teradata

NOTE: You only need to install the Informatica patch if you are using Teradata as the target database.

After you install Informatica PowerCenter version 7.1.4, you need to install one Informatica patches.

To install the Informatica patches on Windows

- 1 Access the patch files on the Oracle BI Applications CD-ROM.
- 2 Locate the folder named Patch1, and install the patch by following the instructions in the readme file.

Setting Informatica Data Code Pages Validation

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

```
[Code Pages]
```

```
ValidateDataCodePages=No
```

NOTE: If the powrmart.ini file does not contain a ValidateDataCodePages parameter, add this parameter as specified above.

Process of Copying Hibernate Libraries

To run the DAC Client or DAC Server, you need to have libraries from an open source software product called Hibernate installed on each machine that runs the DAC client or DAC server. Hibernate libraries are not installed as part of Oracle BI Applications. You must manually download Hibernate V3.2.2 and copy the libraries.

To copy Hibernate libraries

- 1 Go to the Web site <http://hibernate.org>.
- 2 Navigate to the Download area and download Hibernate Core Package Version 3.2.2. GA to a local machine.

- 3 Unzip the files and folders in the hibernate-3.2.x.ga.zip file (for example, hibernate-3.2.2.ga.zip).

For each machine that needs to run the DAC client or DAC server, copy the Hibernate files from the \hibernate-3.2 folder to the \OracleBI\DAC folder, as described in the following table:

Table 28. 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

Configuring the DAC Client

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

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

This section includes the following topics:

- [Process of Configuring the DAC Config.bat File on page 89](#)
- [Installing JDBC Drivers for DAC Database Connectivity on page 90](#)

Process of Configuring the DAC Config.bat File

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

To configure the DAC config.bat file

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

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

- 2 Edit the JAVA_HOME variable to point to the directory where you installed the Java SDK.

Make sure there are no spaces in the path reference.

For example:

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

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

For example:

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

Installing JDBC Drivers for DAC Database Connectivity

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

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

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

```
set ORACLELIB=%DAC_HOME%\lib\classes12.zip
```
- **DB2.** If you are using a DB2 database, find the directory where DB2 is installed. In the Java subdirectory 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 [Creating Stored Procedures for DB2-UDB on page 142](#).
- **MSSQL.** If you are using an MSSQL database, download the sqljdbc.jar driver file from Microsoft's Web site. Copy the sqljdbc.jar file to the \DAC\lib directory (for example, <DRIVE>:\OracleBI\DAC\lib\sqljdbc.jar).

NOTE: You need the SQL Server 2000 Driver for JDBC SP3 for SQL Server 2000.
- **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.

Creating the Required Databases

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

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

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

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

To create the required databases

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

If you perform step 1, you specify the name of the DAC metadata repository database (for example, DAC) when you log into the DAC (for more information about logging into the DAC, see [Creating the DAC Metadata Repository on page 91](#)).

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

NOTE: Make sure that you specify this database account when you create the data warehouse tables in the DAC (for more information, see [Process of Creating Data Warehouse Tables on page 105](#)).

If you omit step 1, you specify the name of the Oracle Business Analytics Warehouse database (for example, OLAP) when you log into the DAC (for more information about logging into the DAC, see [Creating the DAC Metadata Repository on page 91](#)).

Creating the DAC Metadata Repository

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

- [To create a DAC connection on page 91](#)
- [To log in to the DAC on page 92](#)
- [To import metadata into the DAC repository on page 93](#)

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

NOTE: Before you start this procedure, you need to create a database for the DAC metadata repository. This database can be the same as the data warehouse database. For more information, see [Creating the Required Databases on page 90](#).

To create a DAC connection

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

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

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

- 2 In the Login... dialog box, select Configure.
- 3 In the Configuring... dialog box, select Create Connection, and then click Next.

- 4 Enter the appropriate connection details.

NOTE: When you set up the DAC server later in the configuration (for more information, see [To configure the connection between the DAC server and the DAC repository on page 129](#)), you must use the same connection details that you specify here.

Field	Description
Name	A unique connection name that you specify here (for example, DAC).
Connection type	Type of database.
Database name	Database name or database account name in which you want to store the DAC metadata repository. If you are using... ■ 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	Machine name where the database resides.
Database Port	The port number on which the database listens. For example, for an Oracle database the default port is 1521, or for a SQL Server database the default port is 1433.

- 5 Select Test Connection to confirm that the login details are valid.

- 6 Select Apply to save the connection details and return to the login dialog.

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

To log in to the DAC

You log into the DAC to enable you to edit the DAC metadata repository. To 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 [To create a DAC connection on page 91](#).

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

The Login... dialog is displayed.



- 2 In the Login... dialog box, select a connection from the Connection drop down list.
- 3 In the Table owner name field, enter the database user name for the database in which you are creating the DAC metadata repository.
NOTE: For more information about creating a database for the DAC metadata repository, see [Creating the Required Databases on page 90](#).
- 4 In the Password field, enter the database password for the database user name that you specified in the previous step.
- 5 Click Login to start the DAC.

The DAC client starts and connects to the specified database.

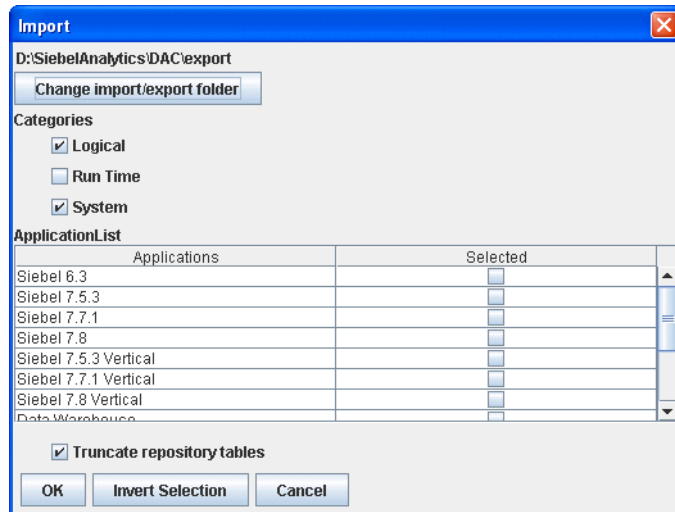
- 6 Click Yes if you are prompted to create or upgrade the repository tables.

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

To import metadata into the DAC repository

- 1 If you are not logged into the DAC, log into the DAC (for more information, see [To log in to the DAC on page 92](#)).

- 2 From the DAC menu bar, choose Tools > DAC Repository Management > Import.



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

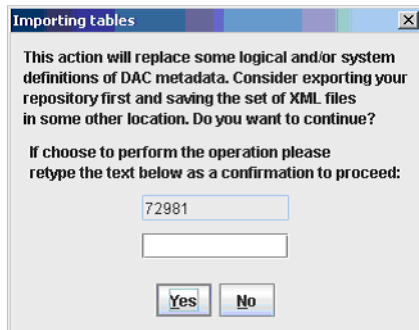
- 3 Using the check boxes in the Selected column, select the check box next to the business applications that you want to deploy.

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 [Process of Creating Data Warehouse Tables on page 105](#)).

- 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.

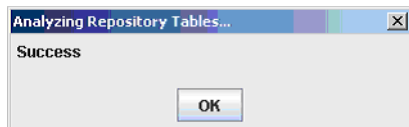
- 5 Click OK to display the Importing tables dialog.



- 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.

If the Import is successful, a 'Success' message is displayed.



- 8 Click OK.

- 9 If your source or target database is a DB2-390 database, run DAC_DB2390_Source.sql or DAC_DB2390_Target.sql immediately after importing the seed data by connecting to the database where the DAC repository resides. These files are stored in the \OracleBI\DAC\ folder.

Configuring the DAC Metadata Repository

This section includes the following topics:

- [Setting Up DAC System Properties on page 96](#)
- [Specifying Transactional and Data Warehouse Data Sources in the DAC on page 100](#)
- [Configuring Email Recipients in the DAC Client on page 103](#)
- [Activating Join Indices for Teradata Databases on page 104](#)

NOTE: You will configure the DAC server to point to the DAC metadata repository later in the installation process, in the procedure ["Configuring the DAC Server" on page 129](#).

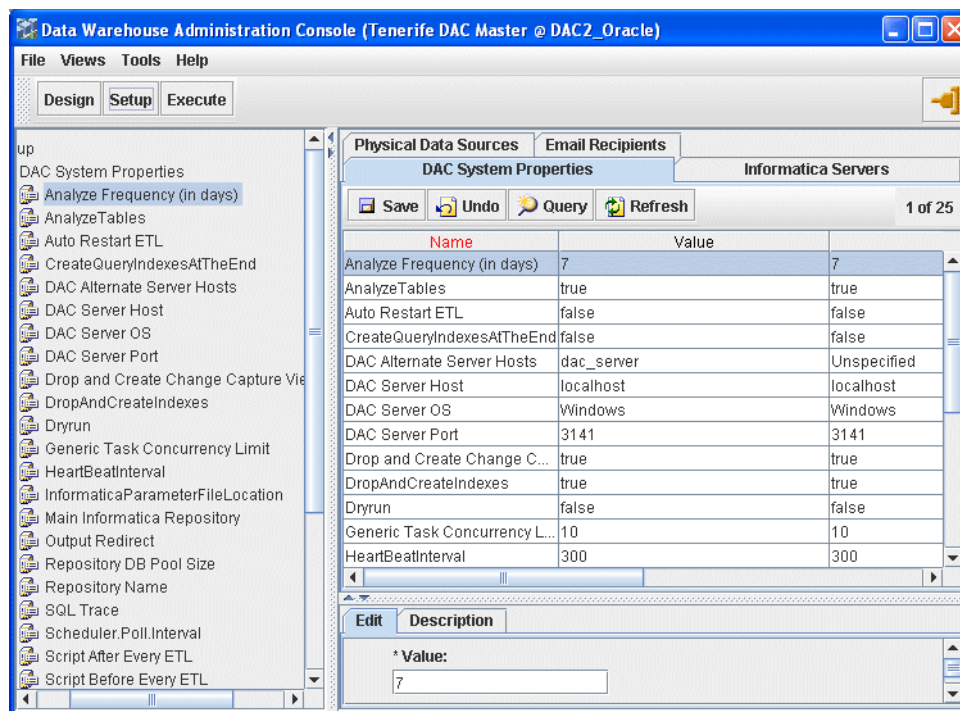
Setting Up DAC System Properties

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

To set up the DAC system properties

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

The DAC System Properties tab is active.



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

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

Property	Description
Analyze Frequency (in days)	For DAC metadata tables, the frequency (in days) the DAC client automatically updates the table and index statistics for the DAC repository. The value must be numerical.
Auto Restart ETL	Possible values are TRUE and FALSE. When set to TRUE: An ETL that is running when the DAC server abnormally terminates will continue running when the DAC server is restarted. When set to FALSE: An ETL that is running when the DAC server abnormally terminates will not automatically restart when the DAC server restarts. The ETL status will be updated to Failed. An administrator will have to manually restart the ETL.
CreateQueryIndexesAtTheEnd	Possible values are True and False. During the ETL process, the DAC server automatically drops and creates indices. When set to True, this property groups all indices of the Query type and creates them after the ETL is complete. The DropAndCreateIndexes property takes precedence over this property. Therefore, if the DropAndCreateIndexes property is set to False, you cannot set the property CreateQueryIndexesAtTheEnd to True to have indices of the Query type created at the end of the ETL process. Also, be aware that when this property is set to True, tables will be analyzed twice, (once at the end of the ETL indexes and once at the end of query indexes). If any indices are marked as Query type indices, and are used by ETL processes, it can adversely affect the performance of the ETL process.
DAC Alternate Server Hosts	The failover server for the DAC.

Property	Description
DAC Server Host	Host name of the machine where the DAC server resides. You cannot use an IP address for this property. NOTE: The DAC server and a given DAC repository have a one-to-one mapping. That is, you can only run one DAC server against any given DAC repository. Thus, in the repository you must specify the network host name of the machine where the DAC sever is to be run. NOTE: This property also takes the value <i>localhost</i>. However, this value is provided for development and testing purposes and should not be used in a production environment.
DAC Server OS	Operating system of the machine where the DAC server resides. Possible values are Windows, Solaris, HP, AIX, Linux. NOTE: If you move the DAC server from another operating system to AIX, you need to do the following: change the DAC server host to the appropriate value; restart the DAC client; reenter all the password fields for the Informatica servers and database connections; and reconfigure the DAC server on the AIX machine by running <code>serverSetupPrompt.sh</code>.
DAC Server Port	Network port to which the DAC server binds in order to listen to client requests. The default value is 3141. If this port has been assigned to another process, you can enter any numerical port value greater than 1024.
Drop and Create Change Capture Views Always	Possible values are True and False. When set to True (the default value), the DAC server drops and creates change capture views every time it performs a change capture process, including for both full and incremental loads. Setting this property to True can create system catalog lock up for DB2-UDB and DB2-390 databases. Therefore, by setting the property to False, the DAC server will drop and create views selectively. NOTE: This setting only applies to Siebel Adaptors. Change Capture Views are built on the Siebel (CRM) or Oracle E-Business Suite OLTP database.
Dryrun	Possible values are True and False. Indicates whether tasks are executed without invoking Informatica workflows. The following processes are executed: change capture, truncation of tables, drop and creation of indices, and analyze statements. Possible values are True and False. NOTE: This option should be used for debugging purposes only and not used in a production environment.

Property	Description
Generic Task Concurrency Limit	Determines how many tasks with execution types other than Informatica can be run concurrently. The value must be numerical. To set this value, you should consider what the external tasks do. For example, if the tasks open connections to a database, you should consider how this would affect the preconfigured tasks.
HeartBeatInterval	Frequency (in seconds) the DAC server checks on the health of the database connections. The value must be numerical. For example, a value of 300 (the default value) indicates the system will perform subsystem diagnostics and recovery procedures every 300 seconds.
InformaticaParameterFile Location	This file location can be any file system location that is accessible to both Informatica server and the DAC server. It is the same location that is entered as the Source File Directory the Informatica in the Informatica workflow manager. NOTE: This file location also should not have spaces.
Output Redirect	Indicates whether logging information and standard output and errors are redirected to files in the log directory (when property is set to True). The file containing standard output starts with out_ and ends with the .log extension. The standard error messages are in the file starting with err_ and ending with the .log extension. If this property is set to False, the logging information is directed to the machine's standard output and error files, which typically defaults to the console from which the DAC server was launched if the server was launched in a visible console mode.
Repository DB Pool Size	Maximum number of connections to the DAC repository the DAC server will maintain. The value must be numerical. NOTE: As the DAC server runs, it needs to perform transactions and queries on the DAC repository. In order for it to do so efficiently, it internally maintains a pool of preexisting database connections. This pool of database connections reduces latency and leads to better use of resources through connection sharing. However, you need to specify the maximum size of the connection pool. This setting should be determined in consultation with the database administrator of the database where the DAC repository resides. NOTE: This property was deprecated as of version 7.7.1.5.
Repository Name	Unique name for the DAC repository (for example DAC).
Scheduler.Poll.Interval	Frequency (in seconds) the DAC server polls for changes in the schedule configuration.

Property	Description
Script After Every ETL	The name of the script or executable to be run after every execution plan. For more information, see the description of the property Script Before Every ETL.
Script Before Every ETL	The name of the script or executable to be run before every execution plan. For example, before running an execution plan, you might want to run a process or perform certain tasks. These can be contained in a script or executable. This file should be placed in the scripts subdirectory of the DAC server. The execution plan runs only after the external process has finished. Therefore, it is important that the script or executable does not fail.
Server Log Level	Output logging level. Possible values are Finest, Finer, Fine, Config, Info, Warning, and Severe. The Severe value produces minimal log details, and Finest produces the most extensive amount of reporting.
SQL Trace	Possible values are True and False. Indicates whether the SQL statements to the DAC repository and database connections are added to the log file. Possible values are True and False. The True value sends a hint to the database connectivity layer of the DAC server to enable SQL tracing; thus, every SQL statement that is run by the DAC server is spooled to the appropriate output log file. It is recommended that you set this property to False.
Verify and Create Non-Existing Indices	Possible values are True and False. Indicates whether indices defined in the DAC repository will be automatically created in the data warehouse database during an incremental load. NOTE: When this system property is set to True, the DAC server verifies whether indices defined in the DAC repository are also defined in the data warehouse database. This verification process can delay the execution of an execution plan.

Specifying Transactional and Data Warehouse Data Sources in the DAC

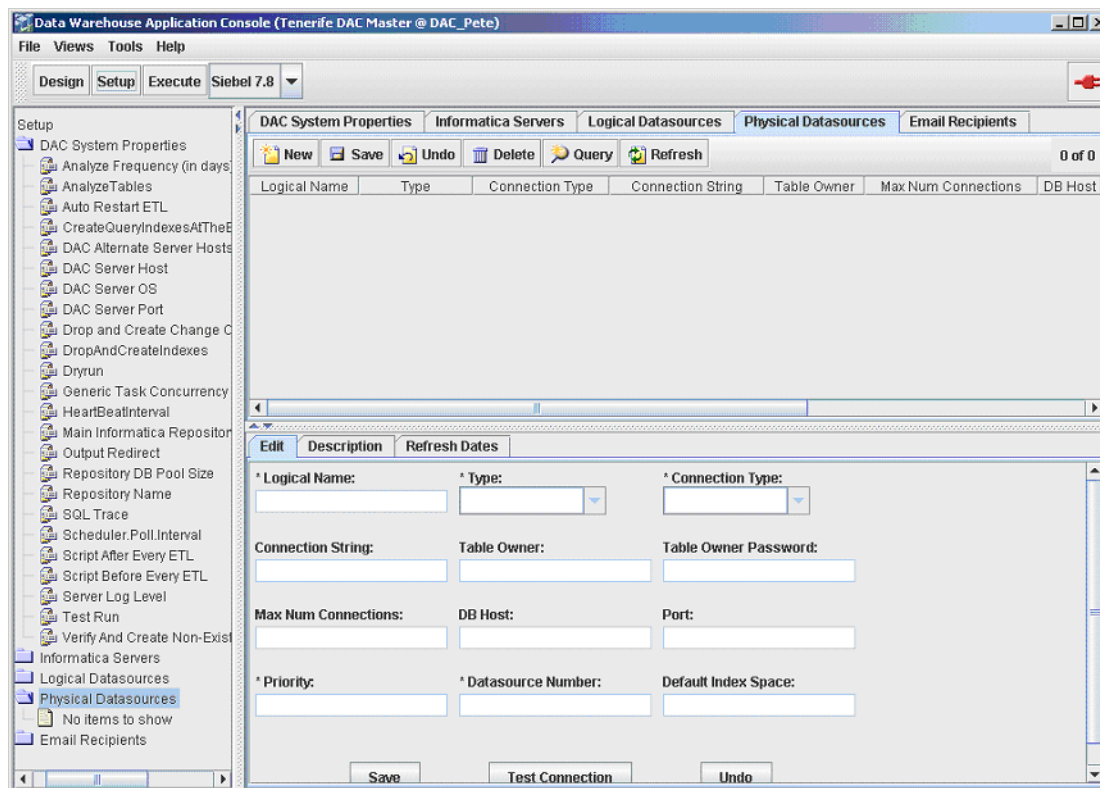
Follow this procedure to specify the transactional and data warehouse data sources in the DAC.

Oracle BI Applications is installed with a set of template data sources that you can edit to specify the data source details. Alternatively, you can specify new data sources.

To specify transactional and data warehouse data sources

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

- 1 Launch the DAC client.
- 2 Click Setup on the toolbar, and then click the Physical Datasources tab.
- 3 Click New.



- 4 In the Edit subtab, enter the following:

Field	Description
Logical Name	Logical name for the database connection.
Type	■ Select Source when you create the database connection for the transactional database. ■ Select Warehouse when you create the database connection for the data warehouse database.
Connection Type	Type of database. Possible values are: ■ Oracle (OCI8) ■ Oracle (Thin) ■ DB2 ■ DB2-390 ■ MSSQL ■ Teradata ■ Flat File
Database Name	Name of database that you have created for the Oracle Business Analytics Warehouse or transactional system (for more information, see Creating the Required Databases on page 90).
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 where the database resides.
Port	Port number where the database listens.
Priority	Number used to generate dependencies when designing execution plans.

Field	Description
Datasource Number	Unique number assigned to the source connection so that the data coming from this source can be identified in the data warehouse. A system column called DATASOURCE_NUM_ID stores a number value that identifies where data originated. This value is passed as a parameter to an Informatica workflow. If you are using multiple sources, each data sources 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.

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

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

Configuring Email Recipients in the DAC Client

Follow this procedure to configure email recipients.

To configure email recipients in the DAC client

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

Field	Description
Name	Logical name of the user to be notified.
Email Address	Email address where the notification is sent.

Field	Description
Notification Level	The notification levels are as follows: ■ 10 -- Notifies recipient of success or failure of each task. ■ 5 -- Notifies recipient of success or 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 in Tools DAC Server Setup).

Activating 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.

Configuring the Oracle Business Analytics Warehouse Database

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

- [Process of Creating the SSE Role on page 105](#)
- [Process of Creating Data Warehouse Tables on page 105](#)
- [Process of Creating Data Warehouse Tables on a Teradata Database on page 109](#)

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

Process of Creating the SSE Role

Follow this procedure to create the SSE role.

To create the SSE Role

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

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

Note the following:

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

Process of Creating Data Warehouse Tables

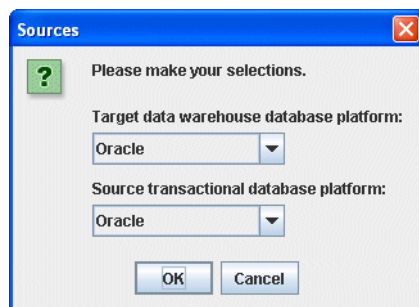
Use this procedure to create the data warehouse tables. Before you perform this procedure make sure that the SSE role has been created for the Oracle Business Analytics Warehouse and transactional database, and that the database user has been associated with the role (for more information, see [Process of Creating the SSE Role on page 105](#)).

If you are using a Teradata database, the DAC creates a SQL file to create the schema tables, not the tables themselves. Therefore, after you have followed this procedure, you need execute the SQL file created by following the instructions in the section [“Process of Creating Data Warehouse Tables on a Teradata Database”](#) on page 109.

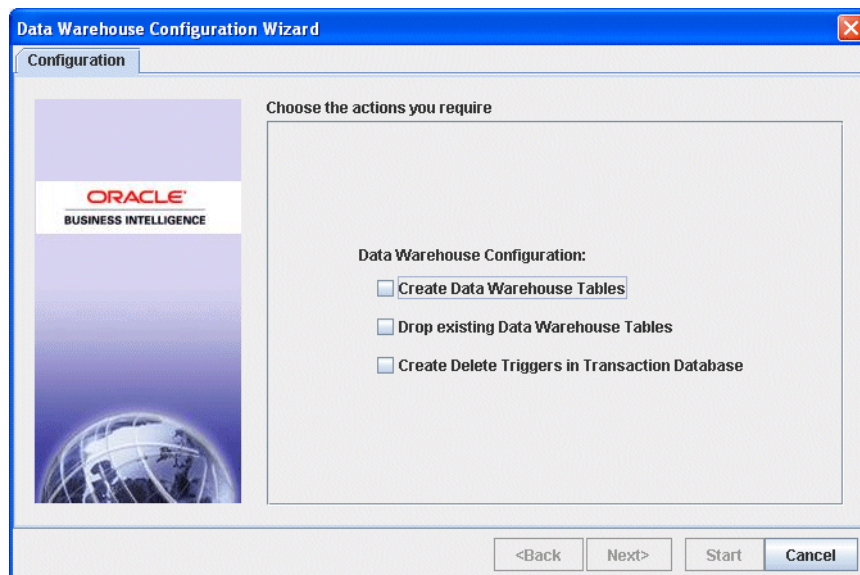
NOTE: Additional work by the database administrator is required if the data warehouse tables need to be moved to different tablespaces for performance or manageability reasons.

To create data warehouse tables

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



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



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

Data Warehouse Configuration Wizard

Configuration Data Warehouse

Choose options that best suit your needs

Database Type
Oracle

Container (leave empty for all containers)

Table Owner

Password

ODBC Data Source

Data Area

Index Area

<Back Next> Start Cancel

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.
Container	The container for the source business applications (or containers) that you want to deploy. The source business application or applications that you selected when you imported the seed data into the DAC metadata repository (for more information, see Creating the DAC Metadata Repository on page 91). 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 create a schema based on the merged containers for that DAC repository, leave this field blank. For example if the DAC repository contains Siebel 7.8 and Oracle 11.5.10 containers: □ If the container name is left blank, the schema created will combine all the tables that are required for both adapters. □ If the container name is specified as Siebel 7.8, the schema is created with the tables that are necessary for only the datawarehouse container. ■ If you want to deploy Oracle 11.5.10, enter: Oracle 11.5.10 ■ 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 <i>Oracle Business Intelligence Data Warehouse Administration Console Guide</i>.
Table Owner	Valid database owner, username, or account that you set up to hold the data warehouse (for more information, see Creating the Required Databases on page 90).
Password	Valid database user password for the database owner, username, or account that you specified in the Table Owner field (for more information, see Creating the Required Databases on page 90).

Field	Description
ODBC Data Source	Data Source Name (DSN) for the Oracle Business Analytics Warehouse. You need to specify the name of the ODBC connection that you created for the data warehouse (for more information, see Process of Creating ODBC Database Connections on page 85).
Data Area	(Optional) Tablespace where data warehouse tables are created.
Index Area	(Optional) Indexspace where data warehouse indices are created (applicable only to Oracle and DB2 databases).
Is Unicode	Specifies whether the data warehouse database is Unicode.

6 Click Start.

The Run Status tab displays information about the data warehouse creation. If a 'Success' message is displayed, the data warehouse tables have been created. If you a 'Failure' message is displayed, the data warehouse tables have not been created (refer to the \DAC\log\config\createtables.log file to diagnose the failure).

Process of Creating Data Warehouse Tables on a Teradata Database

This section includes the procedures for creating and dropping data warehouse tables on a Teradata database. Before you start, make sure that the tables that you create are case specific by setting the session mode to ANSI in your Teradata ODBC configuration (for more information, see [Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse on page 47](#)).

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 Follow the steps in "[Process of Creating Data Warehouse Tables](#)" on page 105.

For Teradata databases, the DAC creates a SQL file to create the schema tables, not the tables themselves.

- 2 Copy the SQL file created in step 1 from \conf\sqlgen\sql\Teradata into SQL Assistant and execute the SQL.

Registering Informatica Servers in the DAC Client

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

TIP: Specify Informatica Server and Repository Server details that are consistent with the Informatica details that you specified when you set up the Informatica Server (for more information, see [Setting Up the Informatica Server and Informatica Repository Server on page 79](#)).

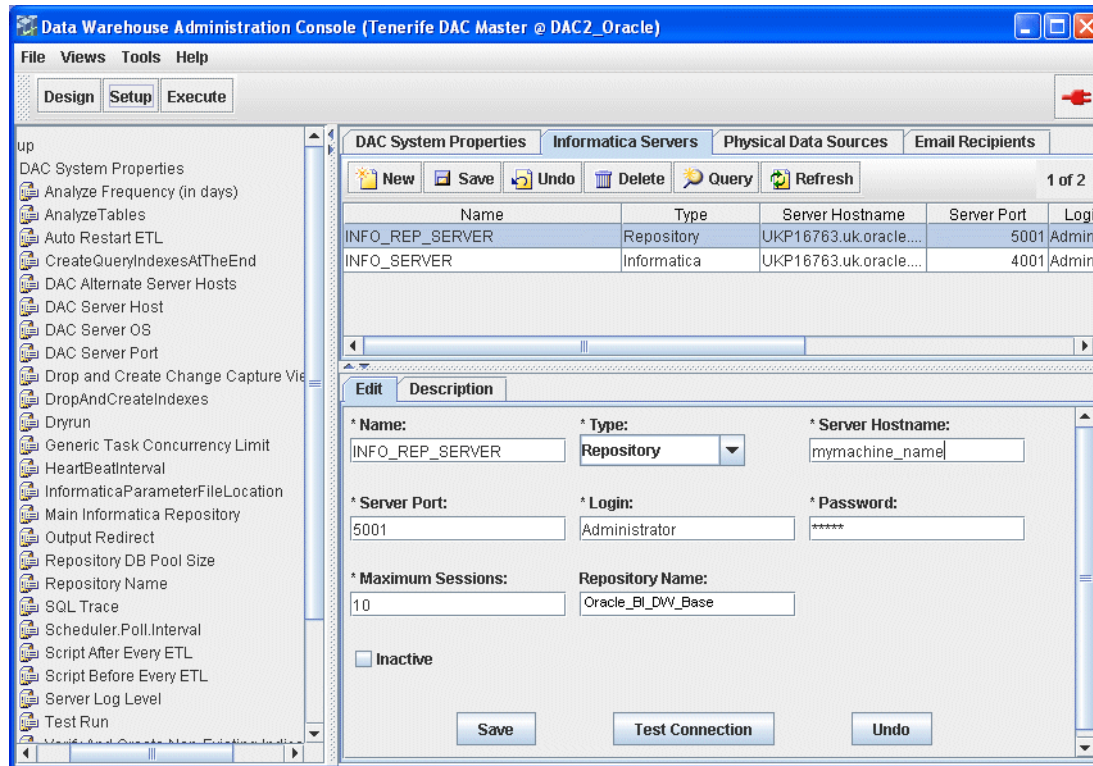
Notes

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

To register Informatica servers in the DAC client

- 1 Launch the DAC client.
- 2 Click Setup on the DAC toolbar, and then click the Informatica Servers tab.

- 3 Click New to display the Edit tab below.



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

- 4 Register the Informatica server, by entering the following information:

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

NOTE: You need to specify Informatica Server and Informatica Repository Server details that are consistent with the Informatica details that you specified when you set up the Informatica Server (for more information, see [Setting Up the Informatica Server and Informatica Repository Server on page 79](#)).

- 5 Click Save to save the details.
- 6 Click New to display the Edit tab below

- 7 Register the Informatica Repository server, by entering the following information:

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

NOTE: You need to specify Informatica Server and Informatica Repository Server details that are consistent with the Informatica details that you specified when you set up the Informatica Server (for more information, see [Setting Up the Informatica Server and Informatica Repository Server on page 79](#)).

- 8 Click Test Connection to make sure that the connection works.
- 9 Click Save to save the details.

If you have not already started the Informatica servers (for more information, see [Process of Starting the Informatica Services on page 84](#)), start Informatica servers now, then go back and test the connections using the Test Connection button.

Process of Configuring and Managing the Informatica Repository

This section contains instructions for configuring and managing the Informatica repository.

A pre-built Informatica repository called Oracle_BI_DW_Base is copied into the OracleBI\dwrep\Informatica\Repository folder during the Oracle BI Applications installation.

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

- 1** Use the Informatica Repository Administration Tool to register the Informatica Server machine. For more information, see [Process of Registering the Informatica Repository Server on page 114](#).
- 2** Use the Informatica Repository Administration Tool to load the pre-built Oracle_BI_DW_Base.rep repository. For more information, see [Process of Loading the Pre-Built Repository Into Informatica on page 115](#).
- 3** Use the Informatica Workflow Manager tool to configure the Informatica Repository. For more information, see [Process of Configuring the Informatica Repository in Workflow Manager on page 118](#).
- 4** Start the Informatica services. For more information, see [Process of Starting the Informatica Services on page 84](#)).

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

- stop and start the repository. For more information, see [Process of Stopping and Starting the Informatica Repository \(Optional\) on page 121](#).
- create and restore backup versions. For more information, see [Process of Backing Up and Restoring the Informatica Repository \(Optional\) on page 122](#).

Process of Registering the Informatica Repository Server

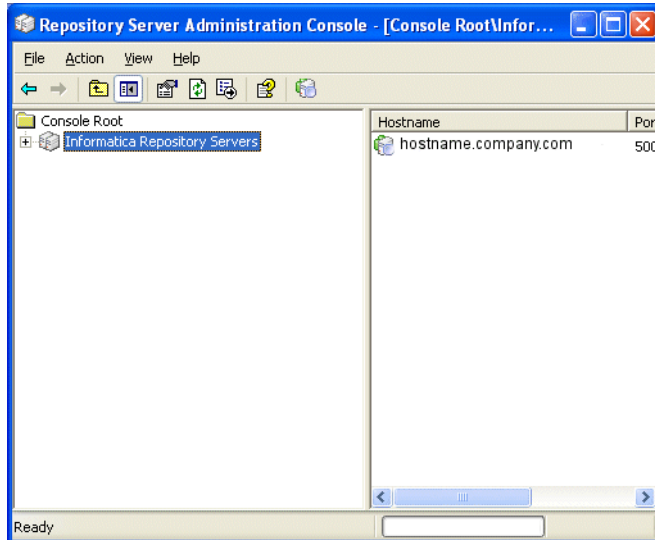
Follow this procedure to register the Informatica Repository Server in the Informatica Repository Server Administration Console.

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

To register the Informatica Repository Server

- 1** Launch the Repository Server Administration Console.

- 2 In the left pane, select the Informatica Repository Servers node, which appears under Console Root.



- 3 From the menu bar, select Action > New Server Registration to display the 'Register a new Repository Server' dialog.
 - a Enter the host name (the machine where the repository server resides).
 - b Accept the default port number 5001 or enter the appropriate port number. Click OK.
The Repository Server host name appears in the right pane under Hostname.
 - c Right-click the server name and choose Connect.
 - d In the Connecting to Repository Server dialog, enter the Administrator password. Click OK.These values should match the values you entered in [Setting Up the Informatica Server and Informatica Repository Server on page 79](#).

Process of Loading the Pre-Built Repository Into Informatica

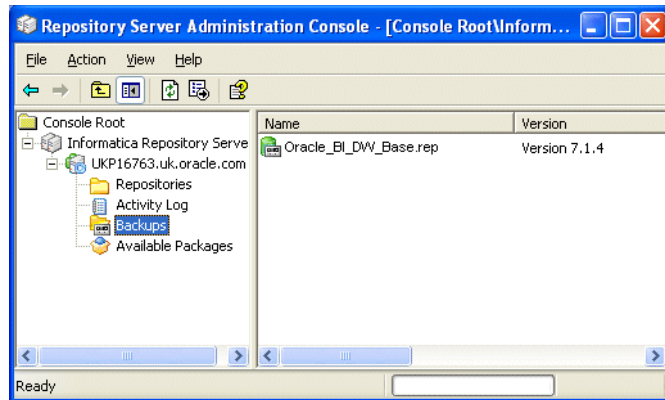
You use the Restore option in the Informatica Repository Administration Tool to load the pre-built Oracle_BI_DW_Base.rep repository into Informatica.

NOTE: When you install Oracle BI Applications in a Teradata environment, you must load the pre-built Informatica repository file `\dwrep\Oracle_BI_DW_Teradata.rep`.

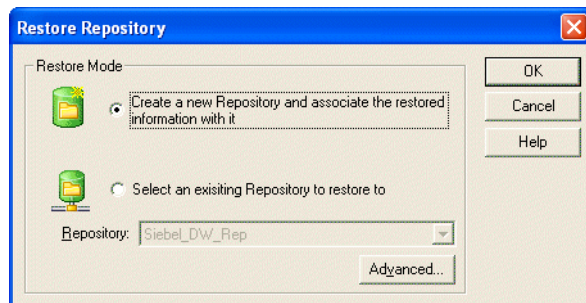
Before you start this procedure, you need to create a database to contain the Informatica Repository. You can use the same database as the data warehouse database. For more information, see [Creating the Required Databases on page 90](#).

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

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



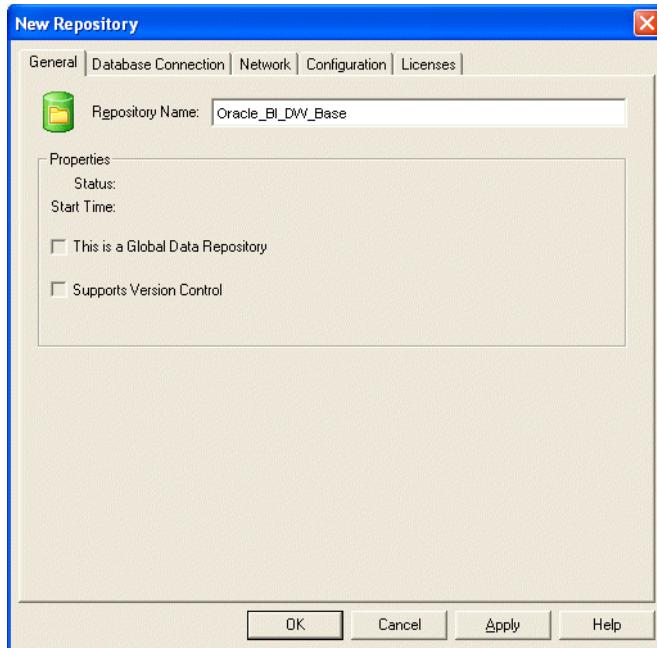
- 4 Right click on Oracle_BI_DW_Base.rep and click Restore to display the Restore Repository dialog.



- 5 Click the 'Create a new Repository and associate the restored information with it' check box.
- 6 Click OK to start the Restore procedure and display the New Repository dialog.

- 7 In the General tab, in the Repository Name field, enter the name for the Informatica repository (for example, Oracle_BI_DW_Base).

NOTE: This value should match the value you entered in “Setting Up the Informatica Server and Informatica Repository Server” on page 79 (for example, Oracle_BI_DW_Base).



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

Field	Description
DatabaseType	The type of database storing the repository.
CodePage	The code page based on the operating system platform.
ConnectionString	The native connect string of the database that stores the data warehouse. For MSSQL Server databases, use <SERVERNAME@DATABASENAME>.
DBUser	The account used to access the DAC repository (for example, DAC).
DBPassword	The password for the DAC repository user (for example, DAC).

- 9 Click the Licenses tab.
You may see a message stating the license file is empty.
- 10 Click OK to close the message dialog box.

- 11 In the License Key field, enter the Product license key, and then click Update.

The license key information appears in the lower window of the Licenses tab.

- 12 Repeat the step above for the Option and Connectivity license keys.

- 13 Click OK to save the repository details and create the repository.

- 14 Select the repository, right-click and then click Start to start the repository.

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

Process of Configuring the Informatica Repository in Workflow Manager

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

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

- 1 Launch Informatica Workflow Manager.

- 2 In the Repositories list, select the Informatica repository.

- 3 Choose Repository > Connect.

The Connect to Repository window opens.

- 4 Log in using the username and password you entered in [“Setting Up the Informatica Server and Informatica Repository Server”](#) on page 79 (for example, with the user name Administrator).

- 5 Click Connect.

- 6 Once connected, double-click the icon of the server name (ORACLE_BI_DW_SERVER is the recommended server name) under the repository icon.

The Server dialog box opens.

- 7 In the Host Name field, enter the computer name (as shown in the Network control panel) of the machine running the Informatica Server.

- 8 Click Resolve Server to confirm the computer name and update the IP address.

- 9 (Optional) Configure the Code Page setting.

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

- 10 Click Advanced to expose the Server Variables list.

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

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

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

- 13 Close the Server window.

To configure the database connections

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

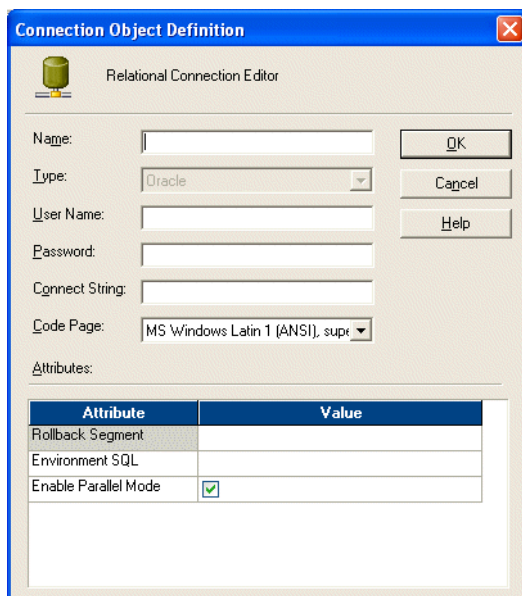
Attribute	Value
Rollback Segment	
Environment SQL	
Enable Parallel Mode	☒

- b If the target database is Oracle, DB2, or Teradata, use the settings as follows:
 - Click New and select the type as Oracle, DB2, or Teradata, then click OK.
 - Name: DataWarehouse (exactly as specified under DAC >setup>Physical Data sources for Data Warehouse).
 - User Name: Table Owner.
 - Password: TBO password.
 - Connect string: Connect string for the connecting to the database.
 - Code Page: UTF-8 encoding of Unicode if the database is unicode.
- c If the target database is SQL Server, use the settings as follows:

- ❑ Click New and select the type as ODBC, then click OK.
- ❑ Name: DataWarehouse (type in the name exactly as specified under DAC > Setup > Physical Data sources for your OLTP (for example, if your source system is Siebel ver 7.8.x then you name this connection as SEBL_78).
- ❑ User Name: Table Owner.
- ❑ Password: TBO password.
- ❑ Connect string: Connect string for the connecting to the database.
- ❑ Code Page: UTF-8 encoding of Unicode if the database is unicode.

3 Create connections for the source warehouse, as follows:

- a** Click New, select the appropriate sub-type, then click OK to display the Relational Connection Editor.



Attribute	Value
Rollback Segment	
Environment SQL	
Enable Parallel Mode	☒

- b** If the target database is Oracle, DB2, use the settings as follows:

- ❑ Click New and select the type as Oracle or DB2, then click OK.
- ❑ Name: Type in the name exactly as specified under DAC > Setup > Physical Data sources for your OLTP (for example, if your source system is Siebel ver 7.8.x then you name this connection as SEBL_78).
- ❑ User Name: Table Owner.
- ❑ Password: TBO password.
- ❑ Connect string: Connect string for the connecting to the database.
- ❑ Code Page: UTF-8 encoding of Unicode if the database is unicode.

- c** If the target database is SQL Server, use the settings as follows:

- ❑ Click New and select the type as ODBC, then click OK.

- ❑ Name: Type in the name exactly as specified under DAC > Setup > Physical Data sources for your OLTP (for example, if your source system is Siebel ver 7.8.x then you name this connection as SEBL_78).
- ❑ User Name: Table Owner.
- ❑ Password: TBO password.
- ❑ Connect string: Connect string for the connecting to the database.
- ❑ Code Page: UTF-8 encoding of Unicode if the database is unicode.

4 Click Close to close the Relational Connection Browser.

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

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

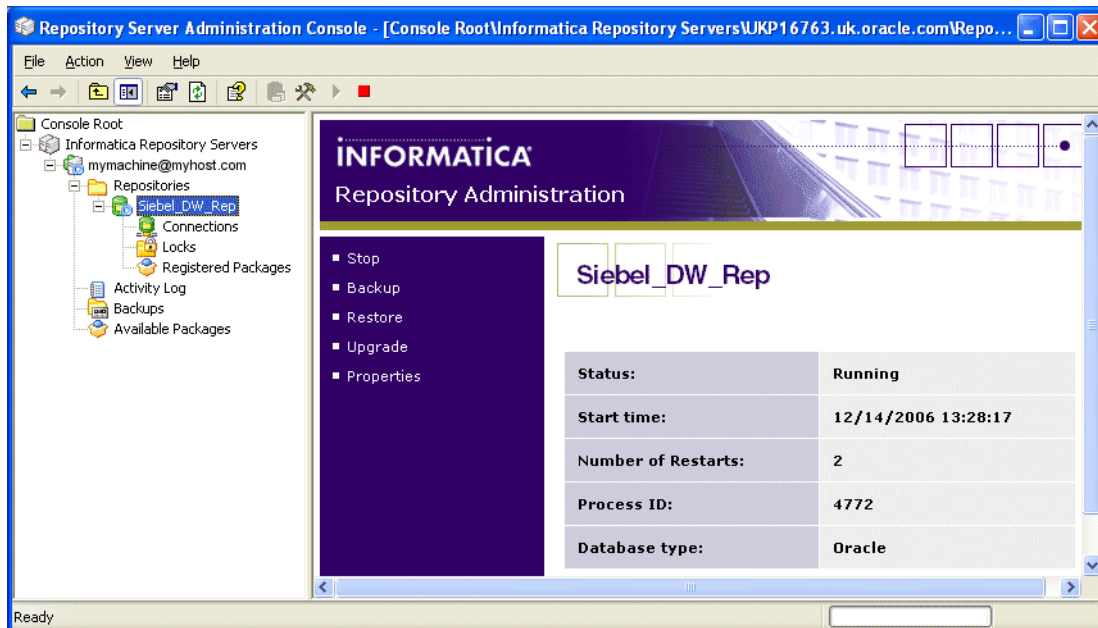
Process of Stopping and Starting the Informatica Repository (Optional)

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

To stop and start the Informatica repository

- 1 Start the Repository Server Administration Console in Informatica PowerCenter.
- 2 Display the Informatica Repository Administration dialog, as follows:

- a In the console navigator at the left hand side, double click the Informatica Repository Server to expand this node.
- b Right click on the server machine name that holds the Informatica Repository Server
- c Choose Action | Connect to display the Connecting to Repository Server <name> dialog.
- d Enter the Administrator password for this machine (do not change the default port number 5001), and click OK.
- e Double click the Repositories node.
- f Double click the Oracle_BI_DW_Base repository.



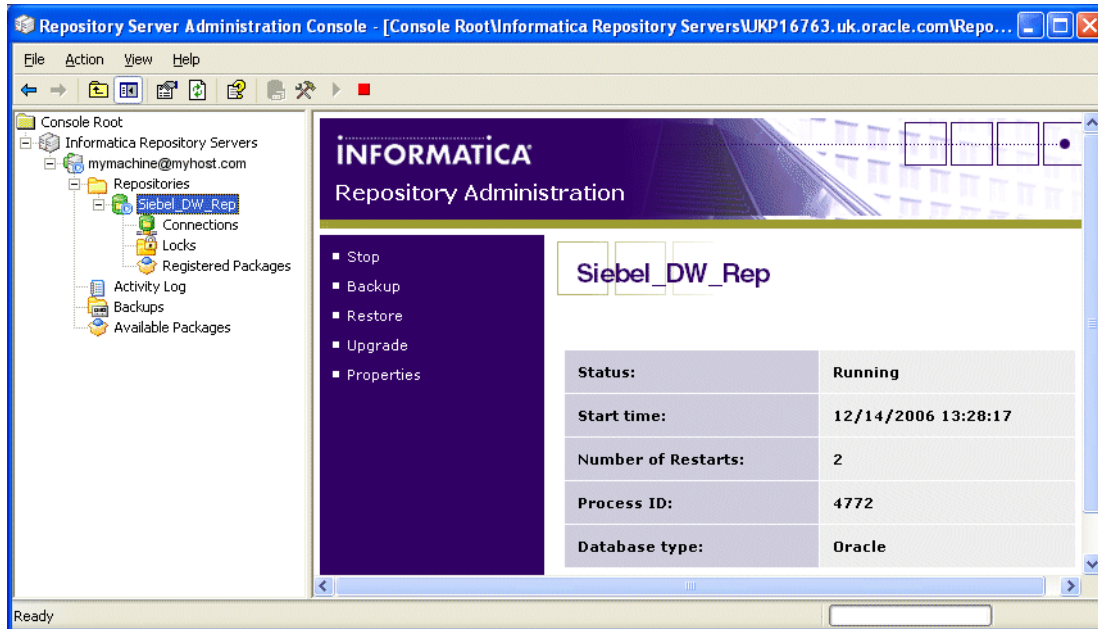
- 3 To stop the Informatica Repository, click the Stop option.
- 4 To start the Informatica Repository, click the Start option.

Process of Backing Up and Restoring the Informatica Repository (Optional)

To backup and restore the Informatica repository

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

- c Choose Action | Connect to display the Connecting to Repository Server <name> dialog.
- d Enter the Administrator password for this machine (do not change the default port number 5001), and click OK.
- e Double click the Repositories node.
- f Double click the Oracle_BI_DW_Base repository.



- 3 To backup the Informatica Repository, do the following:

- a Click the Backup option to display the Backup Repository dialog.
- b Enter the Informatica Repository Administrator username and password.
- c In the File Name field, enter a short name for the repository backup (for example, Inf_rep_backup).
- d Click OK to start the backup.

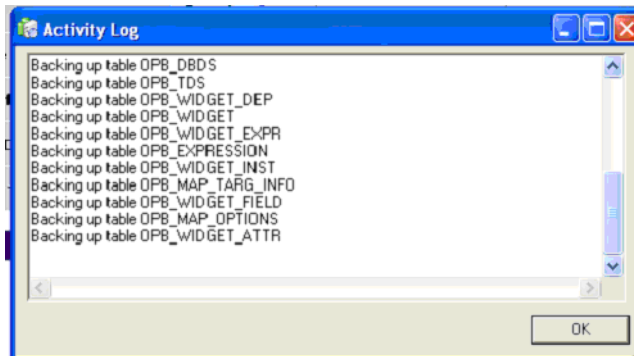
Informatica backs up the repository to the <DRIVE>:\Program Files\Informatica PowerCenter 7.1.4\RepositoryServer\bin\Backup folder.

- 4 To restore the Informatica Repository, do the following:

NOTE: As a pre-requisite step, you must have backed up the Informatica Repository, as described in the previous step 3.

- a Right click on the Oracle_BI_DW_Base node and click Delete to remove the Informatica Repository.
- b Click the Restore option to display the Restore Repository dialog.
- c Select a backup version of the Informatica Repository.
- d Click OK to start the procedure.

- 5 Check the Activity Log dialog to make sure that the backup or restore executes without errors.



Setting the Code Page File Property for Unicode Environments on Teradata Databases

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

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

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

Setting Up the HOSTS File for Teradata Installations on Windows

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

To set up the HOSTS file for Teradata installations

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

For example:

```
172.20.176.208 tdatsvr tdatsvrCOP1
```

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

- 3 Save the HOSTS file.

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

Verifying the Informatica Repository Configuration

Follow this procedure to verify the Informatica repository configuration.

To verify the Informatica repository configuration

- 1 Launch the Informatica Workflow Manager.
- 2 Double-click the ORACLE_BI_DW_SERVER icon that appears under the Oracle_BI_DW_Base icon.
The Server dialog box opens.
- 3 Click Advanced to expose the Server Variables list.
- 4 Verify the following settings:
 - Host Name
 - Port
 - Code Page
 - \$PMRootDir
- 5 Close the Server dialog box.
- 6 In the menu bar, select Connections > Relational.
The Relational Connection Browser window opens.
- 7 Verify the DataWarehouse and OLTP connection settings.

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.

The following table shows the tables used to store calendar information.

Table 29. 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.

Setting Up Fiscal Calendar

Installed out of the box, Oracle BI Applications supports one fiscal calendar. Fiscal data is first loaded in the W_DAY_D table and then the SIL mappings read data from W_DAY_D and load data into the aggregate Fiscal Time Dimension tables such as Fiscal Week, Fiscal Month, Fiscal Quarter and Fiscal Year.

You may choose to provide Fiscal calendar information in terms of the Fiscal Weeks of your organization or in terms of the Fiscal months of your organization. In either case, The SIL mappings are designed to derive the Fiscal Week from the Start and End Date of a Fiscal Month by grouping into periods of seven days each.

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.

Setting Up Fiscal Calendar

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.

Process of Setting Up The Fiscal Calendar

To configure Fiscal Calendar by Fiscal Week

- 1 Open the file fiscal_week.csv using Microsoft WordPad or Notepad in the \$PMRootDir\SrcFiles folder. Enter the Fiscal Year, Fiscal Month, Fiscal Week and the Start Date of Fiscal Week in YYYYMMDD. Records must be inserted in ascending order of Fiscal Year, Fiscal Month, Fiscal Week. Save and close the file.
NOTE: Make sure you enter accurate Fiscal Data as there is no check done within the Informatica mappings.
- 2 Deactivate the task SIL_DayDimension_FiscalMonth_Extract in the Data Warehouse container in DAC. Then right click and push to references.
- 3 Activate the task SIL_DayDimension_FiscalWeek_Extract in Datawarehouse container in DAC. Then right click and push to references.

To configure Fiscal Calendar by Fiscal Month

- 1 Open the file fiscal_month.csv using Microsoft WordPad or Notepad in the \$PMRootDir\SrcFiles folder. Enter the Fiscal Year, Fiscal Month and the Start Date of Fiscal_Month in YYYYMMDD format. Save and close the file.
NOTE: Make sure that you enter accurate Fiscal Data as there is no check done within the Informatica mappings.
- 2 Activate the task SIL_DayDimension_FiscalMonth_Extract in Datawarehouse container in DAC. Then right click and push to references.
- 3 Deactivate the task SIL_DayDimension_FiscalWeek_Extract in Datawarehouse container. Then right click and push to references.

Reloading the Time Dimension Tables After Your Data Warehouse is Populated

The data in time dimension is loaded once during the initial full load. Subsequently, the SIL_*_UpdateFlag mappings run everyday to update the domain value codes, which indicate whether a day, week, month, quarter or year is 'Current', 'Next' or 'Previous' as of the current day. The SIL_Fiscal_UpdateFlag mappings also update the flags that indicate whether a fiscal week, month, quarter or year is 'Current', 'Previous' or 'Next' with respect to the system date.

You might want to extend the range of data that you have in your time dimension sometime after the warehouse is in production. In order to achieve this, please follow the steps below that will kick off the full load ETL run of the W_DAY_D and all the aggregate time dimension tables.

To set up the load strategy of the time dimension table

1 In DAC, go to Setup -> Physical Data Sources and click on the connection DataWarehouse.

2 Go the RefreshDates tab in the bottom panel.

The Refresh Date of all the tables will be given there. Double click on the RefreshDate of W_DAY_D and make it Null. Do the same for all the 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 Go to the TaskDefinition 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.

NOTE: 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.

Notes

- 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.

Configuring the DAC Server

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

This section includes the following procedures:

- [Process of Configuring the Connection Between the DAC Server and DAC Repository on page 129](#)
- [Process of Configuring Email in the DAC Server \(Optional\) on page 130](#)
- [Process of Setting DAC Source System Parameters on page 131](#)
- [Process of Starting the DAC Server on page 137](#)

Process of Configuring the Connection Between the DAC Server and DAC Repository

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

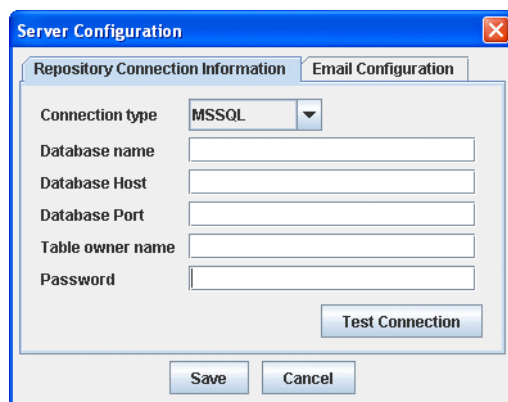
To configure the connection between the DAC server and the DAC repository

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

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

- 3 Click Yes.

The Server Configuration dialog box appears.



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

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

- When you created a database to store the DAC repository (for more information, see [To create the required databases on page 91](#)).
- When you created a DAC connection (for more information, see [To create a DAC connection on page 91](#)).

Field	Description
Connection type	Select the type of database in which you are storing the DAC metadata repository. Depending on what type you select, the connection details below change (see <i>Connection fields</i> below).
<i>Connection fields</i> (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 an Instance (for example, mymachinename). ■ 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 an Connection string. ■ If you select MYSQL, you are prompted below for a Database name (for example, mydacdatabase).
Table owner name	The database table name or instance (for example, DAC).
Password	The database or instance password (for example, DAC).

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

Process of Configuring Email in the DAC Server (Optional)

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

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

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

To configure the email administrator account in the DAC server

- 1** Launch the DAC client.
- 2** From the menu bar, click Tools > DAC Server Management DAC Server Setup.
A confirmation dialog box asks you to confirm that you want to configure the DAC repository connection.
- 3** Click Yes.
The Server Configuration dialog box appears.
- 4** In the Email Configuration tab, enter the appropriate information, and click Save.

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

Process of Setting DAC Source System Parameters

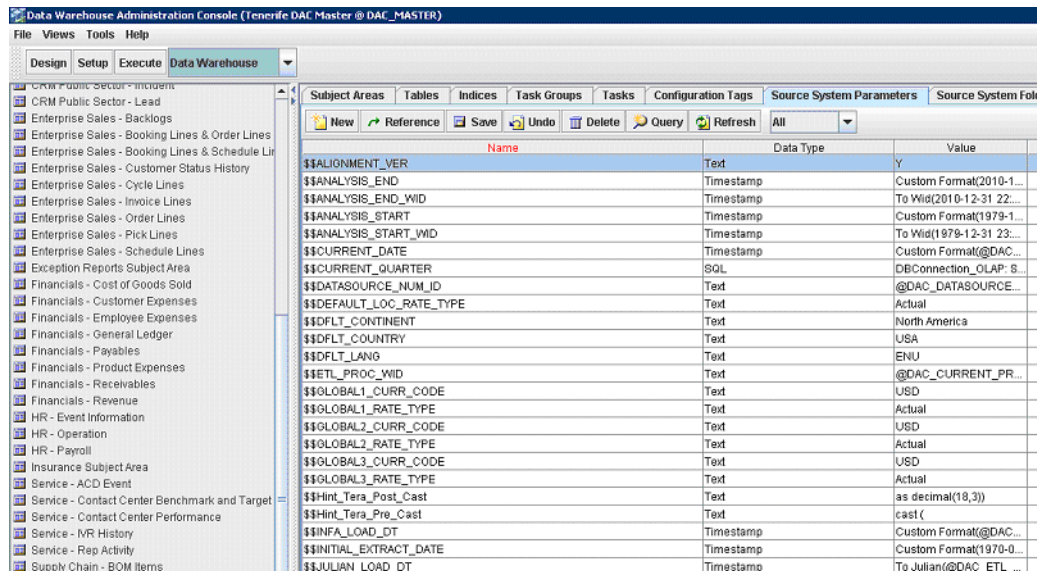
You set Source System Parameters (also known as DAC ETL Preferences) in the DAC to specify how the ETL routines process data for a container. For example, if operating in the United States, you might set the \$\$DLFT_COUNTRY to 'USA' to identify your data.

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

To set DAC source system parameters

- 1** Log into the DAC.
- 2** Display the Design view.
- 3** Make sure that you have selected the correct container from the *containers* drop down list.

4 Select the Source System Parameters tab.



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

6 Click Save.

The following table describes commonly available Source System Parameters.

Table 30. Commonly Available Source System Parameters

Parameter	Description	Parameter Name	Default
Analysis Start Date	The start date used to build the day dimension and to flatten exchange rates and costs lists.	\$\$ANALYSIS_START, \$\$ANALYSIS_START_WID	Jan 1, 1980
Analysis End Date	The end date used to build the day dimension and to flatten exchange rates and costs lists.	\$\$ANALYSIS_END, \$\$ANALYSIS_END_WID	Dec 31, 2010
Max Days in a SalesStage	Applicable to opportunities only. The maximum number of days an opportunity can spend in a sales stage.	\$\$MAX_DAYS_IN_STG	30

Table 30. Commonly Available Source System Parameters

Parameter	Description	Parameter Name	Default
Default Language	Siebel Specific. The language into which columns' display strings are translated.NOTE: Language-independent strings are stored in the _l columns. If you do not know the default language setting, issue the following query against the transactional database: select VAL from S_SYS_PREF where SYS_PREF_CD='ETL Default Language'; 12-MAL12D	\$\$DFLT_LANG	ENU
Alignment Version	Applicable to Siebel vertical applications only. Seed data needs to be updated in certain lists of values to make sure that the ETL process correctly populates the alignment type and product category relationships in the Oracle Business Analytics Warehouse.This setting is used by Siebel Life Sciences to control loading of changes in assignment rules in Siebel Assignment Manager.	\$\$ALIGNMENT_VER	Y
Maximum LOV Size	Siebel specific parameter. Numeric LOVs must have a maximum value for proper analysis and consistent data in the data warehouse. This value determines what the maximum value will be	\$\$LOV_MAX_VAL	9999999999
Minimum LOV Size	Siebel specific parameter. Numeric LOVs must have a minimum value for proper analysis and consistent data in the data warehouse. This value determines what the minimum value will be	\$\$LOV_MIN_VAL	0

Table 30. Commonly Available Source System Parameters

Parameter	Description	Parameter Name	Default
Type 2 Flag	Slow changing dimension (Type-2) specific parameter. If this dimension is “capable” of storing history, you can choose to set the flag value to be ‘Y’ or ‘N’. If set to ‘Y’ and you have already identified columns that your history tracking mechanism will be based upon, then this will trigger insertion of new records with new surrogate keys whenever anything history tracking event happens. If set to ‘N’, it just updates the existing record.	\$\$TYPE2_FLG	Y (This is just an out of box setting. The value of this parameter should be overridden at a task level, because not all dimensions will have the same setting.)
Update All History Flag	Applicable for slowly changing dimensions (Type-2). With a Type-2 change happening along with some other changes in a non-history tracking column, you can choose to update values of that non-Type2 column for all historical records. If so, set this flag to ‘Y’. If you want to preserve the old values for non-Type2 columns for older records, set this to ‘N’.	\$\$UPDATE_ALL_HISTORY	Y (Recommended approach and out of box setting).
Language Code	Datawarehouse language that is used to resolve NAMES for the CODES in your system.	\$\$LANGUAGE_CODE	E (out of box setting)
Global1 Currency Code	Generic parameter identifying the global (Data Warehouse) currency code. Three distinct global currencies are supported; this is the first of the three.	\$\$GLOBAL1_CURR_CODE	USD
Global2 Currency Code	Generic parameter identifying the global (Data Warehouse) currency code. Three distinct global currencies are supported; this is the second of the three.	\$\$GLOBAL2_CURR_CODE	USD

Table 30. Commonly Available Source System Parameters

Parameter	Description	Parameter Name	Default
Global3 Currency Code	Generic parameter identifying the global (Data Warehouse) currency code. Three distinct global currencies are supported; this is the third of the three.	\$\$GLOBAL3_CURR_CODE	USD
Global1 Exchange Rate Type	Generic parameter identifying the default exchange type to be used against the Global1 (Data Warehouse) currency code.	\$\$GLOBAL1_RATE_TYPE	Value of this parameter is source specific. For Siebel Sources, the default value is "Actual". In case of Oracle and Universal Adapters, the value is overridden with "Corporate".
Global2 Exchange Rate Type	Generic parameter identifying the default exchange type to be used against the Global2 (Data Warehouse) currency code.	\$\$GLOBAL2_RATE_TYPE	Value of this parameter is source specific. For Siebel Sources, the default value is "Actual". In case of Oracle and Universal Adapters, the value is overridden with "Corporate".
Global3 Exchange Rate Type	Generic parameter identifying the default exchange type to be used against the Global3 (Data Warehouse) currency code.	\$\$GLOBAL3_RATE_TYPE	Value of this parameter is source specific. For Siebel Sources, the default value is "Actual". In case of Oracle and Universal Adapters, the value is overridden with "Corporate".

Table 30. Commonly Available Source System Parameters

Parameter	Description	Parameter Name	Default
Default Local Exchange Rate Type	Generic parameter identifying the default exchange rate type to be used against the local (OLTP) currency code, in case the same is not available from the source.	\$\$DEFAULT_LOC_RATE_TYPE	Value of this parameter is source specific. For Siebel Sources, the default value is "Actual". In case of Oracle and Universal Adapters, the value is overridden with "Corporate".
Current Date	These are runtime parameters that gets the value of the current date and the current quarter based on the date dimension, W_DAY_D, against the Fiscal or Gregorian calender, as the case may be.	\$\$CURRENT_DATE	Value obtained through firing SQL at the runtime.
Current Quarter		\$\$CURRENT_QUARTER	Value obtained through firing SQL at the runtime.
Data Source Number Identifier	This is a static parameter used to identify the source system type and its instance. For example, you could set '2' for the first instance of Oracle 11.5.8 source, and '3' for the second instance of Oracle 11.5.8, '4' for the only instance of Oracle 11.5.10 and '5' for Siebel OLTP source, and so on. This is a runtime parameter that increments by 1 (normally) for each ETL run. For instance, for the first full load, it might start with a 1. Then the first incremental ETL run gets a 2 and so on. This helps in identifying records that were affected during a given ETL run.	\$\$DATASOURCE_NUM_ID	Value is different for different containers, for example 3 for Oracle 11.5.8 and so on.

Table 30. Commonly Available Source System Parameters

Parameter	Description	Parameter Name	Default
ETL process Identifier		\$\$ETL_PROC_WID	DAC generated. Normally starts with a 1 and then increments by 1.
Dates used for incremental extraction	This date is used to extract data for the full load.	\$\$INFA_LOAD_DT, \$\$INITIAL_EXTRACT_DATE	01/01/1970
Last Extract Date	This date is used to extract data during everyday incremental loads	\$\$LAST_EXTRACT_DATE\$\$JULIAN_LOAD_DT	Runtime, DAC generated. Based on when the last date/time extraction (ETL) happened.
ETL Load Date	The ETL load start Date	\$\$LOAD_DT	Runtime, DAC generated
Exception reporting for CODE-NAME pairs	This static parameter value is used to substitute the CODE part in (mostly) dimension tables when the master code (standardized code) was not found during code lookup against the supplied value of source side code.	\$\$MASTER_CODE_NOT_FOUND	
Exception reporting for CODE-NAME pairs	This static parameter value is used to substitute the CODE part in (mostly) dimension tables when the code lookup failed to resolve to the standardized code in the event where the source-side code itself was not supplied in the staging area tables.	\$\$SOURCE_CODE_NOT_SUPPLIED	<null value> out of the box. But can be any string like "master code is not found" up to a limit of 30 characters.
Tenant Identifier	This is a static parameter used to identify the tenant id, in multi-tenancy situations.	\$\$TENANT_ID	

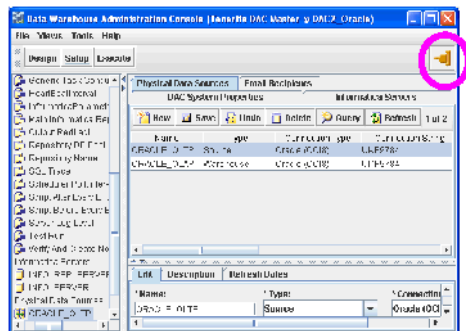
Process of Starting the DAC Server

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

To start the DAC Server

- 1** Choose Oracle Business Intelligence | Oracle DAC | Start DAC Server.

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



Starting the Informatica Server

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

To start the Informatica Server

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

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

Updating Siebel Transactional Database Schema Definitions

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

The process of updating schema definitions in the Siebel transactional database consists of the following tasks:

- Process of Applying Siebel (CRM) and Oracle E-Business Suite Schema Changes For Oracle, DB2/ UDB, MSSQL on page 139
- Process of Applying CRM Schema Changes For DB2 on 390 or z/os on page 140
- Process of Verifying the Siebel (CRM) and Oracle E-Business Suite Schema Changes on page 141
- About Delete Triggers on page 141

Process of Applying Siebel (CRM) and Oracle E-Business Suite Schema Changes For Oracle, DB2 / UDB, MSSQL

To enable change capture for Oracle's Siebel adaptors, you use the ddlimp control file to apply Siebel (CRM) and Oracle E-Business Suite schema changes, which updates the required image tables in the OLTP.

NOTE: Using the ddlimp control file to apply schema changes replaces the use of SIF files in Siebel Tools in previous product releases.

How to Apply Oracle's CRM Schema Changes For Oracle, DB2 / UDB, MSSQL

- 1 On the machine where Oracle BI Applications is installed, run the following command:

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

NOTE: If you are applying schema changes to a Siebel CRM Version 6.3, specify DDL_OLTP_63.CTL as the /F parameter instead of DDL_OLTP.CTL.

For example:

```
DDLIMP /U SADMIN /P SADMIN /C SIEBEL_OLTP /G SSE_ROLE /F  
<DRIVE>:\OracleBI\dwrep\DDL_OLTP.CTL /L <DRIVE>:\OracleBI\dwrep\DDL_OLTP.log
```

Notes:

- /P <PASSWORD> - The password for Oracle's CRM OLTP.
- /C <ODBC connect string> - The name of the ODBC connect string.
- For Oracle databases, use the Oracle Merant ODBC Drivers (installed with Oracle BI Applications).

- In addition, you can use the following commands:

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

/Z Y - (if the OLTP database is DB2 and Unicode or the OLTP database is MS SQL Server and Unicode).

/B <TABLE_SPACE_NAME> - If you want to create these table in a separate table space.

/X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate table space.

/Y - Storage File for DB2/390.

- 2 Restart all servers.

Process of Applying 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.
- %4kBufPool - 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 BI Applications is installed, run the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC_CSN> >>> /G SSE_ROLE  
/F \OracleBI\dwrep\DDL_OLTP_DB2390.CTL> /L <..\oracleBI\dwrep\DDL_OLTP.log> /5 Y /  
A <SCHEMA_OWNER> /Y ..\OracleBI\dwrep\Storeage_DDL_OLTP.CTL
```

Notes:

- /P <PASSWORD> - The password for Oracle's CRM OLTP.
- /C <ODBC connect string> - The name of the ODBC connect string.
- For Oracle databases, use the Oracle Merant ODBC Drivers.

- In addition, you can use the following commands:

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

/Z Y - (if the OLTP database is DB2 and Unicode or the OLTP database is MS SQL Server and Unicode).

/B <TABLE_SPACE_NAME> - If you want to create these table in a separate table space.

/X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate table space.

/Y - Storage File for DB2/390.

- 3 To create indices for the tables created in step 3, Run the following SQL Script from the DB2 command line utility connected to your OLTP Schema:

```
..\OracleBI\Dwrep\Create_OLTP_Db2390_index.sql
```

Process of Verifying the Siebel (CRM) and Oracle E-Business Suite Schema Changes

After applying Siebel (CRM) and Oracle E-Business Suite schema changes, you need to verify that appropriate tables were created in the transactional database.

To verify Siebel (CRM) and Oracle E-Business Suite schema changes

- 1 For Oracle's Siebel Business Applications, make sure that the following tables were created in the transactional database:

- S_ETL_R_IMG_1 through S_ETL_R_IMG_166
- S_ETL_I_IMG_1 through S_ETL_D_IMG_166
- S_ETL_D_IMG_1 through S_ETL_D_IMG_166
- S_ETL_PARAM
- S_ETL_PRD_ATTR
- S_ETL_PRD_REL

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.

Creating 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 [To create DB2 stored procedures on page 143](#)).

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 [To create DB2 stored procedures on page 143](#)).

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 [Creating Stored Procedures for DB2-UDB on page 142](#)).

- 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.

Teradata-Specific Installation and Configuration Checklist (Windows)

Table 31 provides a list of Teradata-specific installation and configuration steps that are performed during the Oracle BI Applications installation and configuration process. You should review this list to make sure that you have performed all of the required Teradata-specific steps.

Table 31. Teradata-Specific Installation Checklist for Windows

Action	Link to Topic
Review Teradata-specific database guidelines.	Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse on page 47.
Activate join indices.	Activating Join Indices for Teradata Databases on page 104.
Create or drop data warehouse tables.	Process of Creating Data Warehouse Tables on a Teradata Database on page 109.
Use recommended name for Informatica repository.	Setting Up the Informatica Server and Informatica Repository Server on page 79.
Create or restore the Informatica repository using the Informatica Repository Manager. You cannot create or restore the Informatica repository using the DAC.	Process of Registering the Informatica Repository Server on page 114.

Table 31. Teradata-Specific Installation Checklist for Windows

Action	Link to Topic
Configure the source and data warehouse database relational connections in the Informatica Workflow Manager. You cannot use the DAC's Data Warehouse Configurator to configure relational connections.	Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse on page 47.
Set the Code Page File Property for Unicode Environments.	Setting the Code Page File Property for Unicode Environments on Teradata Databases on page 124.
Set up the HOSTS file.	Setting Up the HOSTS File for Teradata Installations on Windows on page 124.

6

Installing and Configuring Oracle BI Applications on UNIX

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

- [Overview of Installing and Configuring Oracle BI Applications on UNIX on page 145](#)
- [Installation and Configuration Process Task List for UNIX on page 146](#)
- [Installing Oracle BI Applications on UNIX on page 150](#)
- [Installing and Configuring the Informatica Repository Server on UNIX on page 151](#)
- [Installing and Configuring the Informatica Server on UNIX on page 153](#)
- [Updating the Informatica License Files on UNIX on page 160](#)
- [Installing the Informatica Patch on UNIX on page 161](#)
- [Creating Database Connections for Teradata Installations on page 161](#)
- [Setting Environment Variables for the Informatica Server on UNIX on page 162](#)
- [Setting Up the HOSTS File for Teradata Installations on UNIX on page 163](#)
- [Starting the Informatica Repository Server on UNIX on page 163](#)
- [Stopping the Informatica Repository Server on UNIX on page 164](#)
- [Configuring the DAC Server on UNIX on page 164](#)
- [Starting the Informatica Server on UNIX on page 167](#)
- [Stopping the Informatica Server on UNIX on page 168](#)
- [Starting the DAC Server on UNIX on page 168](#)
- [Stopping the DAC Server on UNIX on page 169](#)
- [Teradata-Specific Installation and Configuration Checklist for UNIX on page 169](#)

NOTE: If you are installing Oracle Business Intelligence Applications in a UNIX environment, you must use the Unicode character data mode. For more information about Unicode, see [Deploying Oracle Business Analytics Warehouse in Latin-1 General, Unicode and Non-English Environments on page 53](#)

Overview of Installing and Configuring Oracle BI Applications on UNIX

Before you begin the installation and configuration process on UNIX, you should review the following:

- [Chapter 4, "Preinstallation and Predeployment Considerations for Oracle BI Applications"](#)
- [Chapter 5, "Installing and Configuring Oracle BI Applications on Windows"](#)

When you install Oracle BI Applications in a UNIX environment, note the following:

- Some of the components that make up Oracle BI Applications can only be installed on Windows. For example, the Informatica client tools, and the DAC client tool. For more information, see [Figure 4](#).
- Although you can install Informatica server components and DAC server components on UNIX machines, Oracle does not provide a separate Oracle BI Applications installer for UNIX. Therefore, to install these component in a UNIX environment, you must first install Oracle BI Applications and the Oracle BI infrastructure onto a Windows machine, then manually copy over the installation directories to a UNIX machine. For more information, see [Installation and Configuration Process Task List for UNIX on page 146](#).
- If your source or target database is Microsoft SQL Server, you will need to use DataDirect ODBC driver for Microsoft SQL Server supplied by Informatica. This can be downloaded directly from Informatica Web site or you can contact Oracle Support for further assistance.

[Figure 4](#) shows a typical deployment of Oracle BI Applications in a UNIX installation. Note that the Informatica client tools and the DAC client are installed on Windows only. You will complete some of the configuration tasks in the Windows environment. [Table 32](#) lists each task in the installation and configuration process and tells you in which environment to complete the task.

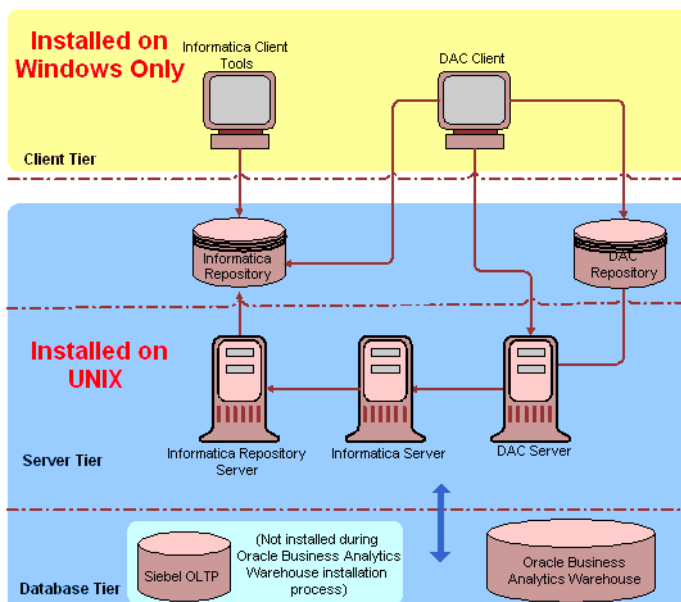


Figure 4. UNIX Installation of Oracle Business Analytics Warehouse

Installation and Configuration Process Task List for UNIX

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

Table 32. UNIX Installation and Configuration Process

Task	Instructions	Environment
1 Install Oracle BI Applications on Windows .	Running the Oracle Business Intelligence Applications Installation Wizard (Windows) on page 72	Windows
2 Install Oracle BI Applications on UNIX by copying across the installation files from a Windows installation directory.	Installing Oracle BI Applications on UNIX on page 150	UNIX
3 Download and copy Hibernate libraries. NOTE: Hibernate libraries must be copied on to each DAC Client machine (Windows only) and DAC Server machine (Windows or UNIX).	Process of Copying Hibernate Libraries on page 88	■ Windows (DAC client) ■ UNIX (DAC server)
4 Install and Configure the Informatica Repository Server on UNIX (using the Informatica Installer).	Installing and Configuring the Informatica Repository Server on UNIX on page 151.	UNIX
5 Install and Configure the Informatica Server on UNIX (using the Informatica Installer).	Installing and Configuring the Informatica Server on UNIX on page 153.	UNIX
6 Update the Informatica license files.	Updating the Informatica License Files on UNIX on page 160.	UNIX
7 Install the Informatica patch.	Installing the Informatica Patch on UNIX on page 161	UNIX
8 Set Informatica Data Code Pages Validation.	Setting Informatica Data Code Pages Validation on page 88.	Windows
9 Install the Java SDK and JDBC drivers on the DAC client machine and the DAC server machine.	Configuring the DAC Client on page 89.	■ Windows (DAC client) ■ UNIX (DAC server)
10 Configure the DAC config.bat file.	Process of Configuring the DAC Config.bat File on page 89.	Windows
11 Create the DAC Metadata Repository (using the DAC client).	Creating the DAC Metadata Repository on page 91.	Windows

Table 32. UNIX Installation and Configuration Process

Task	Instructions	Environment
12 Configure the DAC Metadata Repository (using the DAC client and ODBC Connections software).	Setting Up DAC System Properties on page 96.	Windows
	Process of Creating ODBC Database Connections on page 85.	Windows
	Specifying Transactional and Data Warehouse Data Sources in the DAC on page 100	Windows
	Creating Database Connections for Teradata Installations on page 161.	UNIX
	Configuring Email Recipients in the DAC Client on page 103.	Windows
	Configuring Email Recipients in the DAC Client on page 103.	Windows
	Activating Join Indices for Teradata Databases on page 104.	Windows
13 Configure the Oracle Business Analytics Warehouse (using the DAC client).	Process of Creating the SSE Role on page 105.	Windows
	Process of Creating Data Warehouse Tables on page 105.	Windows
	Process of Creating Data Warehouse Tables on a Teradata Database on page 109.	Windows
14 Set environment variables for the Informatica Server.	Setting Environment Variables for the Informatica Server on UNIX on page 162.	UNIX
15 Set the Windows PATH environment variable for the Informatica Server.	Setting Environment Variables on Windows on page 86.	Windows
16 Start the Informatica Repository Server.	Starting the Informatica Repository Server on UNIX on page 163.	UNIX

Table 32. UNIX Installation and Configuration Process

Task	Instructions	Environment
17 Create or restore and configure the Informatica repository (using the DAC).	Process of Backing Up and Restoring the Informatica Repository (Optional) on page 122.	Windows
	See also Process of Stopping and Starting the Informatica Repository (Optional) on page 121.	
	Setting Up the Informatica Server and Informatica Repository Server on page 79	
	Process of Registering the Informatica Repository Server on page 114.	Windows
	Process of Stopping and Starting the Informatica Repository (Optional) on page 121.	Windows
	Process of Registering the Informatica Repository Server on page 114.	Windows
	Process of Loading the Pre-Built Repository Into Informatica on page 115.	Windows
	Process of Configuring the Informatica Repository in Workflow Manager on page 118	Windows
18 Set the Code Page File Property for Unicode Environments on Teradata Databases.	Setting the Code Page File Property for Unicode Environments on Teradata Databases on page 124	Windows
19 Set Up the HOSTS File for Teradata Installations.	Setting Up the HOSTS File for Teradata Installations on UNIX on page 163.	UNIX
20 Verify the Informatica Repository configuration (using Informatica tools).	Verifying the Informatica Repository Configuration on page 125.	Windows
21 Start the Informatica Server.	Starting the Informatica Server on UNIX on page 167.	UNIX

Table 32. UNIX Installation and Configuration Process

Task	Instructions	Environment
22 Set the DAC ETL Preferences (using the DAC client).	Starting the Informatica Server on page 138.	Windows
23 Configure the DAC server. NOTE: Before you start, make sure that you have copied over the \OracleBI\DAC\ folder from the Windows machine where you installed Oracle BI Applications to a UNIX machine.	Configuring the DAC Server on UNIX on page 164.	UNIX
24 Update the transactional database schema definition.	Updating Siebel Transactional Database Schema Definitions on page 138	Windows

Installing Oracle BI Applications on UNIX

Before you install Oracle Business Intelligence Applications on UNIX you must first complete the following installations:

- Install the Oracle Business Intelligence infrastructure on Windows.
For instructions, see *Oracle Business Intelligence Infrastructure Installation and Configuration Guide*.
- Install the Oracle Business Intelligence infrastructure on UNIX.
For instructions, see *Oracle Business Intelligence Infrastructure Installation and Configuration Guide*.
- Install Oracle Business Intelligence Applications on Windows.
For instructions, see [Installing and Configuring Oracle BI Applications on Windows on page 69](#).

To install Oracle BI Applications on UNIX

- 1** Copy the folder OracleBI\DAC\ from the Oracle Business Intelligence Applications installation directory in the Windows environment to the UNIX environment.

NOTE: You also need to copy the OracleBI\DAC\ folder onto any machine on which you want to run the DAC Server (for more information, see [Configuring the DAC Server on UNIX on page 164](#)).
- 2** Copy the folder OracleBI\dwrep\Informatica from the Oracle Business Intelligence Applications installation directory in the Windows environment to the UNIX environment. Use the same directory structure in the UNIX environment as the Windows environment.

This action copies pmrep, the Informatica repository file (Oracle_BI_DW_Base.rep), and the parameter files to the UNIX environment.

- 3 Copy the following files from the Windows environment into the UNIX environment. Use the same directory structure in the UNIX environment as the Windows environment.

File Type	File Name	File Location
DAC repository	*.*	OracleBI\DAC\export
DAC application related	*.*	OracleBI\DAC\CustomSQLs
Delete trigger	DeleteTriggers.list	OracleBI\DAC\conf
Analytics repository	OracleBIAnalyticsApps.rpd	OracleBI\Server\Repository
Presentation Catalog	*.*	OracleBIData\Web\Catalog
		NOTE: You need to un-zip the ZIP files and copy the un-zipped files to the Unix machine.
Language	*.*	OracleBIData\Web\Res
Informatica seed files and lookup files	*.*	OracleBI\dwrep\Informatica\SrcFiles and OracleBI\dwrep\Informatica\LkpFiles
		NOTE: You need to copy these files to the location specified as \$PMSourceFileDir in Informatica Workflow Manager.

Installing and Configuring the Informatica Repository Server on UNIX

If you are using a non-ENU operating system, make sure you have all the required locale files installed.

Install the Informatica software directly from the CD-ROM. If you cannot install the software directly from the CD-ROM, you must use FTP in binary mode; otherwise, the installation will fail. After the FTP process, use the chmod command to change the permission setting for all install files to read, write, and execute.

To install and configure the Informatica Repository Server

- 1 Log onto the UNIX machine.
- 2 Change to the appropriate directory and run the installer by typing ./install.
- 3 Select the language in which you want to perform the installation.
- 4 Enter the Product license key.
- 5 Accept the license agreement.

- 6 Select Informatica Repository Server from the list of components available to install.
- 7 Specify the directory where the Repository Server will be installed.
- 8 Enter the appropriate values for the options listed below.

Option	Required / Optional	Description
Configuration file name	Required	The name of the Repository Server configuration file.
ServerPort	Required	The port number the Repository Server uses for repository client connections. The default is 5001.
AdminPasswd	Required	The administrative password used to access the Repository Server. It is recommended that you use Administrator as the password.
RaMinPort	Required	The minimum port number the Repository Server can assign to the Repository Agent process. The default is 5002.
RaMaxPort	Required	The maximum port number the Repository Server can assign to the Repository Agent process. The default is 65535.
ConfigDir	Required	The name of the directory in which the Repository Server stores repository configuration information files. You can specify either a relative path or an absolute path. The default is Config.
BackupDir	Required	The name of the directory in which the Repository Server stores repository backup files. You can specify either a relative path or an absolute path. The default is Backup.
PluginDir	Required	The name of the directory in which the Repository Server stores repository plugin files. You can specify either a relative path or an absolute path. The default is Plugin.

Option	Required / Optional	Description
ErrorSeverityLevel	Required	The level of error messages written to the Repository Server log. Specify one of the following message levels: ■ Error. Writes ERROR code messages to the log. ■ Warning. Writes WARNING and ERROR code messages to the log. ■ Information. Writes INFO, WARNING, and ERROR code messages to the log. ■ Tracing. Writes TRACE, INFO, WARNING, and ERROR code messages to the log. ■ Debug. Writes DEBUG, TRACE, INFO, WARNING, and ERROR code messages to the log. The default is Information
LogFileName	Required	The name of the Repository Server log file. The default is pmrepserver.log.

9 Type Y if you want to view the readme file, or N to read it later.

10 Press enter to continue to install the Informatica Server.

Do not exit from the installer. Proceed to [“Installing and Configuring the Informatica Server on UNIX” on page 153](#).

Installing and Configuring the Informatica Server on UNIX

The instructions in this section for installing and configuring the Informatica Server are meant to be followed immediately after the Informatica Repository Server installation, without exiting the installer. If you exited the installer after installing the Repository Server, you may want to review [“Installing and Configuring the Informatica Repository Server on UNIX” on page 151](#).

NOTE: You can install and configure two Informatica Server instances on the same UNIX machine. This configuration requires two copies of pmserver.cfg and pmrepserver.cfg files. Each file must have a unique name, for example, pmserverdev.cfg and pmserverQA.cfg. Each config file will define unique ports for each instance of the Informatica Server.

NOTE: Even if you run the Informatica server from a different machine to the DAC server machine, you still must install Informatica on the DAC server machine, because the DAC server requires the Informatica libraries (utility pmcmd).

To install and configure the Informatica Server

- 1 Select Informatica PowerCenter Server from the list of components available to install.
- 2 Specify the directory where the Informatica Server will be installed.
- 3 Enter the appropriate values for the options listed below.

Option	Required / Optional	Description
Configuration file name	Required	The name of the Informatica Server configuration file.
ServerName	Required	The name of the Informatica Server registered with the repository. It is recommended that you use ORACLE_BI_DW_SERVER.
RepositoryName	Required	The repository name entered when creating or upgrading a repository. It is recommended that you use Oracle_BI_DW_Base.
PMUser	Required	The repository user name used to connect to the repository. It is recommended that you use Administrator
PMPassword	Required	The password corresponding to the repository user (PMUser). It is recommended that you use Administrator.
RepServerHostName	Required	The host name of the machine hosting the Repository Server.
RepServerPortNumber	Required	The port number the Repository Server uses to communicate with repository client applications.
RepServerTimeout	Required	The maximum number of seconds the Informatica Server tries to establish a connection to the Repository Server.
LogFileName	Optional	The filename of the Informatica Server log file.
DataMovementMode	Required	Possible values are ASCII and UNICODE. The default data movement mode is ASCII, which passes 7-bit ASCII character data. To pass 8-bit ASCII and multibyte character data from sources to targets, use the Unicode mode. NOTE: If you use the UNICODE mode, you should set the ValidateDataCodePages option as No and the SessionLogInUTF8 option as Yes.

Option	Required / Optional	Description
ValidateDataCodePages	Required	If set to Yes, the Informatica Server enforces data code page compatibility. If set to No, the Informatica Server lifts restrictions for source and target data code page selection, stored procedure and lookup database code page selection, and session sort order selection. NOTE: If you set the <code>DataMovementMode</code> option to <code>UNICODE</code>, you should set the <code>ValidateDataCodePages</code> option as No.
SessionLogInUTF8	Required	If set to Yes, the Informatica Server writes to the session log using the UTF-8 character set. If set to No, the Informatica Server writes to the session log using the Informatica Server code page. NOTE: If you set the <code>DataMovementMode</code> option to <code>UNICODE</code>, you should set the <code>SessionLogInUTF8</code> option as Yes.
MaxSessions	Required	The maximum number of sessions stored in shared memory. This sets the maximum number of sessions the Informatica Server can run in parallel.
LMSharedMem	Required	The amount of shared memory available for use by the Informatica Server Load Manager process. For every 10 sessions in <code>MaxSessions</code> , you need at least 2,000,000 bytes reserved in <code>LMSharedMem</code> . The default is 2,000,000 bytes.
TimeStampWorkflowLogMessages	Optional	Set to Yes if you want to append a time stamp to messages written to the workflow log.
FailSessionIfMaxSessionsReached	Optional	Set to Yes to have the Informatica Server fail the session if the number of sessions already running is equal to the value configured for <code>MaxSessions</code> . If you set this option to No, the Informatica Server places the session in a ready queue until a session slot becomes available.
ExportSessionLogLibName	Optional	If you want the Informatica Server to write session log messages to an external library, enter the name of the library file.

Option	Required / Optional	Description
ErrorSeverityLevel	Optional	The level of error messages written to the Informatica Server log. Specify one of the following message levels: ■ Error. Writes ERROR code messages to the log. ■ Warning. Writes WARNING and ERROR code messages to the log. ■ Information. Writes INFO, WARNING, and ERROR code messages to the log. ■ Tracing. Writes TRACE, INFO, WARNING, and ERROR code messages to the log. ■ Debug. Writes DEBUG, TRACE, INFO, WARNING, and ERROR code messages to the log.
DateHandling40 Compatibility	Optional	Set to Yes if you want the Informatica Server to handle dates as in PowerCenter 1.0/PowerMart 4.0. Set to No if you want the Informatica Server to handle dates as defined in the current version of PowerCenter. Date handling significantly improved in PowerCenter 1.5 and PowerMart 4.5. If you need to revert to PowerCenter 1.0 or PowerMart 4.0 behavior, you can configure the Informatica Server to handle dates as in PowerCenter 1.0 and PowerMart 4.0.
PMServer3X Compatibility	Optional	Set to Yes if you want the Informatica Server to handle dates as in PowerCenter 1.0/PowerMart 4.0. Set to No if you want the Informatica Server to handle dates as defined in the current version of PowerCenter. Date handling significantly improved in PowerCenter 1.5 and PowerMart 4.5. If you need to revert to PowerCenter 1.0 or PowerMart 4.0 behavior, you can configure the Informatica Server to handle dates as in PowerCenter 1.0 and PowerMart 4.0.
AggregateTreatNull AsZero	Optional	Set to 1 to have the Informatica Server treat nulls as zero in Aggregator transformations. Set to 0 to have the Informatica Server treat nulls as nulls in aggregate calculations.

Option	Required / Optional	Description
AggregateTreatRowAsInsert	Optional	Set to 1 to have the Informatica Server perform aggregate calculations before flagging records for insert, update, delete, or reject in Update Strategy expressions. Set to 0 to have the Informatica Server perform aggregate calculations based on the Update Strategy transformation.
DateDisplayFormat	Required	If specified, the Informatica Server validates the date display format and uses it in session log and server log entries. If the date display format is invalid, the Informatica Server uses the default date display format. The default date display format is DY MON DD HH 24:MI:SS YYYY.
JoinerSourceOrder6xCompatibility	Optional	If set to Yes, the Informatica Server processes master and detail pipelines sequentially as it did in versions prior to 7.0. The Informatica Server processes all data from the master pipeline before starting to process the detail pipeline. Also, if you enable this option, you cannot specify the Transaction level transformation scope for Joiner transformations. If set to No, the Informatica Server processes the master and detail pipelines concurrently.
TreatNullInComparisonOperatorAs	Required	Determines how the Informatica Server evaluates null values in comparison operations. Specify one of the following options: ■ Null. The Informatica Server evaluates null values as null in comparison expressions. If either operand is null, the result is null. This is the default behavior. ■ High. The Informatica Server evaluates null values as greater than non-null values in comparison expressions. If both operands are null, the Informatica Server evaluates them as equal. When you choose High, comparison expressions never result in null. ■ Low. The Informatica Server evaluates null values as less than nonnull values in comparison expressions. If both operands are null, the Informatica Server treats them as equal. When you choose Low, comparison expressions never result in null.

Option	Required / Optional	Description
WriterWaitTimeOut	Optional	In target-based commit mode, the amount of time in seconds the writer remains idle before it issues a commit when the following conditions are true: ■ The Informatica Server has written data to the target. ■ The Informatica Server has not issued a committed. The Informatica Server may commit to the target before or after the configured commit interval. Default is 60 seconds. If you configure the timeout to be 0 or a negative number, the Informatica Server defaults to 60 seconds.
LoadManagerAllow Debugging	Optional	If set to Yes, you can run the Debugger.
CreateIndicatorFiles	Optional	If set to Yes, the Informatica Server creates indicator files when you run a workflow with a flat file target.
XMLWarnDupRows	Optional	If set to Yes, the Informatica Server writes duplicate row warnings and duplicate rows for XML targets to the session log.
OutputMetaData ForFF	Optional	If you specify this option, the Informatica Server writes column headers to flat file targets. It writes the target definition port names to the flat file target in the first line, starting with the # symbol. By default, this option is disabled.
MaxLookupSPDB Connections	Optional	Allows you to specify a maximum number of connections to a lookup or stored procedure database when you start a session. If the number of connections needed exceeds this value, session threads must share connections. This can result in a performance loss. If you do not specify a value, the Informatica Server allows an unlimited number of connections to the lookup or stored procedure database. If the Informatica Server allows an unlimited number of connections, but the database user does not have permission for the number of connections required by the session, the session fails.

Option	Required / Optional	Description
MaxMSSQL Connections	Optional	Allows you to specify a maximum number of connections to a MSSQL Server database when you start a session. If the number of connections required by the session is greater than this value, the session fails.
NumOfDeadlock Retries	Optional	Allows you to specify the number of times the Informatica Server retries a target write on a database deadlock.
DeadlockSleep	Optional	Allows you to specify the number of seconds before the Informatica Server retries a target write on database deadlock. If set to 0 seconds, the Informatica Server retries the target write immediately.
LicenseFile	Required	The name of the file that contains the license keys. The default file name is pm.lic. It is recommended that you use this file name. During the installation, the product license key is automatically added to this file. After the installation, you must add any additional Option and Connectivity license keys to this file. For instructions, see Updating the Informatica License Files on UNIX on page 160.
JVMDIIPath	Optional	The absolute path to the JVM library file.
JVMClassPath	Optional	You can set the CLASSPATH to any JAR files you need to run a session using a web service source, target, or transformation. The Informatica Server appends the values you set to the system CLASSPATH.
JVMMinMemory	Optional	Set this value to increase the minimum amount of memory for JVM to use during an Informatica Server session. The default value is 32 MB. If the session fails due to a lack of memory, you may want to increase this value.
JVMMaxMemory	Optional	Set this value to increase the maximum amount of memory for JVM to use during an Informatica Server session. The default value is 64 MB. If the session fails due to a lack of memory, you may want to increase this value.

Option	Required / Optional	Description
TrustStore	Optional	Enter the value for TrustStore using the following syntax: <path>/<filename>
ClientStore	Optional	Enter the value for ClientStore using the following syntax: <path>/<filename>
JvmOptionName[n]	Optional	You can enter names and values for up to 10 custom VM options. Use the following syntax: JvmOption[n]=-D<VMOption_prefix>=<VMOption_value> n can range from 1 to 10 and must be unique for each option.
JCEProvider	Optional	Enter the JCEProvider class name to support NTLM authentication.
HttpProxyServer	Required	Name of the HTTP proxy server.
HttpProxyPort	Required	Port number of the HTTP proxy server. This must be a number.
HttpProxyUser	Optional	Authenticated username for the HTTP proxy server.
HttpProxyPassword	Optional	Password for the authenticated user.
HttpProxyDomain	Optional	Domain for authentication.

- 4 When the installation is complete, proceed to ["Updating the Informatica License Files on UNIX" on page 160.](#)

Updating the Informatica License Files on UNIX

When you install the Informatica PowerCenter, you enter a product license key. The installation process adds the product license key to a license file called pm.lic. Copies of this license file are stored in the installation directories for the Informatica Server and the Repository Server. You must add to the license file any additional Option and Connectivity license keys you have.

To update the Informatica license files

- 1 Navigate to the Informatica Server installation directory.

- 2 Update the license file with a license key using the following syntax:

```
pmlic update -key <keystring_to_add>
```

For example, if the license key is AAAA-AAAA-AAAA-AAAA-AAAA, enter the following:

```
pmlic update -key AAAA-AAAA-AAAA-AAAA-AAAA
```

- 3 Repeat [Step 2](#) for every license key you need to add.
- 4 Navigate to the Repository Server installation directory, and repeat [Step 2](#) for every license key you need to add.

NOTE: To execute pmlic, you must set the following environment variables: LD_LIBRARY_PATH = <InformaticaInstallroot>/server PATH=<InformaticaInstallroot>/server

Installing the Informatica Patch on UNIX

After you install Informatica PowerCenter version 7.1.4, you need to install an Informatica patch.

To install the Informatica patch

- 1 Access the patch files on the Oracle BI Applications CD-ROM.
- 2 Locate the folder named Patch1, and install the patch by following the instructions in the readme file.

Creating Database Connections for Teradata Installations

If you are using a Teradata database, follow this procedure to create database connections.

To create database connections for Teradata installations

- 1 Using the Teradata ODBC driver, create an ODBC connection for the Teradata database.
- 2 Set the following variables in the ODBC.INI file:

```
DateFormat=AAA
```

```
SessionMode=ANSI
```

```
NoScan=Yes
```

For UNICODE environments, add the following:

```
CharacterSet=UTF8
```

Setting Environment Variables for the Informatica Server on UNIX

This section includes the following procedures:

- To add `setenv` variables to `.cshrc` on page 162
- To set the code page for non-`ENU` operating systems on UNIX on page 162
- To set Siebel UnicodeDB on the Informatica Server if source and target databases are Unicode on UNIX on page 163
- To disable the bulk loader for DB2/390 databases on page 163

To add `setenv` variables to `.cshrc`

Add `setenv` variables to `.cshrc`. Otherwise, you have to set the environment variables every time you log into the UNIX machine before configuring, starting, or stopping the Informatica servers.

- For Solaris, in the root directory, add these command lines to `.cshrc`:

```
setenv PATH /<INFA_home>/informatica/pm:.${PATH}

setenv LD_LIBRARY_PATH /<INFA_home>/informatica/pm:${LD_LIBRARY_PATH}

setenv PATH /<INFA_home>/informatica/repositoryserver:.${PATH}

setenv LD_LIBRARY_PATH /<INFA_home>/informatica/
repositoryserver:${LD_LIBRARY_PATH}
```

- For AIX, in the root directory, add these command lines to `.cshrc`:

```
setenv PATH ${PATH}:/<INFA_home>/informatica/pc:.

setenv LIBPATH ${LIBPATH}:/<INFA_home>/informatica/pc

setenv PATH ${PATH}:/<INFA_home>/informatica/repositoryserver:.

setenv LIBPATH ${LIBPATH}:/<INFA_home>/informatica/repositoryserver
```

- For HP-UX, in the root directory, add these command lines to `.cshrc`:

```
setenv PATH /<INFA_home>/informatica/pm:.${PATH}

setenv SHLIB_PATH /<INFA_home>/informatica/pm:${SHLIB_PATH}

setenv PATH /<INFA_home>/informatica/repositoryserver:.${PATH}

setenv SHLIB_PATH /<INFA_home>/informatica/repositoryserver:${SHLIB_PATH}
```

To set the code page for non-`ENU` operating systems on UNIX

- If you are using a non-`ENU` operating system, you must set the environment variable `PMREPCODEPAGE` appropriately. For example: `PMREPCODEPAGE=MS932`.

To set Siebel UnicodeDB on the Informatica Server if source and target databases are Unicode on UNIX

- For all UNIX platforms, if both the transactional and data warehouse data sources are Unicode, add the following line in /<INFA_home>/informatica/pm/pmsserver.cfg file.

```
SiebelUnicodeDB=<OLTP_TB0>@<OLTP_connectionString>  
<OLAP_TB0>@<OLAP_ConnectionString>
```

For example:

```
SiebelUnicodeDB=ora16221@sdcd1580i016_qa16b ora16222@sdcd1580i016_qa16b
```

To disable the bulk loader for DB2/390 databases

- For DB2/390 databases, add the following line in /<INFA_home>/informatica/pm/pmsserver.cfg file.

```
DisableDB2BulkMode = Yes
```

Setting Up the HOSTS File for Teradata Installations on UNIX

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

To set up the HOSTS file for Teradata installations

- 1 On the machine where the Informatica Server is installed, go to the /etc directory and open the HOSTS file.
- 2 In the HOSTS file, enter a line in the following format:

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

For example:

```
172.20.176.208 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.

Starting the Informatica Repository Server on UNIX

Follow this procedure to start the Informatica Repository Server on UNIX.

To start the Informatica Repository Server

- Go to /<INFA_home>/informatica/repositoryserver and enter `./pmrepserver`.

Check the log file to verify the server started. If it started correctly, the log file will state: "Server Initialization Completed." If multiple instances are running, you need to specify the name of the configuration file (for example, enter `./pmserver pmserverdev.cfg` or `./pmserver mserverQA.cfg`).

Stopping the Informatica Repository Server on UNIX

Follow this procedure to stop the Informatica Repository Server on UNIX.

To stop the Informatica Repository Server

- Go to /<INFA_home>/informatica/repositoryserver, and enter the following command:

```
pmrepserver -s -h <hostname>:<port number>
```

Check the log file to verify the server stopped.

Configuring the DAC Server on UNIX

Although you can run the DAC server on a UNIX machine, Oracle does not provide a separate Oracle BI Applications installer for UNIX. Therefore, to install the DAC server in a UNIX environment, you must first install Oracle BI Applications onto a Windows machine, then manually copy over the \OracleBI\DAC\ directory to a UNIX machine, and then follow the steps below (see [To configure the DAC server on UNIX on page 165](#)).

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 33 lists the available shell scripts and their usage. These files contain comments that provide information about how to configure the scripts.

Table 33. Oracle Business Analytics Warehouse Scripts for Bash and C Shells

Script	Purpose
config.sh config.csh	Configures the environment variables for DAC_HOME and JAVA_HOME.
dacCmdLine.sh dacCmdLine.csh	Invokes DAC commands on the DAC server. See the file for usage information.
dac_env.sh dac_env.csh	Configures environment variables, including Informatica's pmcmd utility, which enables the DAC server to invoke workflows. Other variables included are NLS_LANG and ORA_HOME.
serverSetupPrompt.sh serverSetupPrompt.csh	Configures DAC metadata repository connection information.
serverinfo.sh serverinfo.csh	Displays the DAC server version.
startserver.sh startserver.csh	Starts the DAC server on machines running Solaris and HP.
startserver_aix.sh startserver_aix.csh	Starts the DAC server on AIX machines.
stopserver.sh stopserver.csh	Shuts down the DAC server.

NOTE: All the files listed in Table 33 need to have read, write, and execute permissions. If the files do not have these permissions, modify them using the chmod command.

To configure the DAC server on UNIX

- 1 Copy the \OracleBI\DAC\ folder from the Windows machine where you installed Oracle BI Applications to a UNIX machine.

NOTE: You can use FTP to copy the shell scripts under DAC using a dos2unix utility in an ascii mode. The other files can be copied in a binary mode.

- 2 Using a MS-DOS to UNIX conversion tool, convert the script files in [Table 33](#) above to UNIX format (that is, remove the carriage return and line feed characters).

There are many MS-DOS to UNIX conversion tools that are freely available for download on the Internet. Alternatively, you can manually remove the carriage return and line feed characters from the script files.

- 3 Edit the `dac_env.sh` or `dac_env.csh` file to add the path of the directories for the Informatica Server and Informatica Repository Server to the appropriate environment variable:

- For Solaris: `LD_LIBRARY_PATH`
- For AIX: `LIBPATH`
- For HP: `SHLIB_PATH`

For example:

```
setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:/<INFA_HOME>/informatica<version>
```

- 4 If you are using an Oracle database and the OCI driver, configure the following variables and uncomment in the `dac_env.sh` or `dac_env.csh` file them by removing the `#` sign at the beginning of the line:

- `TNS_ADMIN`: To set the directory for Oracle to use `tnsnames.ora` files.
- `LD_LIBRARY_PATH` (Solaris), `LIBPATH` (AIX), or `SHLIB_PATH` (HP): To set the Oracle library path.

NOTE: Informatica does not support 64-bit libraries.

- 5 Edit the `config.sh` or `config.csh` file to set the `JAVA_HOME` directory.

CAUTION: Do not edit any other parameters in this file.

- 6 Copy the database-specific `.jar` files to the `$DAC/lib` directory:

- **Oracle.** If you are using an Oracle database (other than 8.x), find the directory where Oracle is installed. Copy the file named `ojdbc14.jar` in the `jdbc/lib` directory and paste it in the `$DAC/lib` directory.

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

```
setenv ORACLELIB=$DAC_HOME/lib/classes12.zip
```

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

NOTE: If your source or target database is DB2-UDB, you also need to create stored procedures to be used during the ETL process. For instructions, see [Creating Stored Procedures for DB2-UDB on page 142](#).

- **MSSQL.** If you are using an MSSQL database, download the SQL Server JDBC drivers from Microsoft's Web site. Copy the files `msbase.jar`, `mssqlserver.jar`, and `msutil.jar` to the `$DAC/lib` directory.

- 7 Run `serverSetupPrompt.sh` or `serverSetupPrompt.csh` to set the repository connection information and the email settings:
 - a At the initial prompt, enter 1 to set the repository connection information.
 - b Enter 3 to test the repository connection.
 - c Enter 4 to save the information you entered.
 - d Enter 2 to set up the administrator email account.

NOTE: For the email functionality to work, you must be using an SMTP server in which the SMTP authentication mode LOGIN is enabled. For example, if you are using Microsoft Exchange Server, you must enable the SMTP module and enable authentication in the Basic Mode. The SMTP server outbound email authentication must be turned on for the DAC server to be able to send email notifications.
 - e Enter 5 to exit.
- 8 Copy content-specific XML files from the `OracleBI\DAC\CustomSQLs` folder in the Windows installation to the `OracleBI/DAC/CustomSQLs` directory in the UNIX installation.
- 9 Start the DAC server using the appropriate script (see table above).

For example, use the `startserver.sh` script on a Solaris or HP machine.

Starting the Informatica Server on UNIX

Follow this procedure to start the Informatica Server.

Requirements

- From the UNIX machine, make sure you can connect to the transactional database and to the data warehouse database.
- The Informatica repository must have been configured using the Informatica Workflow Manager.
 - For instructions, see [Process of Configuring the Informatica Repository in Workflow Manager on page 118](#).
- The environment variables for the Informatica Server on UNIX must be set.
- The appropriate parameter files should have been copied to the `SrcFiles` directory.
- For Unicode, the `SiebelUnicodeDB` variable must be set.

To start the Informatica Server

- Go to the Informatica Server directory, and enter `./pmserver`.

For Teradata installations, enter `./pmserver ./pmserver.cfg>./pmserver.out`

Check the log file to verify the server started. If it started correctly, the log file will state: "Server Initialization Completed."

Stopping the Informatica Server on UNIX

Follow this procedure to stop the Informatica Server on UNIX.

To stop the Informatica Server

- Go to the Informatica Server directory, and enter the following command:

```
./pmcmd stopserver <repository username> <repository password> <port #>.
```

For example:

```
./pmcmd stopserver Administrator Administrator 4001
```

You can check whether the Informatica Server is running by entering the command `ps -elf |grep pmserver`.

If it lists `pmserver` process, the server is running. If it lists `grep pmserver`, that is the process of the `ps` command.

Starting the DAC Server on UNIX

Follow this procedure to start the DAC server on UNIX. Before you can start the DAC server, you must have already configured the `config.csh` or `config.sh` and `dac_env.csh` or `dac_env.sh` files.

To start the DAC server on UNIX

- In `csh`-related shells, issue the following command:

```
./startserver.csh
```

- In `bash`-related shells, issue the following command:

```
./startserver.sh
```

To run the DAC server in the background

- In `csh`-related shells, issue the following command:

```
nohup startserver.csh 2>&1 &
```

- 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.

Stopping the DAC Server on UNIX

Follow this procedure to stop the DAC server on UNIX.

To stop the DAC server on UNIX

- In csh related shells, issue the following command:

```
./stopserver.csh
```

- 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.

Teradata-Specific Installation and Configuration Checklist for UNIX

Table 34 provides a list of Teradata-specific installation and configuration steps that are performed during the Oracle BI Applications installation and configuration process for UNIX. You should review this list to make sure that you have performed all of the required Teradata-specific steps. You should also review [“Teradata-Specific Installation and Configuration Checklist \(Windows\)” on page 143](#) for Teradata-specific information related to the Windows environment.

Table 34. Teradata-Specific Installation Checklist for UNIX

Action	Link to Topic
Review Teradata-specific database guidelines	Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse on page 47.
Create database connections	Creating Database Connections for Teradata Installations on page 161.
Set up the HOSTS file	Setting Up the HOSTS File for Teradata Installations on UNIX on page 163.

7

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:

- [Oracle Business Analytics Warehouse Customization Scenarios on page 171](#)
- [Type I Customizations: Adding Columns to Existing Fact or Dimension Tables on page 172](#)
- [Other Types of Customizations Requiring Special Handling on page 185](#)
- [Type II Customizations: Adding Additional Tables on page 190](#)
- [Type III Customizations: Adding New Data as a Whole Row into a Standard Dimension Table on page 197](#)

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 or Oracle) use prepackaged adaptors.
- Non-packaged data sources use the Universal adaptor.

[Figure 5](#) shows the categories of supported customization scenarios, based on the data source.

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

		Data Warehouse Modifications		
		Add Additional Column to Existing Fact or Dimension Table	Add Additional Rows to Existing Fact or Dimension Tables	Add New Fact or Dimension Tables
Data Sources	Packaged Application (Uses Prepackaged Adaptor)	Type 1	Configure Filter	Type 2
	Non-Packaged Data (Uses Universal Adaptor)	Type 1	Type 3	Type 2

Figure 5. Data Warehouse Customization Scenarios

Types of Customizations

- **Type I.** In a Type I customization, you add additional columns (that are already mapped) from a source system and load the data into existing data warehouse tables.
- **Type II.** In a Type II customization, you use prepackaged adaptors to add new fact or dimension tables to the data warehouse, regardless of whether they are already mapped. Type II customizations normally require that you build new SDE and SIL mappings.
- **Type III.** In a Type III customization, you use the Universal adaptor to create new mappings.

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 Business Intelligence 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.

Type I Customizations: Adding Columns to Existing Fact or Dimension Tables

This section includes the following topics:

- [About Extending Mappings on page 173](#)
- [Impact of Customization on Upgrade on page 174](#)
- [Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse on page 177](#)
- [Scenario for Extending the Oracle Business Analytics Warehouse on page 178](#)

About Extending Mappings

Type I customizations involve extracting additional columns from source systems that are already mapped (for example, Siebel or Oracle) and loading the data into existing data warehouse tables. For Type I customizations, data can also come from non-packaged sources, but this section assumes that the sources have already been mapped with a Universal adaptor and only need to be extended to capture additional columns. (The initial mapping of a Universal adaptor is considered a Type III customization. For information, see [Type III Customizations: Adding New Data as a Whole Row into a Standard Dimension Table on page 197](#).)

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 [Figure 6](#) the preconfigured logic is shaded in gray. You should not modify anything contained within these objects. You should add customizations to the existing mapping, which allows them to run parallel to the existing logic.

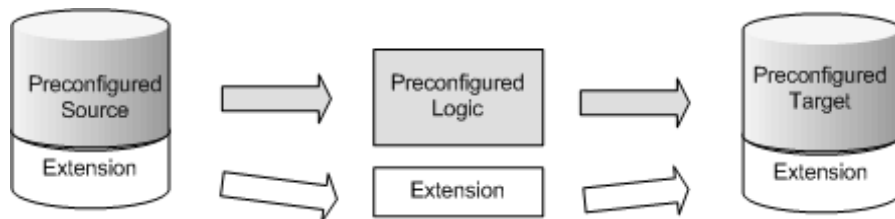


Figure 6. Type I Customization

Because some objects need to be modified in order to allow for extensions, Oracle BI Applications breaks down extensions into two 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. They should never be changed in any way unless directed by Oracle. 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.

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 [Figure 7](#).

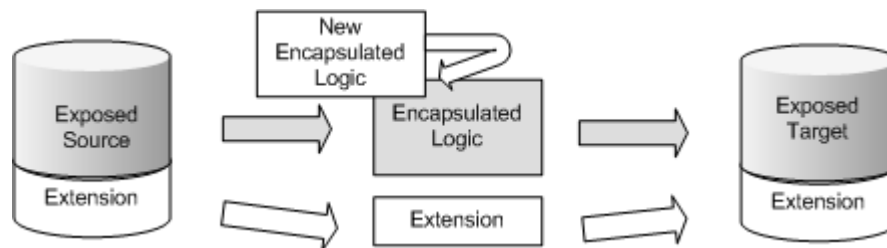


Figure 7.

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.

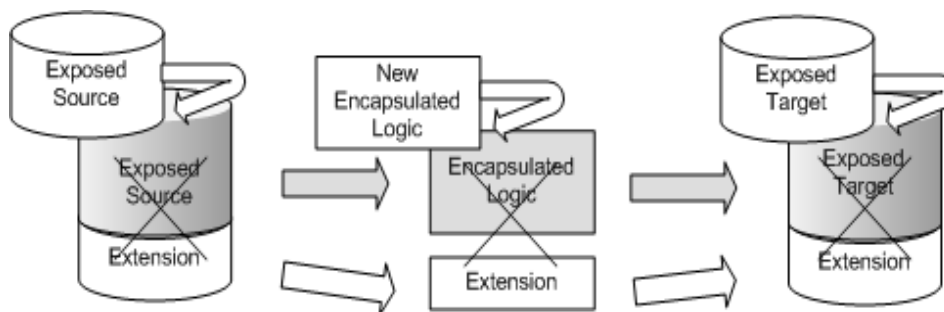


Figure 8.

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 reextended and then reconnected to the extension logic.

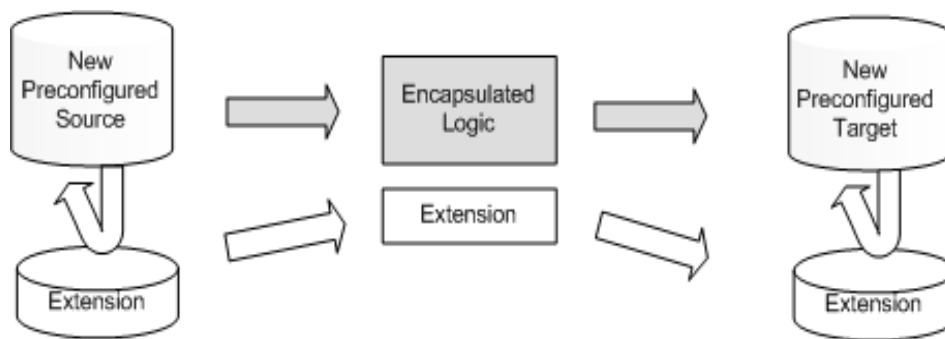


Figure 9.

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.

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.

NOTE: Most SDE adaptor folders use the concept of Business Component mapplets. These are extract mapplets that may contain relational, application, or flat file sources. The Siebel adaptor 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.

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 adaptor mapping. If the data is coming from a non-packaged source, then you must use a Universal adaptor mapping. (You will need to create a Universal adaptor 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.
- 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.

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 adaptors for Siebel, non-Siebel prepackaged application sources, and non-packaged data.

This section includes the following topics:

- [Tips for Modifying the SQL Override in a Source Qualifier on page 178](#)
- [Example of Extracting Data from a Siebel Source on page 179](#)
- [Including a Source Table for the Change Capture Process on page 181](#)
- [Example of Extracting Data from a Non-Siebel Packaged Source on page 181](#)
- [Example of Extracting Data from a Universal Source on page 183](#)
- [Example of Loading Data into an Existing Target Table on page 183](#)
- [Updating the DAC on page 185](#)

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 [“Example of Extracting Data from a Siebel Source” on page 179](#), 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.

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 adaptor 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 reimport 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 the Workflow Manager.

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

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.

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 adaptor 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 reimport the source table into the custom folder, replacing the existing version. For this example, the source table has not changed.

- 4 Open the mapping.
- 5 Edit the Business Component 'mplt_BC_ORA_OrganizationDimension_Customer' by right-clicking the mapplet and selecting 'Open Mapplet.'
- Remember, the Business Component mapplets are the only mapplets you can normally edit. You should not edit any other mapplets unless directed by Oracle.
- 6 Drag the columns LAST_UPDATE_LOGIN and ATTRIBUTE1 to the Source Qualifier, and then drag these columns to the Mapplet Output.
- 7 Edit the Source Qualifier to include the new columns, as shown below:

```
SELECT
...

HZ_PARTIES.SIC_CODE
- Added by J.Smith on 1/10/2007
, HZ_CUST_ACCOUNTS.LAST_UPDATE_LOGIN
, HZ_CUST_ACCOUNTS.ATTRIBUTE1

FROM
HZ_CUST_ACCOUNTS, HZ_PARTIES

WHERE
...
```

- 8 Return to the mapping.
- 9 Add a new expression and rename it to 'X_CUSTOM.'
- 10 Connect the new columns from the Business Component mapplet to this expression.
- 11 As a best practice, you should rename these ports to indicate both the table and column they came from. If the mapping is changed and the related exposed objects are replaced, this will make it easier to reconnect, because the custom expression will not be replaced
- 12 Connect these columns to the appropriate columns in the target definition.
- 13 Save your changes.
- 14 Refresh and validate the session in the Workflow Manager.

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

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 adaptor 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)
X_LAST_LOGIN	VARCHAR2(10)

- 5 Open the mapping.
- 6 Drag the columns LAST_UPDATE_LOGIN and ATTRIBUTE1 to the Source Qualifier.
- 7 Add a new expression and rename it to 'EXP_CUSTOM.'
- 8 Connect the new columns from the Source Qualifier to this expression.
- 9 Connect these columns to the appropriate columns in the target definition.
- 10 Save your changes.
- 11 Refresh and validate the session in the Workflow Manager.

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

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 exposed object, just like the Filter.

- 10** Connect these columns to the appropriate columns in the target definition.
- 11** Save your changes.
- 12** Refresh and validate the session in the Workflow Manager.

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

Updating the DAC

After making these changes to the mappings, you need to register the changes in the DAC. You need to include the table definition, with any additional columns or indexes, and the required changes so the tasks execute the modified sessions in the new custom folders. For information about registering data warehouse objects in the DAC, see the *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

Other Types of Customizations Requiring Special Handling

This section includes the following topics:

- [Modifying Type II SCD Triggers on page 185](#)
- [Adding Dimensions to Existing Facts on page 186](#)
- [Adding Date Dimensions to Existing Facts on page 189](#)
- [Adding Currencies to an Existing Table on page 189](#)

Modifying Type II SCD Triggers

All dimensions are configured as Type I dimensions when first installed. If you want to capture historical changes, the dimension must be enabled to capture Type II changes. However, a common form of customization is to change the criteria that triggers a Type II change in a dimension. Most changes in a dimension are treated as Type I 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 Type II change. You can extend the logic that triggers a Type II change by adding additional columns to the logic that tracks Type II changes. In addition, you can remove columns from this logic in case you do not want these types of changes to trigger a Type II change. Modifying the Type II tracking logic is one of the only exceptions to the rule that you should not make changes to shipped logic. The logic that tracks Type II changes is contained in exposed objects in each SIL dimension mapping that supports Type II 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 Type II 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 Type II change.

This expression contains a variable port named 'TYPE2_COLS_DIFF'. If this port is flagged as 'Y' then a Type II change will be triggered. If it is flagged as 'N' then a Type I change will be triggered.

To change the columns used to determine a Type II change, modify the lookup to pass any additional columns you want to be evaluated for Type II 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 Type II 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')))))
```

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 35 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 35. 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_A MTS	Not applicable

Table 35. Formats to Derive INTEGRATION_ID

Dimension	Foreign Key	When Source is Oracle Application	When Source is Siebel Application
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_ROLE S	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
W_ORG_D	CUSTOMER_WID	TO_CHAR(CUSTOMER_ID) - CUSTOMER_ID is CUST_ACCOUNT_ID from HZ_CUST_ACCOUNTS	UNION OF S_ORG_EXT AND S_CONTACT. When source is S_ORG_EXT, ROW_ID is passed. When source is S_CONTACT, use "C- ' ROW_ID - the ROW_ID is the ROW_ID from the S_PERSON (not from S_ORG_EXT). This is the new value passed to lookup the Contact Customers in W_ORG_D for resolving the ACCOUNT_WID (read as CUSTOMER_WID)
W_PAYMENT_METHOD_D		LOOKUP_CODE	Not applicable
W_PAYMENT_METHOD_D		TO_CHAR(TERM_ID)	Not applicable
W_PERSON_D	CUST_CONTACT_WID	TO_CHAR(PARTY_ID) - PARTY_ID from HZ_PARTY_RELATIONS	S_CONTACT.ROW_ID

Table 35. Formats to Derive INTEGRATION_ID

Dimension	Foreign Key	When Source is Oracle Application	When Source is Siebel Application
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 Type II 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.

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.

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.

Type II Customizations: Adding Additional Tables

This section includes the following topics:

- [About Creating New Dimension or Fact Tables on page 190](#)
- [Using Custom Folders in the Oracle Business Analytics Warehouse on page 191](#)
- [Creating Custom Informatica Workflows on page 192](#)
- [Important Notes About Customizing the Oracle Business Analytics Warehouse on page 193](#)
- [Including a Source Table for Change Capture Process on page 195](#)
- [Adding a New Dimension in the Oracle Business Analytics Warehouse on page 195](#)
- [Adding a New Fact Table in the Oracle Business Analytics Warehouse on page 196](#)
- [Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse on page 197](#)

About Creating New Dimension or Fact Tables

If you are creating 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*.

When you create a new extension table, use the same naming convention as is used in the Oracle Business Analytics Warehouse by adding the prefix WC to the table name, for example, WC_TABLENAME_DS. This will simplify future upgrades to the Oracle Business Analytics Warehouse.

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.

Required Columns

For main and extension 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 example, Siebel OLTP is 1. For data from any external data source assign a number greater than 1.

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.

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.

Using Custom Folders in the Oracle Business Analytics Warehouse

If you want to make changes to the Oracle Business Analytics Warehouse, you must create a custom folder and make the changes in it. Do not change objects in any of the shipped folders unless explicitly directed by Oracle. The shipped folders and the objects within them may be overwritten in future upgrades.

The deployed repository does not include any custom folders; you must create your own. You should create a custom folder for each SDE folder you have deployed that will have customizations. These folders hold the extract mappings to various sources. You should also create a separate custom folder for customizations you want to make to the SILOS folder. Do not store customized extract and load mappings in the same folder.

The easiest way to modify an object is to copy an existing object from the shipped folder into the corresponding custom folder and to reuse existing business components, source and target definitions, transformations, mapplets, and mappings.

NOTE: If source tables are extended, then the source tables require manual editing in Informatica Designer. Do not import the tables from the database into the repository, because it changes the source table definitions for the entire Oracle Business Analytics Warehouse.

When importing new tables from the database into the custom folder, use the Oracle Business Analytics Warehouse and transactional database ODBC database connections (using the ODBC drivers supplied by your database vendor) to connect to the source and target databases.

After importing new table definitions, change the database type to Oracle in the Informatica repository, regardless of the database platform you are using. This has no effect on the choice of a relational database. This step is very important because in Informatica if the database type of the source tables are not identical, all mappings and workflows that refer to the source tables will be invalid.

Creating Custom Informatica Workflows

You must create custom workflows for all customized mappings. The general requirements for creating custom workflows are as follows:

- Create the workflow such that each loads only one table. This helps to integrate the workflow with the DAC.
- The workflow name should match a session name that is used inside the workflow. This helps the DAC to collect certain statistics.
- The flag "Fail parent if this task fails" should be checked for all the sessions within a workflow.
- The flag "Fail parent if this task does not run" should be checked for all the sessions within a workflow.
- The Stop on Errors parameter should be set to 1 for all sessions within a workflow. This parameter is located in the Error Handling area of the Config Object tab in Informatica Designer.
- Set the appropriate source and target connection values in Informatica Designer.
- If the workflow is going to be used for full load command, you can choose to load in bulk mode (applicable only to Oracle and DB2-UDB databases). If the workflow is going to be used for the full load command in the DAC, then in the Properties tab of Informatica Designer, set the Target Load type to Bulk. This requires that the target table does not have any indices during the load. The DAC will drop the indices automatically; no action is required on your part.
- For all entities, such as dimension and fact tables, create two workflows, one to be used for a full load and the other to be used for an incremental load. Both workflows are based on the same mapping. The same mapping gets executed during both full and incremental loads. This provides an opportunity to tune each of these load scenarios.
- If the workflow is designed for loading a dimension in a full mode, make sure the workflow also includes a session for the unspecified row creation.

- When defining a task in the DAC, you must choose the appropriate truncate option. This helps the DAC to decide whether to drop and create indices on the table.
- You should not truncate target tables using the Informatica 'truncate target' option. It is especially important that the DAC handle truncating tables when extracting from more than a single source system and loading into the same data warehouse. The DAC will dynamically determine when tables need to be truncated. By setting the truncate option in the Informatica workflow, the table will always be truncated, limiting your ability to extract and load data from multiple sources. For example, you cannot extract data from two systems and load the data into the same staging table in parallel as the Informatica session will truncate the staging table when another session is running.
- If some sessions have to be run in serial order and if all the sessions need to be rerun upon failure of the workflow, then you should consider designing a single workflow that has sessions run in serial. If upon failure, not all of the sessions need to be run again, then consider designing separate workflows and define dependencies in the DAC.
- The custom workflows can be plugged into an ETL process by registering them in the DAC. All new tasks need to be registered in the DAC, with the appropriate properties set. Also, you must register in the DAC, source and target tables, task definitions, and dependencies.

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.

Additional Customization Instructions

- **Table definitions in Informatica.** Make sure that the SQL style is set to Oracle while importing the table definitions from external data sources. Even if the actual data source is of another database type, such as DB2 or MSSQL, it does not affect the logic of how data gets loaded.
- **Update strategy.** For loading new fact and dimension tables, design a custom process on the source side to detect the new and modified records. The SDE process should be designed to load only the changed data (new and modified). If the data is loaded without the incremental process, the data that was previously loaded will be updated again, which is a costly process. For example, the logic in the preconfigured SIL mappings looks up the destination tables based on the INTEGRATION_ID and DATASRC_NUM_ID and returns the ROW_WID if the combination exists, in which case it updates the record. If the lookup returns null, it inserts the record instead. In some cases, last update date(s) stored in target tables are also compared in addition to the columns mentioned above to determine insert or update. Look at the similar mappings in the preconfigured folder for more details.
- **ETL process.** When using multiple sources for the data warehouse, you can decide to load from all of them at the same time or at different time frequencies using different execution plans.

- **Truncating target tables.** Truncating should be done through the DAC. A single task has place holders for a full load, and one for an incremental load.
 - For the SDE workflows, the commands for full load and incremental load are the same. They should have the Truncate Always flag checked in the DAC. For these kinds of tasks, the command for full load and incremental load are based on the same mapping.
 - For SIL workflows, the command can be different for full and incremental loads. They should have the Truncate For Full Load option checked in the DAC. When a table gets truncated, the indices are automatically dropped and created after the data is loaded. The workflow associated with the full load command can have the Bulk Load option turned on for an optimized version of the mapping that quickly inserts data. Note that if there are indexes on the table, the bulk load may fail, so it is very important that the indices are registered in the DAC and that you drop all of the indexes on this table during a full load if you use the bulk load option.
 - If a source requires an auxiliary task, it needs to be run only during incremental mode. So, for these tasks, the full load command is empty. No truncate options should be set.
- **ETL_PROC_WID.** Use the same ETL_PROC_WID in W_PARAM_G table in custom mappings. ETL_PROC_WID is a reference key to Run History in the DAC. To use the same ETL_PROC_WID, copy the reusable lookup (called LKP_ETL_PROC_WID) defined in the SILOS folder. The input to the lookup is a constant (hard coded to 1).
- **DATASRC_NUM_ID.** Use a parameter to define this value in the mapping. The DAC will automatically create a parameter file with the correct DATASOURCE_NUM_ID, which will be picked up by the parameter in the mapping. This allows you to make multiple copies of the same mapping when you have multiple instances of the same transactional database type. You do not have to do any additional hard-coding other than to register the sources in the DAC.
- **Creating indices and naming conventions.** Staging tables typically do not require any indices. Use care to determine if indices are required on staging tables. Create indices on all the columns that the ETL will use for dimensions and facts (for example, ROW_WIDs of Dimensions and Facts, INTEGRATION_ID and DATASRC_NUM_ID, and flags). Carefully consider which columns or combination of columns filter conditions should exist, and define indices to improve query performance. Inspect the OTB objects for guidance. Name all the newly created tables as WC_. This helps visually isolate the new tables from the OTB tables. Keep good documentation of the customizations done; this helps when upgrading your data warehouse. Once the indices are decided upon, they should be registered in the DAC, either manually or by right-clicking on the certain table and invoking the Import Indices command.
- **Currency.** For currency-related data, populate the table with the base currency and exchange date field (in order to convert the data appropriately). The data for currency conversion should be maintained in the main data source. (The currency data is maintained by converting all the currency information into a single Base Currency code specified in the DAC.)
- **Day Dimension.** For the data that is related to W_DAY_D, use the reusable transformation EXP_DAY_DIMENSION_FK_RESOLUTION. Note that this transformation will take a date as input and return the foreign key to the Day dimension as output in the appropriate format (YYYYMMDD) avoiding the need of costly join or lookup to the W_DAY_D dimension table each time for resolution. Copy the reusable transformation and use it.

- **List Of Values.** This applies in particular to Type I 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.

Including a Source Table for Change Capture Process

This procedure applies to Siebel source tables only.

To include a source table for the change capture process

- 1 Verify whether the source table is registered in the DAC.
 - a If an entry is not in the DAC, create a new record for the table and assign an image suffix to it.
 - b If the table is registered, make sure there is an image suffix assigned for this table.
- 2 If an image suffix does not exist for the source table, assign one.

The image suffix should be three characters long. The recommended naming convention is to start with C. For example, use C1, C2, CA1, CA2, and so on.
- 3 Make sure the image suffix is not used by other tables by querying the tables list in the DAC for that image suffix.

The DAC client does not validate this information when entering data.
- 4 Create Image tables in the Siebel transactional database.
 - a Right-click on the table record in the DAC client, and choose Generate Change Capture Scripts.

This will assist you in creating the image tables, necessary indices, and triggers if you plan on tracking deletes on the transactional database.
 - b Execute these scripts on the transactional database if you have the appropriate permission, otherwise ask the OLTP DBA to create it.
- 5 For the task that is created for the extract process, set the Build Image flag to True and select the new table as an auxiliary table or primary table.

NOTE: At the end of the change capture process, the DAC creates a view on the actual source table. Use this view as the main source table in all the extract procedures. For example, if your new source table is S_COMPENSATION, then the default view name is V_COMPENSATION.

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** Make the following changes to the fact-loading mapping to link the new dimension:
Do one of the following:
 - Use the extension table's predefined fact table foreign keys that join to this new dimension. (In the above example, the extension table for W_REVN_F must be used to store the foreign key of this new dimension.)
 - Create a new extension table for the fact table with predefined structure to hold a new foreign key, if the foreign keys in existing the extension table have been exhausted.
 - Create new mappings (SDE and SIL) to populate the extension table columns of the fact extension table.
- 6** 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.
- 7** Register the workflows as tasks in the DAC.
- 8** 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.
- 9** For SIL workflows of the dimension make sure you set only Truncate for Full Load option to True.
- 10** Make sure the target table of the SDE_XYZ is defined as source table for SIL_XYZ.
- 11** Make sure the target tables of the SIL_XYZ and SIL_RevenueFact (load base fact task) are defined as source tables for CustomSIL_RevenueFact.

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 [Including a Source Table for Change Capture Process on page 195](#).
- 4** Create SDE mappings to populate the custom stage table. Use the view created by change capture as the main table in the SQL so that it extracts only the incremental data. Use the reference maps (above) as examples of how to populate the system columns. Be sure to truncate the stage table in corresponding workflows.
- 5** Create SIL mapping to populate the custom fact table. Use reference maps as examples of how to populate the system columns.
- 6** Use lookups or SQL override joins to dimension tables for populating dimension foreign keys (ROW_WIDs) pointing to the existing dimension.
- 7** In the DAC, register the target tables.
- 8** Create new tasks for the workflows.
- 9** For the SDE task, make sure you have the Build Image flag set to True, and list all the source tables that it queries from. Choose one or more tables as primary or auxiliary. For the target tables choose the staging table. Set the Truncate Always flag to True.
- 10** For the SIL task, list all the dimensions that will be required under source tables.

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.

Type III 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.

8

Configuring the Oracle BI Repository for Oracle Business Analytics Warehouse

This chapter describes how to configure the Oracle BI Repository for the Oracle Business Analytics Warehouse.

It contains the following topics:

- [Modifying the Oracle BI Repository for Siebel version 6.3 and version 7.5 applications on page 199](#)
- [Configuring the Oracle BI Repository Connections on page 200](#)
- [Setting up Date Specific Metrics on page 202](#)
- [Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse on page 203](#)
- [Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse on page 204](#)
- [About Oracle BI Repository Variables on page 204](#)
- [About Oracle BI Repository Variables on page 204](#)
- [About User Authentication on page 207](#)
- [About the Security or Visibility Configuration on page 208](#)
- [About the Group Variable on page 211](#)
- [About the Period Ago Keys for Oracle Business Analytics Warehouse on page 212](#)
- [About Configuring Usage Tracking for Oracle Business Analytics Warehouse on page 213](#)
- [About the Incremental Deployment of the Oracle BI Applications Repository on page 213](#)

Modifying the Oracle BI Repository for Siebel version 6.3 and version 7.5 applications

Before you can enable Siebel version 6.3 and version 7.5 applications, you must modify the Oracle BI Repository.

To modify the Oracle BI 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

Configuring the Oracle BI Repository Connections

You need to configure the Oracle BI repository variables and connection pools to connect to your database.

Oracle BI repository uses three databases defined in the Physical layer:

- Oracle Data Warehouse
- Oracle EBS OLTP
- Siebel OLTP

The Oracle Business Analytics Warehouse physical database has two predefined connection pools:

- **Oracle Business Analytics Warehouse Connection Pool.** The Oracle Business Analytics Warehouse Connection Pool is the main connection pool in the Oracle BI repository. You need to configure this connection pool to connect to your physical data warehouse. The connection is used by the session initialization blocks. You can use this connection pool to set up a dynamic data source name.
- **Oracle Business Analytics Warehouse Repository Initblocks Connection Pool.** You need to configure the Oracle Business Analytics Warehouse Repository Initblocks Connection Pool to connect to 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.

The 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.

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 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.

To configure the Oracle Data Warehouse, Oracle EBS OLTP, and Siebel OLTP physical databases

- 1 Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the \$SAHome\OracleBI\Server\Repository folder.
- 2 In the Physical pane, double-click the Oracle Data Warehouse object.
- 3 In the Database list, click your database type.
- 4 Save the repository.
- 5 Click Yes to Check Global Consistency.
- 6 Repeat steps 1-5 for the Oracle EBS OLTP and Siebel OLTP data warehouse objects
- 7 Click OK when the Warnings are displayed.

To configure the Oracle BI repository connection pools

- 1 Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the \$SAHome\OracleBI\Server\Repository folder.
- 2 In the Physical pane:
 - a Double-click the Oracle Data Warehouse Connection Pool within the Oracle Data Warehouse.
 - b Repeat step 2 for the other connection pools listed above.
- 3 Repeat Step a through Step d 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.

To configure the Oracle BI repository variables

- 1 Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the \$SAHome\OracleBI\Server\Repository folder.
- 2 On the Manage menu, click Variables.
- 3 In the Variables Manager window, click Static.
- 4 Edit the OLAP_DSN, OLAP_USER, and OLAPTBO variables, and close the Variables Manager window.
- 5 On the Manage menu, click Security
- 6 In the Security Manager window, click Users, and then:
 - a Double-click on Administrator user, and type a new password.
 - b Double-click on SADMIN, and type a new password.
 - c Save and close the Security Manager.
- 7 Save the repository.
- 8 Click Yes to Check Global Consistency.
- 9 Click OK when the Warnings are displayed.

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.

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, if the Invoice fact table has three metrics called Invoice Amount, Fulfill Amount, and Paid Amount, then 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 36](#), 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 36. Invoice Fact Table Example

Date	Invoice Amount	Fulfill Amount	Payment Amount
January	4000	5000	4500

To implement data specific metrics

- 1** Open the OracleBIAnalyticsApps.rpd with the Oracle BI Administration Tool.
The OracleBIAnalyticsApps.rpd file is located in the \OracleBI\server\Repository folder.
- 2** Right-click on Oracle Business Analytics Warehouse in the Physical layer, and create a new physical alias for the fact table.
- 3** Create Joins for the physical alias which are the 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.

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 premapped period ago metrics, however you can map other metrics by using the following procedure.

To set up additional time series metrics

- 1** Open the OracleBIAnalyticsApps.rpd with the Oracle BI Administration Tool.
- 2** Right-click on Oracle Business Analytics Warehouse in the Physical layer, and create a new Period Ago physical alias table.
- 3** Create additional tables in the Physical Layer for each Period Ago alias required.
For example, Quarter Ago, Year Ago, and so on.
These aliases need to have the same joins as the base fact table, except for the date join, which you can change in the next step. Setting up this alias is easier to accomplish by copying the base table.
- 4** Change the join to the date dimension (W_DAY_D) to use the appropriate Period Ago Key.
- 5** Map the Period Ago metrics in the logical table using the new fact alias by creating a new logical table source under the fact table.
- 6** Set the content pane levels for the period ago logical table source, to specify the level of the source data.

These settings are the same as the base fact table.

- 7 Save and close the OracleBIAnalyticsApps.rpd file.

Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse

Oracle Business Analytics Warehouse is preconfigured to map dimension tables required for analysis. The physical layer in the Oracle BI repository provides several other dimensional table keys that can be used for certain specific analysis. If you need to set up any of the additional dimensions tables to the physical layer, perform the following procedure.

To set up additional dimension tables

- 1 Validate that the dimension table key is resolved appropriately for the data source that you are using.
NOTE: Dimension tables do not apply to every source system.
- 2 Open the OracleBIAnalyticsApps.rpd with the Oracle BI Administration Tool.
- 3 Add a dimension table alias in the physical layer.
- 4 Join the dimension table alias to the fact table alias using the appropriate keys.
- 5 Save and close the OracleBIAnalyticsApps.rpd file.

About Oracle BI Repository Variables

The Oracle BI repository is preconfigured with variables that are used for both reporting and internal usage.

Table 37 lists the Oracle BI repository date variables and their descriptions.

Table 37. Oracle BI Repository Date Variables

Variable Name	Description
BUILD	Holds the internal build number information for the Oracle BI repository.
CAL_MONTH_YEAR_AGO	Returns the value of Previous Year Month in the YYYY/MM format.
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 Fiscal Quarter in the YYYY Q n format.
CURRENT_FSCL_WEEK	Returns the value of Current Fiscal Week in the YYYY Week nn format.

Table 37. Oracle BI Repository Date Variables

Variable Name	Description
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 Fiscal 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 Fiscal Quarter in YYYY Q n format.
PREVIOUS_FSCL_MONTH	Returns the value of Previous Fiscal Month in the YYYY/MM format.
PREVIOUS_FSCL_QUARTER	Returns the value of Previous Fiscal 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 Fiscal 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.
NEXT_FSCL_MONTH	Returns the value of Next Fiscal Month in the YYYY / MM format.
NEXT_FSCL_QUARTER	Returns the value of Next Fiscal 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 Fiscal Quarter in the YYYY Q n.
NEXT_WEEK	Returns the value of Next Week in the YYYY Weeknn format.
NEXT_YEAR	Returns the value of Next Year in the YYYY format.
YEAR_AGO_DAY	Returns the value of year ago date in the mm/dd/yyyy format.
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.

Table 37. Oracle BI Repository Date Variables

Variable Name	Description
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.
IS_CME_ORDER_NUM	Set to 1 if the Order is a CME Order. This variable should not be changed.
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.

Presentation Services Configuration Variables

The Presentation Services configuration variables are used to control the number of years displayed in the global filters for the year and fiscal year variables.

[Table 38](#) lists the Presentation Services configuration variables and their descriptions in the Oracle BI repository.

Table 38. Presentation Services Configuration Variables

Variable Name	Description
FILTER_CAL_FROM_YEAR	You need to set this variable to the earliest year for the Year global filter.
FILTER_CAL_TO_YEAR	You need to set this variable to the earliest year for the Year global filter.
FILTER_FSCL_FROM_YEAR	You need to set this variable to the earliest year for the Year global filter.
FILTER_FSCL_TO_YEAR	You need to set this variable to the earliest year for the Year global filter.

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.

Table 39. Presentation Services Configuration Variables

Variable	Description
FILTER_CAL_FROM_YEAR	You need to set this variable to the earliest year for the Year global.
FILTER_CAL_TO_YEAR	You need to set this variable to the earliest year for the Year global filter.
FILTER_FSCL_FROM_YEAR	You need to set this variable to the earliest year for the Year global filter.
FILTER_FSCL_TO_YEAR	You need to set this variable to the earliest year for the Year global filter.

For more information about configuring user authentication, see *Oracle Business Intelligence Server Administration Guide*.

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 40](#) lists the groups in the Oracle Business Analytics Warehouse repository.

Table 40. 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 Oracle's Siebel Enterprise Contact Center and Agent Performance application content.
Contact Center and Agent Performance User	This user group is able to view a subset of Oracle's Siebel Enterprise Contact Center and Agent Performance application content.
Contact Center Sales Analyst	This user group is able to view Oracle's Siebel Enterprise Contact Center Sales Analytics application content.
Contact Center Sales User	This user group is able to view a subset of Oracle's Siebel Enterprise Contact Center Sales Analytics application content.

Table 40. Repository User Groups

Repository User Group	Description
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 Applications Consumer application content.
Enterprise Contact Center User	This user group is able to view Oracle BI Applications Consumer 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 Workforce Analytics application content.
Human Resources Vice President	This user group is able to view high-level application content for Oracle Workforce Analytics application.
Inventory Analyst	This user group is able to view application content for Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics).
Inventory Manager	This user group is able to view high-level application content for Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics).
Primary Owner-Based Security	Used for securing owner-based data elements that come from the transactional system.
Primary Position-Based Security	Used for securing position-based data elements that come from the transactional system.
Purchasing Buyer	This user group is able to view Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to purchasing.
Sales Executive Analytics	This user group is able to view high-level application content for Siebel Enterprise Sales Analytics application.
Sales Manager	This user group is able to view most of the high-level application content for Oracle BI Applications Consumer application.
Sales Manager Analytics	This user group is able to view most of the high-level application content for Oracle BI Applications Consumer application.

Table 40. Repository User Groups

Repository User Group	Description
Sales Operations Analytics	This user group is able to view operational application content for Oracle BI Applications Consumer application.
Sales Representative Analytics	This user group is able to view low-level application content for Oracle BI Applications Consumer application.
Sales Rev and Fulfill Analyst	This user group is able to view the content for Oracle BI Applications Consumer Revenue and Fulfillment application.
Sales Rev and Fulfill Exec	This user group is able to view the high-level application content for Oracle BI Applications Consumer 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 Applications Consumer Revenue and Fulfillment application.
Sales Rev and Fulfill Rep	This user group is able to view low-level application content for Siebel Enterprise Sales Analytics Revenue and Fulfillment application.
Sales Revenue Analyst	This user group is able to view the content for Oracle BI Applications Consumer Revenue application.
Sales Revenue Exec	This user group is able to view the high-level application content for Oracle BI Applications Consumer Revenue application.
Sales Revenue Mgr	This user group is able to view most of the high-level application content for Oracle BI Applications Consumer Revenue application.
Sales Revenue Rep	This user group is able to view low-level application content for Oracle BI Applications Consumer Revenue application.
Service Delivery and Costs Analyst	This user group is able to view Service Delivery and Costs for Oracle BI Applications Consumer application content.
Service Delivery and Costs User	This user group is able to view a subset of Service Delivery and Costs for Oracle BI Applications Consumer application content.
Supplier Performance Analyst	This user group is able to view Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to supplier performance.

Table 40. Repository User Groups

Repository User Group	Description
Supplier Performance Manager	This user group is able to view high-level content for Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to supplier performance.
Supply Chain Executive	This user group is able to view Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content.

About the Group Variable

The Group variable determines the membership of a user in the various security groups. If you are using Oracle Business Analytics Warehouse with Siebel Relationship Management Warehouse, you need to associate users to the Group variable as part of the repository configuration.

For more information about configuring the Group variable, see *Oracle Business Intelligence Presentation Services Administration Guide*. [Figure 10](#) shows an example of an initialization block that associates a user to a Group membership.

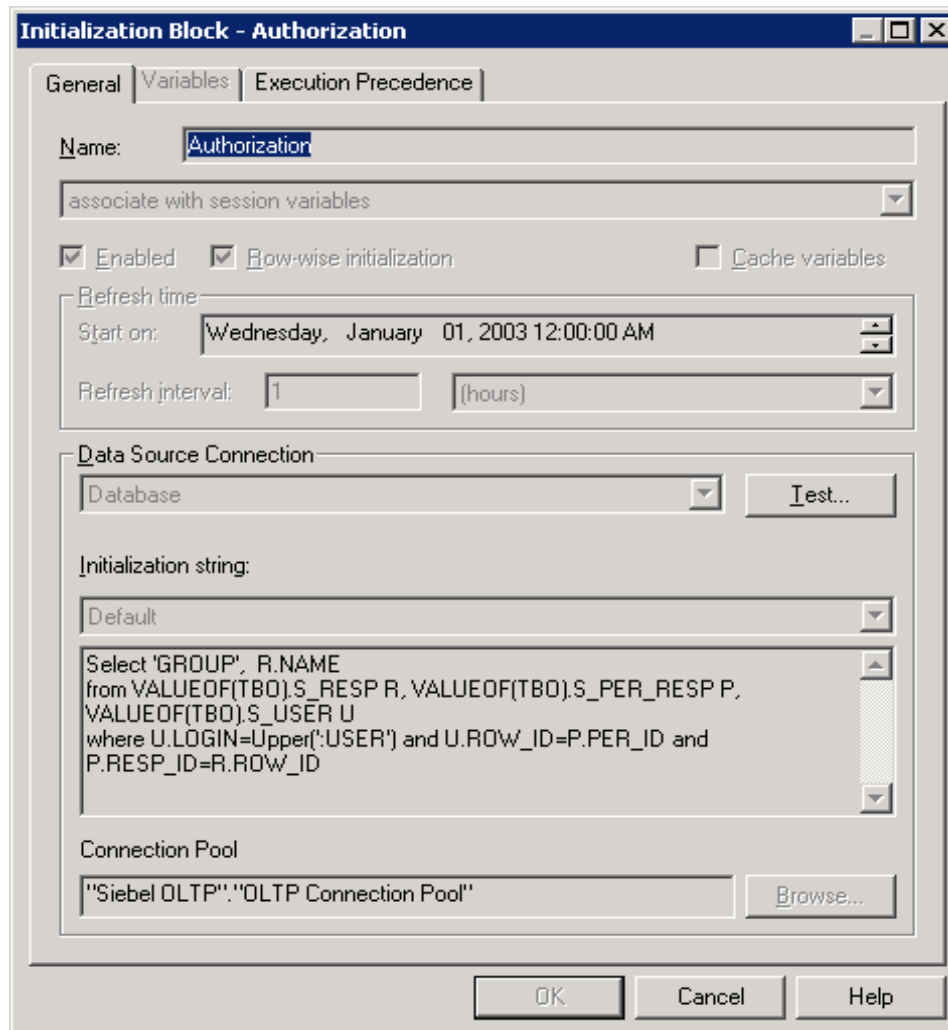


Figure 10. Group Initialization Block Example

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_KEY

- 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 surrogate keys in Oracle Business Analytics Warehouse fact tables uses are different to the surrogate keys that the Oracle Business Analytics Warehouse uses. The joins in Oracle Business Analytics Warehouse uses the Period Ago fields in the W_DAY_D table.

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*.

About the Incremental Deployment of the Oracle BI Applications Repository

Oracle Business Analytics Warehouse consist of various families of Oracle BI Applications, for example, Supplier Performance Analytics, Contact Center Telephony Analytics, General Ledger & Profitability Analytics, and so on. You can purchase these applications at different times. You can customize functionality and incrementally add new application or applications.

This section describes the procedure for deploying multiple applications. You can repeat the procedure to add applications incrementally.

Figure 11 shows a single Oracle Business Analytics Warehouse application environment. A license key is used to extract the Oracle BI repository. You can modify the Oracle BI repository to suit your business needs.

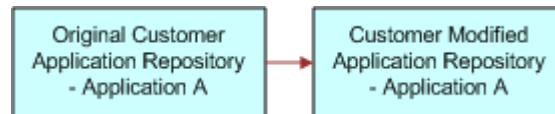


Figure 11. Single Repository Environment

When you purchase another Oracle Business Analytics Warehouse application, you need to use the combined license key to extract both Oracle BI application repositories. 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 [About Oracle BI Repository Variables on page 204](#).

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 Figure 12.

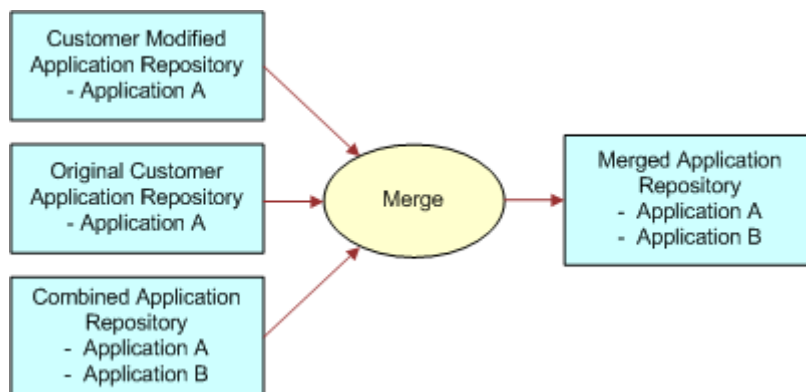


Figure 12. Merging Repositories

You can repeat this merging procedure to add more Oracle Business Analytics Warehouse applications to the Oracle BI repository.

9

Configuring Common Components of the Oracle Business Analytics Warehouse

This chapter provides procedural information on how to configure components that are common, regardless of which application you purchased.

It contains the following topics:

- [Configuring Extracts on page 215](#)
- [Configuring Loads on page 219](#)
- [Filtering and Deleting Records on page 219](#)
- [Configuring Slowly Changing Dimensions on page 222](#)
- [About Document, Local, and Global Currencies on page 227](#)
- [Process for Configuring Currencies on page 227](#)
- [Handling European Monetary Union \(EMU\) Currencies on page 228](#)
- [About Stored Lookups on page 234](#)
- [Codes Lookup on page 234](#)
- [About Resolving Dimension Keys on page 237](#)
- [About Domain Values on page 238](#)
- [Configuring the Domain Value Set with CSV Worksheet Files on page 243](#)
- [Configuring the Domain Value Set Using PowerCenter Designer on page 244](#)
- [Configuring Conformed Dimensions on page 245](#)

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:

- [Extracting Additional Data on page 216](#)
- [Setting Up the Delimiter for Source Files on page 217](#)

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.

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.

Extracting Data from a New Source Table

Business Components are packaged as mapplets, which reside in source-specific folders within the repository. Business Components are used to extract data from the source system. You can configure these mapplets to perform the following:

- Extract data from a new source table
- Set incremental extraction logic

The following procedure contains instructions for adding a new table to the Business Component. The procedure includes adding a new source definition, connecting the ports to the Source Qualifier, editing the Source Qualifier, connecting the ports to the Output transformation, and editing the Output transformation.

To add a new source table to an existing Business Component mapplet

- 1 In PowerCenter Designer, open the applicable source system configuration folder.

- 2 Open Mapplet Designer tool.
- 3 Drag the Business Component mapplet into Mapplet Designer to view the transformations that comprise the Business Component.
- 4 Expand the Sources folder, and copy a source table into the mapplet by dragging and dropping the table into Mapplet Designer.
- 5 Connect the applicable ports from the new source definition to the Source Qualifier by clicking on the port in the new source table and dragging it to the connecting port in the Source Qualifier.
- 6 Double-click the Source Qualifier to open the Edit Transformations box.
In the Ports tab, make any changes to the new ports for data type, precision, scale, or all these values, as necessary.
- 7 Connect the applicable ports from the Source Qualifier to the 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.

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.

Configuring a Source Adapter Maplet

The majority of all source-specific transformations occur in the Source Adapter maplet; source-independent transformations generally take place in the Analytic Data Interface (load mapping). The Source Adapter maplet converts source-specific data elements into standard formats and then stores them in a staging table. The source independent loading mapping then picks up these records, which are already transformed into standard format.

Figure 13 illustrates the three components of the Source Adapter maplet that allow transformations of data to occur. The three components are Maplet Input (MAPI), Expression transformation (EXP), and Maplet Output (MAPO).

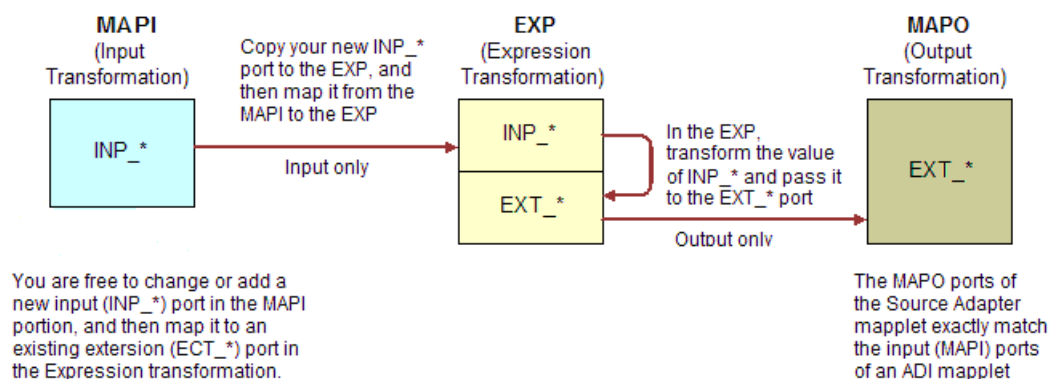


Figure 13. Input, Expression, and Output Ports of a Source Adapter Maplet

In Figure 13, 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 maplet

- 1 In PowerCenter Designer, open the applicable source system configuration folder.
- 2 Open the applicable Source Adapter maplet.
- 3 Double-click the MAPI component of the maplet, 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.

Configuring Loads

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

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 warehouse as well.

CAUTION: It is important to note that delete and primary extract mappings must always be disabled together; you may not disable only one type.

About Primary Extract and Delete Mappings Process

NOTE: This section does not apply to Oracle's Siebel adaptor, 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.

Figure 14 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.

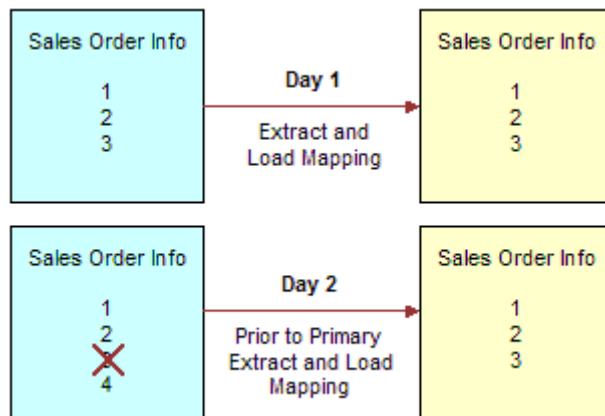


Figure 14. Changing Sales Order Information

Figure 15 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 15. When the extract and load mappings run, the new sales order is added to the warehouse.

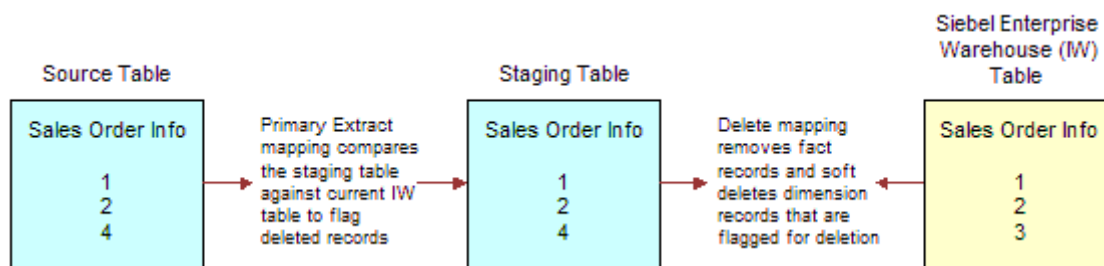


Figure 15. Primary Extract and Delete Mappings

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.

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 [About Working with Primary Extract and Delete Mappings on page 221](#).

Enabling Delete and Primary Extract Sessions

If you want to mark your source-deleted records as deleted in the Oracle Business Analytics Warehouse, you need to enable the delete and primary extract tasks for your application.

To enable primary extract and delete sessions

- 1 Log into DAC
- 2 Go to your container
- 3 Select the 'Tasks' tab
- 4 Query for all tasks containing the string 'Delete' or 'Primary'
- 5 Uncheck the 'Inactive' check box for those tasks.

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, as shown in [Figure 16](#). Users may configure Oracle BI Applications to support both Type I SCDs, in which data is overwritten with updates, and Type II SCDs, in which the original records are maintained while a new record stores the updated data. Choosing Type I or Type II SCDs depends on identifying your historically significant attributes.

By default, all dimensions are using Type I updates. If you need change a dimension to Type II SCD update, follow the procedure below.

To turn a dimension to Type II SCD update enabled

1. In the parameter file `parameterfileDW.txt` located under Informatica PowerCenter 7.1.4\Server\SrcFiles
2. Locate the parameter section for the `SIL_*` mapping which load this dimension.
3. Set the `$$TYPE2_FLG` to 'Y'.

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.

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.

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.

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 [Figure 16](#), 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.

Supplier (Source)				Supplier (Analytic Data Whs.)		
Supplier Name	State Name	Contact Name		Supplier Name	State Name	Contact Name
Acme	NY	Chris	→	Acme	NY	Chris
KMT	CA MI	Suzanne	→	KMT	MI	Suzanne

Figure 16. Type I Slowly Changing Dimension Example

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). In [Figure 17](#), 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 TS_CUSTOMERS is loaded into the target table W_ORG_D, and each record is assigned a unique primary key (ROW_WID). In the table below, change 'IA_CUSTOMERS' to W_ORG_D. Change TS_CUSTOMERS to SOURCE_CUSTOMERS.

TS_CUSTOMERS				IA_CUSTOMERS			
Customer Name	Sales Territory	Customer Contact		Customer KEY	Customer Name	Sales Territory	Customer Contact
ABC Co.	East	Mary	→	101	ABC Co.	East	Mary
XYZ Inc.	West	John	→	102	XYZ Inc.	West	John

Figure 17. Day One: The CUSTOMERS Extract and Data Warehouse Tables

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. Figure 17 shows that records for the two customers, ABC Co., and XYZ inc. have changed when compared with Figure 16. 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 Figure 18, 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.

TS_CUSTOMERS			IA_CUSTOMERS			
Customer Name	Sales Territory	Customer Contact	Customer KEY	Customer Name	Sales Territory	Customer Contact
ABC Co.	East	~~Mary~~ Jane	101	ABC Co.	East	Jane
XYZ Inc.	~~West~~ North	John	102	XYZ Inc.	West	John
			172	XYZ Inc.	North	John

Figure 18. Changes in the Customer Extract Table

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 41](#).

Table 41. Records Before a Wrapper Session

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 42](#).

Table 42. Records After a Wrapper Session

W_CUSTOMER					
Customer Name	Sales Territory	Customer Contact	Effective From	Effective To	Current
ABC	East	Jane	1/1/2001	1/1/3714	Y
XYZ	West	John	1/1/2001	1/1/2002	N
XYZ	North	John	1/1/2002	1/1/3714	Y

In the previous paragraph, the wrapper session corrected the effective to dates and current flag. However, if the record's dates had been correct, the wrapper mapping would simply have set the current flag as needed, because its logic is set to check dates and flags and only adjust columns that contain discrepancies. Finally, if your source system does not supply any Type II information, you may disable the wrapper session completely; in this case all Type II work is handled by the Analytics Data Interface maplet.

About Document, Local, and Global Currencies

Currency lookups are required because your business may have transactions involving multiple currencies. To create a meaningful report, you have to use a common currency. The Oracle Business Analytics Warehouse provides a means for converting a variety of currencies, as well as prepackaging the following three currency types for each amount stored in the Oracle Business Analytics Warehouse:

- **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.
- **Local currency.** The currency in which the financial books, including the transaction, are closed. For example, if your business organization is located in France and orders a part from a supplier in Britain, it may pay in British pounds, but it closes its books in French francs. In this case the local currency for the transaction is French francs and the document currency for the transaction is British pounds. The local currency is useful when each business unit of the enterprise creates its own internal reports. For example, your Japanese site may produce internal reports using Japanese yen, while your United States site may produce internal reports using United States dollars.
- **Global currencies.** Out of the box, Oracle BI Applications provides three global currencies, which are the standard currencies used by your entire enterprise. For example, if a multinational enterprise has its headquarters in the United States, its group currency is probably U.S. dollars. 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 and global 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.

Process for Configuring Currencies

Depending on how your organization handles currency, some of the following currency configuration points may apply to you. For example, you may find it necessary to reconfigure the global currencies, or you may need to reconfigure the way exchange rates are handled. The following sections provide procedures for configuration points related to these currency types.

To configure currencies, perform the following tasks:

- [Extracting Exchange Rates from a Custom Table on page 228](#)
- [Configuring Global Currencies and Exchange Rate Types on page 228](#)

Extracting Exchange Rates from a Custom Table

This task is a step in the [Process for Configuring Currencies on page 227](#).

If you do not maintain exchange rates in your ERP tables, but instead, you maintain them in your custom tables, then you can load them into the W_EXCH_RATE_G table of the Oracle Business Analytics Warehouse by creating a new mapping. The easiest way to create the mapping is to copy an existing exchange rate extraction. After you copy it, you can modify the mapping to work with your custom tables.

Configuring Global Currencies and Exchange Rate Types

This task is a step in the [Process for Configuring Currencies on page 227](#).

You can configure the three global currencies, as well as the exchange rate types you want to use when converting into each of these three global currencies. Also, you can configure the default exchange rate type you want to use when converting.

To configure global currencies and exchange rate types

- 1 Log into DAC
- 2 Navigate to your custom container.
- 3 Click on the 'Source System Parameters' tab
- 4 Locate the following parameters: \$\$GLOBAL1_CURR_CODE, \$\$GLOBAL1_RATE_TYPE, \$\$GLOBAL2_CURR_CODE, \$\$GLOBAL2_RATE_TYPE, \$\$GLOBAL3_CURR_CODE, \$\$GLOBAL3_RATE_TYPE, \$\$DEFAULT_LOC_RATE_TYPE.
- 5 For each of the paramter, in the 'value' box, set your own values.

Handling European Monetary Union (EMU) Currencies

For transactions that take place after the Euro effective date (January 01, 1999), the published exchange rates involving *individual* EMU member currencies may not be available. To address the need for conversions between the individual currency of an EMU member and any other currency, you may need to set up a mechanism for retrieving these rates. The Oracle Business Analytics Warehouse prepackages two methods for handling European Monetary Union currencies:

- Flat file (file_xrates_emu.csv)

- Expression transformation (EXP_CURR_CONVERSION_TRANSFORM).

Each of the two methods is discussed in the following sections.

Using the Flat File to Supply EMU Exchange Rates

The flat file contains currency conversion rates for all combinations of EMU currencies (including individual EMU member currencies as well as the joint EMU currency, the Euro). This file includes the following combinations:

- **EMU to EMU.** This is the conversion from the individual currency of one EMU member country to another.
- **EUR to EMU.** This is the conversion from the joint EMU currency, the Euro, to the individual currency of an EMU member.
- **EMU to EUR.** This is the conversion from the individual currency of an EMU member to the joint EMU currency, based on the Euro.

If you use this method, the Oracle Business Analytics Warehouse assumes that the source system supplies currency conversion rates between individual EMU member currencies and national currencies for countries outside of the EMU. In addition, this method does not conform to the rounding rules set by the European Monetary Union.

To use this method, you must upload the exchange rates from this flat file to the W_EXCH_RATE_G table. After the exchange rates are available in the W_EXCH_RATE_G table, the load mapping can look up these exchange rates from the W_EXCH_RATE_G table and convert currency as required.

To load the individual EMU member currency exchange rates using the flat file

- 1 Before you load the exchange rates from the flat file, if necessary, change the default values in the flat file for the parameters DATASOURCE_NUM_ID, XRATE_TYPE_CODE, XRATES_TYPE_DESC, and integration ID.
 - The XRATE_TYPE_CODE and XRATE_TYPE_DESC default is Corporate.
 - The integration ID uses the XRATE_TYPE_CODE as part of its definition. The Key ID is defined as the concatenation of the From Currency Code, To Currency Code, Exchange Rate Type Code, and the Effective From Date. Therefore, if you change the Exchange Rate Type Code, then you must also modify the Key ID accordingly.
- 2 Create the session for the M_F_XRATES_EXTRACT mapping, using the file_xrates_emu.csv flat file as the source system file.
- 3 Run the extract session to load the flat file exchange rates into the W_EXCH_RATE_GS staging table.
- 4 Create a session for the M_F_XRATES_LOAD mapping.
- 5 Run the load session to load the exchange rates into the W_EXCH_RATE_G table.

About the EXP_CURR_CONVERSION_TRANSFORM Expression Transformation to Supply EMU Exchange Rates

The Expression transformation contains currency conversion logic for converting any combination of EMU currency (including individual EMU member currencies as well as the Euro) and other currencies (OTH) outside of the EMU. This logic applies to the following combinations:

- **EMU to EMU.** This is the conversion from the individual currency of one EMU member country to another.
- **EUR to EMU.** This is the conversion from the joint EMU currency, the Euro, to the individual currency of an EMU member.
- **EMU to EUR.** This is the conversion from the individual currency of an EMU member to the joint EMU currency, the Euro.
- **OTH to EUR.** This is the conversion from any other currency outside the EMU to the Euro.
- **EUR to OTH.** This is the conversion from the Euro to any other currency outside the EMU.
- **EMU to OTH.** This is the conversion from the individual currency of an EMU member country to another currency outside the EMU.
- **OTH to EMU.** This is the conversion from any other currency outside the EMU to the individual currency of an EMU member.

Unlike the flat file method, the transformation handles conversions between EMU currencies and currencies for countries outside the EMU. In addition, this method conforms to the rounding rules set by the European Monetary Union.

The transformation performs the conversion in different ways, depending on the types of input and output currencies. In the transformation, there are ten sets of amount input fields; each set consists of a document amount, local amount, and group amount field. In this discussion, document currency is referred to as the *From* currency and global and local currencies as the *To* currencies. Source-supplied document amounts are output as the same value. However, if the local and global amounts are not supplied, then a particular currency conversion process occurs. There are six different conversion processes that could occur. They are described in the list that follows:

- **Non-EMU to Non-EMU.** If the From Currency and To Currency are not EMU currencies, the following logic is used:
 - The supplied exchange rate is used to compute the new amount.
 - If the exchange rate is not supplied as an input to the transformation, the exchange rate is retrieved using a lookup to the W_EXCH_RATE_G table. The new amount is then calculated using the looked up exchange rate and the From Currency amount.
- **EMU to EMU.** If the From Currency and To Currency are EMU currencies, the following logic is used:
 - Euro-triangulation is used to derive the new amount. Please note that the Euro-triangulation logic only applies if the exchange rate date is later than or equal to the Euro effective date.

- If the exchange rate date is earlier than the Euro effective date, this case is treated as a Non EMU to Non EMU currency conversion case.

The Euro-triangulation logic is used to convert one EMU currency to another EMU currency (EMU to EMU). Again, this logic is only applicable if the exchange rate date is equal to or later than the Euro effective date. The Euro-triangulation logic is as follows:

- Convert from one national denomination (EMU_DOC) to its Euro equivalent (EUR). For example, $EUR = EMU\_DOC / (EUR\text{-}TO\text{-}EMU\_DOC \text{ conversion rate})$.
- Round the previous step's result to the nearest three decimal points. The default rounding is to three decimal points, but this is configurable.
- Convert the Euro equivalent into the resulting national denomination (EMU_LOC). For example:

$$EMU\_LOC = (Result \text{ from previous step}) * (EUR\text{-}TO\text{-}EMU\_LOC \text{ conversion rate})$$

- **EMU to OTH.** If the From Currency is EMU and To Currency is OTH, the following logic is used in the given order:

- The supplied exchange rate is used to compute the amount.
- If the exchange rate is not supplied and the NONEMU_EMU_TRI_FLAG = Y, then the two-step conversion method is used to derive the amount. The two-step conversion method is defined as follows:
 - First, the From Currency amount is converted to the Euro Currency amount using the fixed EMU conversion rates supplied by the transformation's logic.
 - Second, the Euro Currency amount is converted to the To Currency amount using the appropriate exchange rate.

NOTE: The two-step conversion method is only used if the exchange rate date is greater than or equal to the Euro effective date and the NONEMU_EMU_TRI_FLAG is set to Y.

- If the NONEMU_EMU_TRI_FLAG is set to N, this is treated as a Non EMU to Non EMU currency conversion case.
- **OTH to EMU.** If the From Currency is OTH and the To Currency is EMU, then the following logic is used:
 - The supplied exchange rate is used to compute the To Currency amount.
 - If the exchange rate is not supplied and the NONEMU_EMU_TRI_FLAG = Y, then the two-step conversion method is used to derive the amount. The two-step conversion consists of the following two steps:
 - First, the From Currency amount (OTH) is converted to the Euro Currency amount using the exchange rate from the W_EXCH_RATE_G table.
 - Second, the Euro Currency amount is converted to the EMU Currency amount using the fixed Euro conversion rate available in the transformation. Please note that the two-step conversion method only applies if the NONEMU_EMU_TRI_FLAG = Y.
 - If the NONEMU_EMU_TRI_FLAG = N, this is treated as a Non EMU to Non EMU currency conversion case.

- **EUR to EMU.** If the From Currency is EUR and the To Currency is EMU, the following logic is used:
 - If the exchange rate date is greater than or equal to the Euro effective date, then the From Currency amount is converted to the To Currency amount using the fixed Euro conversion rates supplied by the transformation's logic.
 - In all other cases, this is treated as a Non-EMU to Non-EMU currency conversion case.
- **EMU to EUR.** If the From Currency is EMU and the To Currency is EUR, the following logic is used:
 - If the exchange rate date is greater than or equal to the Euro effective date, the From Currency amount is converted to the To Currency amount using the fixed Euro conversion rates supplied by the transformation's logic.
 - If the exchange rate date is less than the Euro effective date, this is treated as a Non EMU to Non EMU currency conversion case.

To use an Expression transformation, you must add the transformation to every load mapping that requires any of the previously listed types of currency conversions. See the `M_F_EURO_TRIANG_EXP_USAGE_EXAMPLE` mapping for an example of how to incorporate an Expression transformation. This mapping is located in the Configuration for Universal Source folder.

To add the transformation to a load mapping

- 1 Open the applicable source system configuration folder.
- 2 Select Tools > Transformation Developer.
- 3 Open the Oracle BI folder and drag and drop the `EXP_CURR_CONVERSION_TRANSFORM` Expression transformation to create a shortcut.

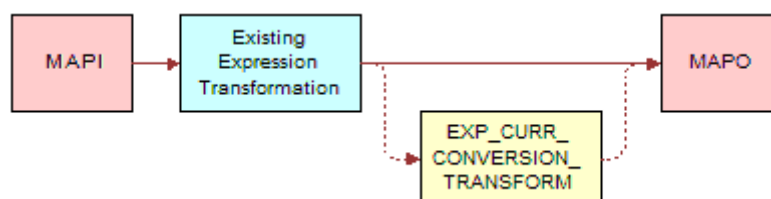
You can place the shortcut in the Transformations folder contained in the applicable source system configuration folder.

- 4 Select Maplet Designer.
- 5 Open the applicable Source Adapter maplet.
- 6 Add `LKP_XRATES` from the Transformations folder.

NOTE: `LKP_XRATES` is a lookup to the `W_EXCH_RATE_G` table. If the Transformations folder does not have a lookup to `W_EXCH_RATE_G`, you must make a shortcut to `LKP_XRATE` in the Siebel Analytics Enterprise Applications folder.

- 7 Add the shortcut to the `EXP_CURR_CONVERSION_TRANSFORM` Expression transformation to the Source Adapter maplet.

Place the transformation between the load mapping's existing Expression transformation and MAPO, as shown in the following figure.



- 8 Reconnect the ports as necessary so that the data flows through the transformation and into the MAPO.

Table 43 lists all of the prepackaged exchange rates used by the transformation. Table 43 only lists exchange rates from EMU currency to Euro. Using these exchange rates, the transform can calculate any of the five scenarios (EMU to EMU, EUR to EMU, EMU to EUR, EMU to OTH, and OTH to EMU). For EMU to OTH, OTH to EMU, and EMU to EMU, the transformation converts the From Currency to the Euro and then from the Euro to the To Currency. For that reason, Table 43 only has EMU currency to the Euro conversion rates.

Table 43. Prepackaged Exchange Rates in Expression Transformation

Currency Code	Currency Name	Effective From Date	Conversion Rate for One Euro
BEF	Belgian Francs	01/01/1999	40.3399
DEM	German Marks	01/01/1999	1.95583
ESP	Spanish Pesetas	01/01/1999	166.386
FRF	French Francs	01/01/1999	6.55957
IEP	Irish Púnt	01/01/1999	.787564
ITL	Italian Lira	01/01/1999	1936.27
LUF	Luxembourg Francs	01/01/1999	40.3399
NLG	Dutch Guilders	01/01/1999	2.20371
ATS	Austrian Schillings	01/01/1999	13.7603
PTE	Portuguese Escudo	01/01/1999	200.482
FIM	Finnish Markka	01/01/1999	5.94573

You can incorporate additional EMU currency conversion rates into the transformation. When doing so, you must consider whether or not the conversion rate is effective before January 1, 1999 or on/after January 1, 1999. (January 1, 1999 is the date the Euro was enacted.)

To add a new EMU currency with the same effectivity date (January 1, 1999)

- Edit the decode statement for the columns EMU_TO_EURO_CONV_FACT_DOC, EMU_TO_EURO_CONV_FACT_LOC, and EMU_TO_EURO_CONV_FACT_GRP, to add the new currency and its conversion rate.

For example, if you are adding EMU1, where the Euro-to-EMU1 conversion rate is 2.34567, then you would modify the EMU_TO_EURO_CONV_FACT_DOC column definition as follows:

```
EMU_TO_EURO_CONV_FACT_DOC = IIF (ISNULL (EXT_XRATE_LKP_DT) OR
TO_NUMBER(TO_CHAR(EXT_XRATE_LKP_DT, 'J')) >= TO_NUMBER(TO_CHAR(TO_DATE('01/01/
1999', 'MM/DD/YYYY'), 'J')), DECODE(EXT_DOC_CURR_CODE, 'BEF', 40.3399000000,
'NLG', 2.2037100000, 'ATS', 13.7603000000, 'PTE', 200.4820000000, 'FIM',
```

```
5.9457300000, 'IEP', 0.7875640000, 'LUF', 40.3399000000, 'FRF', 6.5595700000,  
'ITL',1936.2700000000, 'DEM', 1.9558300000, 'ESP', 166.3860000000,  
'EMU1',2.3456700000, 'EUR', 1.0000000000, NULL), NULL)
```

To add a new EMU currency with a different effectivity date than January 1, 1999

- Edit the decode statement for the columns EMU_TO_EURO_CONV_FACT_DOC, EMU_TO_EURO_CONV_FACT_LOC, and EMU_TO_EURO_CONV_FACT_GRP, to add the new currency, its conversion rate, and the effectivity date logic.

For example, if you are adding EMU2, where the Euro-to-EMU2 conversion rate is 3.45678, and the exchange rate effectivity date is 01/01/2001, then you would modify the EMU_TO_EURO_CONV_FACT_DOC column definition as follows:

```
EMU_TO_EURO_CONV_FACT_DOC = IIF (ISNULL (EXT_XRATE_LKP_DT)  
OR(TO_NUMBER(TO_CHAR(EXT_XRATE_LKP_DT, 'J')) >= TO_NUMBER(TO_CHAR(TO_DATE('01/01/  
1999', 'MM/DD/YYYY'), 'J')) AND EXT_DOC_CURR_CODE!= 'EMU2') OR  
TO_NUMBER(TO_CHAR(EXT_XRATE_LKP_DT, 'J')) >= TO_NUMBER(TO_CHAR(TO_DATE('01/01/  
2001', 'MM/DD/YYYY'), 'J'))),DECODE(EXT_DOC_CURR_CODE, 'BEF', 40.3399000000,  
'NLG', 2.2037100000, 'ATS', 13.7603000000, 'PTE',200.4820000000, 'FIM',  
5.9457300000, 'IEP', 0.7875640000, 'LUF', 40.3399000000, 'FRF', 6.5595700000,  
'ITL',1936.2700000000, 'DEM', 1.9558300000, 'ESP',166.3860000000, 'EMU2',  
3.4567800000, 'EUR', 1.0000000000, NULL), NULL)
```

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:

- [Codes Lookup on page 234.](#)
- Dimension Key Lookups. For more information, see [About Resolving Dimension Keys on page 237.](#)

Codes Lookup

Some source systems use intelligent codes that are intuitively descriptive, such as HD for hard disks, while other systems use nonintelligent 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 nonintelligent 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:

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.

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 44](#) describes these columns.

Table 44. 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]. [Figure 19](#) shows an example of a code mapping in PowerCenter Mapping Designer.

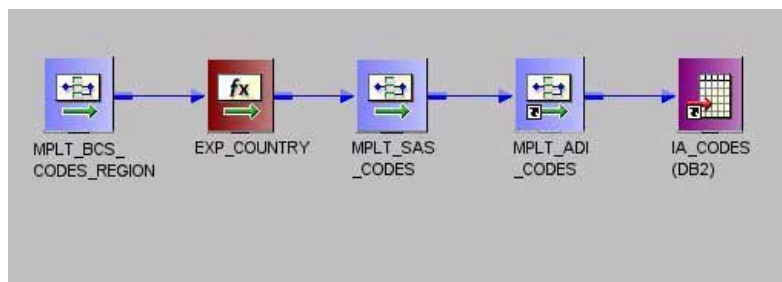


Figure 19. Sample Codes Mapping: SDE_PSFT_CodeDimension_Region

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.

Configuring Extension Column Code Description Lookups

You can configure dimension and fact load sessions to perform specific lookups by editing the category of the data to be extracted from the W_CODES table and loading the code information into a target table. If the code and code name do not exist in the W_CODES table, then you must add them to the table. To configure the lookup, create a session override; do not modify the load mapping in the load mapping.

To configure sessions for lookups

- 1 In PowerCenter Workflow Manager, open the applicable source system configuration folder.
- 2 Open the Edit Tasks box.
- 3 In the Transformations tab, edit the SQL statement for the lookup.

For example, you may wish to edit the following lookup:

```
MPLT_ADI_SUPPLIERS.LKP_SPLR_ATTR1
```

- 4 Edit the SQL statement to use the desired code category.
- 5 Edit the SQL statement from 'GENERIC' to the category you wish to use for the lookup.

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.

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 45](#).

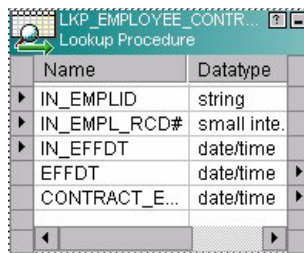
Table 45. Columns Used in the load mapping Dimension Key Lookup

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 [Figure 20](#), 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 Figure 20, you can see the effective dates used to provide the effective period of employee contracts.



Name	Datatype
IN_EMPLID	string
IN_EMPL_RCD#	small inte.
IN_EFFDT	date/time
EFFDT	date/time
CONTRACT_E...	date/time

Figure 20. Effective Dates Used in Slowly Changing Dimension

About Domain Values

The Oracle Business Analytics Warehouse foundation comprises a data model that accommodates data from disparate source systems. Data is sourced from operational systems and systematically molded into a source-independent format. After the data is source independent, it can then be used to create key metrics for analytic reporting, so that metric calculations are not source dependent. This clear separation allows you to swap source systems or integrate additional source systems without having to reconfigure the metric calculations to accommodate each source system's requirements.

One method for transforming source data into a source-independent format is to convert the source-supplied values to domain values. Domain values are a set of distinct values used to calculate prepackaged metrics. These values are provided by the Oracle Business Analytics Warehouse to allow you to create metric calculations independent of source system values.

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.

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 [Configuring the Domain Value Set Using PowerCenter Designer on page 244](#).

Figure 21 illustrates how the source values are converted to the domain values—HIR and REH.

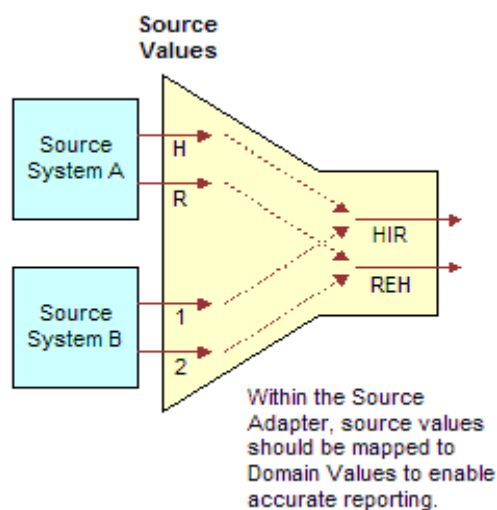


Figure 21. Source Values Translated to Domain Values

Figure 22 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.

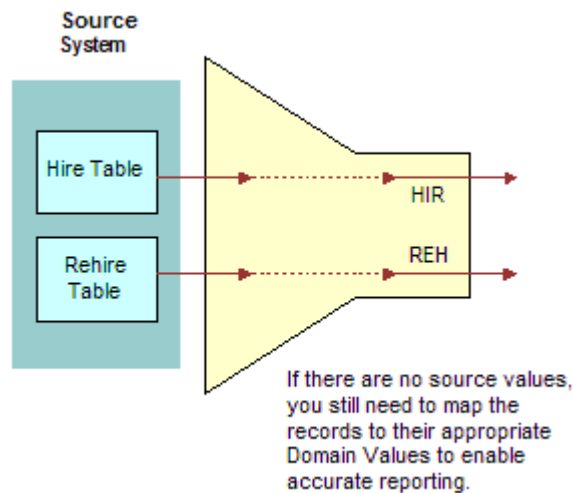


Figure 22. Records Mapped to Domain Values

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 Figure 23.

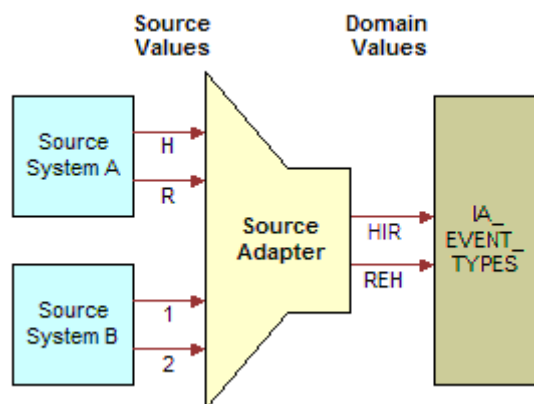


Figure 23. Domain Value Loading Oracle Business Analytics Warehouse Table

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:

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)
```

Rehires Ratio

This metric determines the ratio of rehires to all employees hired during a specified period. The calculation is:

```
CASE WHEN SUM(CASE WHEN CMMNEVTP.W_EVENT_GRP_CODE IN ('REH', 'HIR') THEN EVNT.EVENT_CNT ELSE 0 END) = 0 THEN 0 ELSE SUM(CASE WHEN CMMNEVTP.W_EVENT_GRP_CODE IN ('REH') THEN EVNT.EVENT_CNT ELSE 0 END)/SUM(CASE WHEN CMMNEVTP.W_EVENT_GRP_CODE IN ('REH', 'HIR') THEN EVNT.EVENT_CNT ELSE 0 END) END
```

New Hire Count

This metric counts the headcount 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)
```

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)
```

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)
```

Special Disabled Veteran Headcount - 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)
```

Vietnam Era Veteran Headcount - 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 [Figure 24](#).

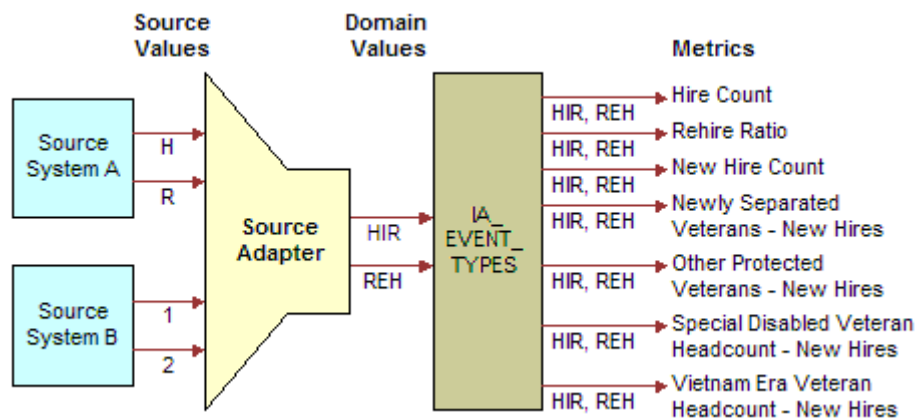


Figure 24. Domain Values Used in Metric Definitions

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.

Configuring the Domain Value Set with CSV Worksheet Files

Domain values are a set of distinct values used to calculate prepackaged metrics. These values are provided by Oracle Business Analytics Warehouse to allow you to create metric calculations independent of source system values. Oracle Business Analytics Warehouse provides CSV worksheet files to map source system values to domain values.

You can add to these worksheet files if you need extra source system values and map them to domain values. You can also modify the worksheet files if you need to customize the domain values. You can use an existing domain value if you want to change the preconfigured metrics. Otherwise you can create a new domain value and create new metrics based on this domain value.

The source system values that are not mapped to a domain values in the CSV worksheet files have a question mark (?) as the domain value in the Oracle Business Analytics Warehouse. These values do not affect the domain values metrics.

If there are no worksheet files to map the source system values to the domain values, you need to modify the domain values using PowerCenter Designer. For more information on configuring domain values using PowerCenter Designer, see [Configuring the Domain Value Set Using PowerCenter Designer on page 244](#).

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 *Oracle Business Analytics 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 *Oracle Business Analytics Warehouse Data Model Reference*.

- 4 Open the CSV worksheet file in the ...\\Informatica\\SrcFiles folder.

- 5 Edit the file to map your source values to the existing domain values.
Alternately, if you want to add additional domain values, add them in this worksheet file.
- 6 Save and close the worksheet file.

Configuring the Domain Value Set Using PowerCenter Designer

If there are no worksheet files to map the source system values to the domain values, you need to modify the values using PowerCenter Designer. For more information on configuring the domain value set with CSV worksheet files, see [Configuring the Domain Value Set with CSV Worksheet Files on page 243](#).

Configuring the domain value set for a particular column, using PowerCenter Designer, entails one or both of the following activities:

- Mapping source-specific values to domain values
- Adding more domain values to the prepackaged set of values

Regardless of which activity you choose, the configuration occurs in the Expression transformation of the applicable Source Adapter mapplet. The following procedure shows how to configure the Expression transformation to change the domain values.

To map source values to domain values using PowerCenter Designer

- 1 Identify all the Oracle Business Analytics Warehouse table columns that use domain values.
For a list of columns that use domain values, see the *Oracle Business Analytics 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 *Oracle Business Analytics Warehouse Data Model Reference*.
- 4 In PowerCenter Designer, open the applicable Source Adapter mapplet.
- 5 Open the Expression transformation.
- 6 Locate the applicable port's expression so that you can modify it.
- 7 Edit the port's expression to map your source values to the existing domain values.
Alternately, if you want to add additional domain values, add them in this same expression.
- 8 Save and validate your changes to the repository.

Configuring Conformed Dimensions

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

- [Configuring Conformed Dimensions for Universal Source on page 245](#)

Configuring Conformed Dimensions for Universal Source

This section provides configuration procedures for modifying dimensions that are loaded using a universal business adapter.

Product Effective Dates in the Products Dimension

The Oracle Business Analytics Warehouse stores product Effective To (SRC_EFF_TO_DT) and From (SRC_EFF_FROM_DT) dates in the Products dimension table, W_PRODUCTS. In addition, the Products dimension stores a Support Withdrawn Date column, SPRT_WITHDRAWN_DT.

By default, the Support Withdrawn Date takes precedence over the product Effective To Date. This prioritization means that if you supply a value for the Support Withdrawn Date column in your flat file upload, the Oracle Business Analytics Warehouse uses that value as the product Effective To value as well, overwriting anything in the SRC_EFF_TO_DT column. You can change this default behavior by modifying the Products Expression in the Universal Source Products Extract mapping.

To modify the product Effective To Date logic for a flat file extract

- 1 In PowerCenter Designer, open the Configuration for Universal Source folder.
- 2 In the SDE_Universal_ProductDimension mapping, open the W_PRODUCT_D expression.
- 3 Edit the logic for the SRC_EFF_TO_DT_OUT port.
- 4 Validate and save your changes.

10

Configuring Oracle Business Analytics Warehouse for Oracle 11i

Regardless of which application you are implementing, there is some general configuration information that is specific to Oracle 11i. In this chapter, you learn about each of these points.

This chapter contains the following topics:

- [Configuring the Group Currency Code for Oracle 11i on page 247](#)
- [About Adding Oracle Flex Fields to the Data Model on page 247](#)
- [Mapping Source Customer Hierarchies to the Customers Dimension Table on page 248](#)
- [Configuring Product Categories on page 250](#)
- [Modifying the DATASOURCE_NUM_ID on page 251](#)
- [Modifying the Last Extract Date on page 252](#)

Configuring the Group Currency Code for Oracle 11i

Oracle 11i provides document and local currencies, but not a group currency. Therefore, the Oracle Business Analytics Warehouse uses a flat file called `grp_curr.csv` that contains a group currency code, which is loaded into the data warehouse during the initialization phase. It is initially set as USD for United States dollars. However, if your group currency is different, you must edit the currency code in the flat file before you load it. For example, if your group currency is the European Euro, then you might set the currency code to EUR.

To configure the group currency code for Oracle 11i

- 1 Open the `grp_curr.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.
- 2 Edit the group currency code.
- 3 Save and close the file.

About Adding Oracle Flex Fields to the Data Model

Oracle Applications have two kinds of flex columns:

- **Key flex fields.** Key flex fields are key fields that are required by Oracle 11i. You can modify their definitions when configuring Oracle Applications.
- **Descriptive flex fields.** Descriptive flex fields are extension fields in the Oracle Applications database.

Key Flex Fields

The Oracle Business Analytics Warehouse has prepackaged mappings to extract some of the key flex field data. However, if you configure any of these key flex fields other than the default configuration prepackaged by the Oracle Business Analytics Warehouse, you must modify the mappings. The following are the key flex fields preconfigured for extraction:

- **GL Account, account segment only.** If you want to incorporate any other segment, you must modify the extract and load mappings.
- **Territory, segments 1, 2, and 3 only.** These key flex fields are used for Business Organizations Sales Geographical hierarchies only. If you want to incorporate any other segments, you need to modify the extract and load mappings.
- **Product Category, segments 1 and 2 only.** These key flex fields are used for classification only. If you want to incorporate any other segments, you need to modify the extract and load mappings.

If you wish to add other key flex field data to the data model, it is recommended that you use the extension columns available in the tables. For more information on using extension columns, see

[Customizing the Oracle Business Analytics Warehouse on page 171.](#)

Mapping Source Customer Hierarchies to the Customers Dimension Table

Customer hierarchies are typically custom-defined in Oracle Applications. If you want to include this data in the data warehouse, you must perform two customization processes:

- Import the hierarchies into the TI_CUSTOMERS staging table for Oracle 11i.
- Redefine the category lookup so that the new category data is loaded into the Oracle Business Analytics Warehouse.

The two processes are illustrated in [Figure 25](#).

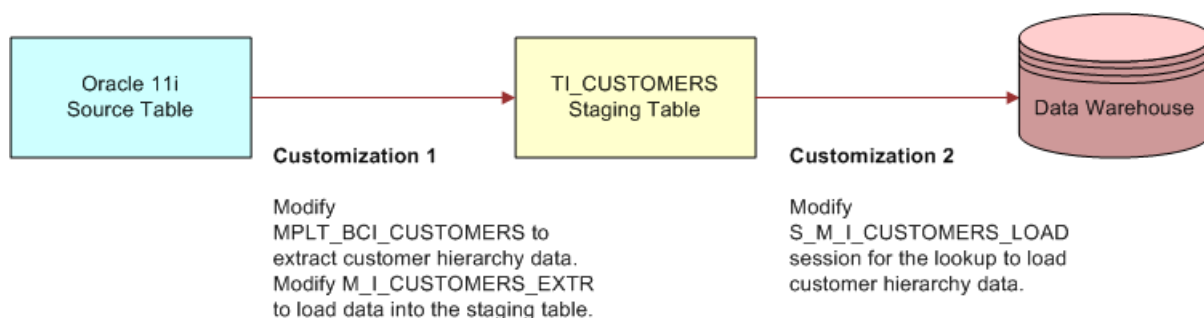


Figure 25. Oracle Applications: Customization Processes for Custom Customer Hierarchies Load

To load the source-defined customer hierarchies into the TI_CUSTOMERS staging table for Oracle 11i, you must first edit the MPLT_BCI_CUSTOMERS Business Component maplet to extract the hierarchy in addition to the customer information. After your Business Component is set up to extract the customer hierarchies, you must verify that the M_I_CUSTOMERS_EXTRACT extract mapping outputs this data to the data warehouse.

To map Oracle-defined customer hierarchies to the Customers dimension table

1 In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.

2 Open the MPLT_BCI_CUSTOMERS Business Component maplet for Oracle 11i.

Modify the maplet to extract the customer hierarchy columns.

If the hierarchies and customers are maintained in the same Oracle Applications source table, then load these columns into the SQL Source qualifier and map them to the Business Component output.

However, if the hierarchies and customers are stored in two different tables, then the Business Component must be modified to include both source tables so that it can include both sets of information.

- Modify the M_I_CUSTOMERS_EXTRACT extract mapping, to map the source customer hierarchy columns to the extension hierarchy columns in the TI_CUSTOMERS staging table.

If the source table has hierarchy codes, but no descriptions associated with these codes, map the Oracle Applications codes to both the hierarchy name columns and the hierarchy code columns. Hierarchy name columns are named as CUST_HIERX_NAME, where X denotes the level of the customer hierarchy.

If the source table has code values, but the corresponding descriptions are in a different source table, you must build new codes mappings that load the data into the W_CODES table.

3 Save your changes to the repository.

NOTE: After you complete the previous process, you must modify a hierarchy lookup in the customer dimension so that the system extracts the new categories.

To configure the category lookup

1 In PowerCenter Workflow Manager, open the Configuration for Oracle Applications v11i folder.

2 Open the S_M_I_CUSTOMERS_LOAD session for Oracle 11i, to open the Edit Tasks box.

3 In the Transformations tab, modify the lookup in the MPLT_ADI_CUSTOMERS.LKP_CUST_HIERX field in the W_CODES table by adding the new category in the SQL statement.

Select the arrow to edit the WHERE clause.

You can use the following statement as a sample of how to structure your SQL statement:

```
WHERE W_CODES.CATEGORY = 'GENERIC' AND W_CODES.LANGUAGE = 'E'
```

For example, if you have mapped something to the CUST_HIER1_CODE, then the SQL to the W_CODES table is now a new category code in place of GENERIC.

- 4 Validate and save your changes to the repository.

Configuring Product Categories

As initially configured, the Oracle Business Analytics Warehouse extracts product categories where the CATEGORY_SET_ID is 2 or 27. However, it is likely that the categories you extract from the source system are different from these prepackaged categories. Therefore, you must reconfigure your product categories by making two customizations:

- Identify and extract only the categories that you want to report against.
- Properly position the data so that it loads into the Oracle Business Analytics Warehouse.

There are two dimension tables that have built-in product hierarchies—the Product and Sales Product dimension tables. These dimension tables share one category staging table. ETL extracts the Product and Sales Product staging tables separately, and then joins these tables with the shared category staging table to load hierarchies. The category extract mapping controls the category sets that are used in the Product and Sales Product dimensions. The load mappings for the Product and Sales Product dimensions specifies which Category Set is used to load to the hierarchy columns.

To configure product category extract from Oracle Applications

- 1 Identify the categories that need to be mapped to the extension hierarchy columns.
These categories are extracted from the source and placed in the hierarchy column specified.
- 2 In PowerCenter Workflow Manager, open the Configuration for Oracle Applications v11i.
- 3 Open the S_M_I_CATEGORY_EXTRACT session with the Task Developer.
- 4 In the Mappings tab, click MPLT_BCI_CATEGORY.SQ_MTL_CATEGORIES.
- 5 In the right pane, scroll down and click SQL Query to edit session SQL override.
- 6 Click the arrow to edit the WHERE clause.

The following statement is an example of how to structure the WHERE clause:

```
WHERE...MTL_CATEGORY_SETS_B.CATEGORY_SET_ID IN (27,2)
```

In this example, the WHERE clause extracts categories where the Category Set ID is 27 or 2.

- 7 Click OK, and then click OK to close the Edit Tasks box.

To modify the default Category Set

- 1 Open the file_parameters_ora11i.csv file in the \$pmserver\srcfiles folder

- 2 Replace the default Category Set ID (27) with your new value.

The following table is an example of the `file_parameters_ora11i.csv` file.

S_M_I_SALES_PRODS_LOAD:CATEGORY_SET_ID	0	0	0	0	S	27
S_M_I_PRODUCTS_LOAD:CATEGORY_SET_ID	0	0	0	0	S	27

- 3 Save and close the file.

To reconfigure the product hierarchy loads

- 1 In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder, and expand the Maplets folder.
- 2 Open the MPLT_SAI_PRODUCTS maplet for Oracle 11i.
- 3 Double-click the EXP_PRODUCTS expression transformation to open the Edit Transformation box.
- 4 In the Ports tab, scroll down to find the hierarchy code port.

Hierarchy levels are named with the following convention `EXT_PROD_HIERX_CODE`, where X denotes the level within the hierarchy. For example, if you want to edit the first level of your hierarchy, you must edit the definition for `EXT_PROD_HIER1_CODE` port. The first two levels are preconfigured as follows:

```
EXT_PROD_HIER1_CODE = IIF(ISNULL(INP_SEGMENT1) OR ISNULL(INP_SEGMENT2), NULL,
INP_SEGMENT1 || '~' || INP_SEGMENT2)
EXT_PROD_HIER2_CODE = INP_SEGMENT1
```

- 5 Modify the expression that defines your hierarchy code.
- 6 Validate and save your changes to the repository.

NOTE: You can configure six hierarchy extension columns in the Oracle Business Analytics Warehouse. To resolve the names for each level, you need to extend the `W_CODES` table with the correct codes when configuring the new hierarchy levels.

Modifying the DATASOURCE_NUM_ID

The `DATASOURCE_NUM_ID` is set to `OAP11i` in the mappings corresponding to the Oracle sessions. The `file_parameters_ora11i.csv` file contains the value for the `$$DATASOURCE_NUM_ID` field for each of the sessions.

After you have modified the file, the next time you run the session, a new `DATASOURCE_NUM_ID` is loaded.

To modify the DATASOURCE_NUM_ID

- 1 Open the parameterfileDW.txt file using Microsoft WordPad or Notepad in the \$pmserver\srcfiles folder.
- 2 Replace the default DATASOURCE_NUM_ID with the new DATASOURCE_NUM_ID.
Perform this action for all applicable sessions.
- 3 Save and close the file.

Modifying the Last Extract Date

By default, the \$\$LAST_EXTRACT_DATE is set to 01/01/1970 for Oracle 11i. However, if you need to modify this date, you can do so by modifying the parameterfileOLTP.txt file. After you have modified the file, the next time you run the session, the new Last Extract date is populated.

To modify the Last Extract date

- 1 Open the parameterfileOLTP.txt file.
- 2 Replace the default Last Extract date with the new Last Extract date.
Perform this action for all applicable sessions.
- 3 Save and close the file.

11 Checklist for Configuring Oracle Business Analytics Warehouse Applications

This chapter describes how to configure certain objects for particular sources to meet your business needs.

It contains the following topics:

- [Checklist for Initializing Oracle Business Analytics Warehouse on page 253](#)
- [Checklist for Configuring Oracle BI Applications Consumer on page 253](#)
- [Checklist for Configuring Oracle BI Applications Consumer on page 254](#)
- [Checklist for Configuring Oracle Workforce Analytics on page 256](#)
- [Checklist for Configuring Oracle Financial Analytics on page 257](#)
- [Checklist for Configuring Supply Chain Analytics on page 259](#)

Checklist for Initializing Oracle Business Analytics Warehouse

This section contains Oracle Business Analytics Warehouse mandatory configuration points for Oracle 11i, universal source, and PeopleSoft 8.4:

- **Time Dimension files.** It is important that the fiscal calendar-related files are in the \$pmserver\SrcFiles folder, and have been configured correctly. For more information about time dimensions, see [Setting Up The Time Dimension on page 126](#)).

Checklist for Configuring Oracle BI Applications Consumer

This section contains the mandatory and optional configuration points for the Oracle BI Applications Consumer application for Universal Source and post-load processing (PLP).

Checklist for Configuring Oracle BI Applications Consumer for Universal Source

This section contains the Oracle BI Applications Consumer application configuration points that are specific to universal source.

The following section contains mandatory Oracle Contact Center Telephony Analytics configuration points:

- **Dimension Key Resolution.** When creating the source files, the data and the format of the reference to the Dimension data (foreign key ID column) must match the INTEGRATION_ID in the corresponding dimensional source file. This is critical for accurate resolution of the foreign keys on the fact record when loading the data warehouse. For more information on Dimension key resolution, see [About the Dimension Key Resolution Process for Universal Source on page 264](#).
- **Representative Activities spanning a date boundary.** 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. For more information on dates and setting up the Representative Activities table, see [Setting Up the Representative Activities Table on page 269](#).
- **Setting the effective date ranges for Benchmark and Targets.** When configuring the source files for the Benchmarks and Targets fact table, you need to make sure that the targets and benchmarks cover the time periods for the capture of the Contact Center Performance and Representative Activity data. For more information on setting the effective date ranges for Benchmark and Targets, see [Setting Up the Benchmarks and Targets Table on page 272](#).

The following section contains optional Oracle Contact Center Telephony Analytics configuration points.

- **Configuring flags.** 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 have logic to assign a default value to the flag while reading from the source files. For more information on configuring flags, see [Configuring Flags for Oracle BI Applications Consumer on page 276](#).
- **Excluding calls from the Answered Contact Count.** You may choose not to count calls which are completed in the IVR as an answered call. To do this you need to flag such calls with a new contact status. For more information excluding calls from the Answered Contact Count, see [Setting Up the Contact Center Performance Table on page 271](#).

Checklist for Configuring Oracle BI Applications Consumer for post-load processing

The following section contains an optional configuration point for Oracle BI Applications Consumer that is specific to post-load processing.

Excluding representative data from the Contact Representative aggregate tables. For more information on excluding representative data from the Contact Representative aggregate tables, see [Excluding Representative Data for Post-Load Processing on page 277](#).

Checklist for Configuring Oracle BI Applications Consumer

This section contains optional configuration points for the Oracle BI Applications Consumer application for Oracle 11i.

Checklist for Configuring Oracle BI Applications Consumer for Oracle 11i

This section contains optional configuration points for Oracle BI Applications Consumer that are specific to Oracle 11i:

- **Including nonbooked lines in the Sales Booking Line fact table.** By default, only booked lines are brought over to the Sales Booking Line fact table. You can change this to include Nonbooked Lines. For more information on configuring the handling of Booked and Nonbooked Orders in the Order Lines and Bookings Table, see [Configuring Sales Order Lines Data Storage on page 339](#).
- **Tracking changes in Booking Lines.** By default, only the changes in Order Amount, Quantity, Line ID, and Warehouse are tracked in the Booking Lines table (W_SALES_BOOKING_LINE_F). 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. For more information on viewing the Data Warehouse changes by Salesperson ID, see [About Tracking Attribute Changes in Bookings on page 341](#).
- **Loading bookings at the Schedule Line Level instead of the Sales Order Line level.** You can configure the load process to load bookings at the Sales Schedule Line level instead of the Sales Order Line level. For more information on loading bookings at the Schedule Line level, see [Configuring Sales Schedule Lines Data Storage on page 343](#).
- **Setting up early and late tolerances for shipping.** To define early or late shipments with reference to the scheduled pick date, you need further configurations. For more information on defining early and late tolerances for shipments, see [Configuring Early and Late Tolerances for Shipments on page 345](#).
- **Including incomplete Sales Invoices.** By default, the Siebel Enterprise Sales Analytics application is configured to extract completed sales invoices when performing the Sales Invoice data extract. To extract incomplete sales invoices you would need further configurations. For more information on configuring Sales Invoice Lines data storage, see [Configuring Sales Invoice Lines Data Storage on page 346](#).
- **Including closed orders for backlog calculations.** By default, only orders with status Open are included in the backlog calculations. To include the closed orders, you would need further configurations. For more information on adding closed orders to backlog calculations, see [Configuring Different Types of Backlog Calculations on page 346](#).
- **Configuring order types for backlog calculations.** By default, all order types are included in the backlog calculations. To exclude certain order types you would need further configurations. For more information on configuring order types for backlog calculations, see [Configuring Different Types of Backlog Calculations on page 346](#).
- **Configuring the backlog history period date.** This configuration allows you to change the default monthly backlog snapshot to a different grain—for example weekly, daily, and so on. For more information on configuring the backlog period date, see [Accounting for Negative Values in Orders, Invoices, and Picks on page 348](#).

- **Setting the negative sign for the Order and Invoice Lines.** By default, the Oracle Business Analytics Warehouse does not use negative values in the quantity or amount columns for the W_SALES_IVCLNS table or the W_SALES_ORDLNS table. You can configure mapplets to account for negative values. For more information on accounting for negative values for Orders, Invoices, and Picks, see [Accounting for Negative Values in Orders, Invoices, and Picks on page 348](#).
- **Domain Values.** For a list of CSV worksheet files and domain values for Oracle BI Applications Consumer for Oracle 11i, see [Domain Values and CSV Worksheet Files for Oracle Order Management and Fulfillment Analytics Option on page 360](#).
- **Processing Deletes.** If you want to track deletes in a Sales Fact table, create a new subject area and add to that the table(s) using the out of the box sales subject areas as reference. Go to Configuration Tags in the DAC and find "Enterprise Sales Identify and Soft Delete Tasks". Add the subject area that you created. Make sure that the "Inactive Flag" is not checked. Design the subject area and include that in your custom execution plan.
- **Post load processing.** Oracle EBS has interface programs that update the Order tables with picking/shipping and invoicing information. The out of the box ETL relies on these interface programs to source related data. If you do not run these interface programs in Oracle EBS, you can use the packaged post load ETL steps to update that information from the data available in the various warehouse tables. Create a new subject area and add to that the table(s) using the out of the box sales subject areas as reference. Go to Configuration Tags in the DAC and find "Enterprise Sales PLP Tasks". Add the subject areas that you created. Make sure that the "Inactive Flag" is not checked. Design the subject areas and include them in your custom execution plan.

Checklist for Configuring the Siebel Enterprise Sales Analytics for Post-Load Processing

This section contains optional configuration points for Oracle BI Applications Consumer that are specific to post-load processing.

- **Aggregating Oracle BI Applications Consumer tables.** To aggregate the Sales Invoice Lines and Sales Order Lines tables, see [Process of Aggregating Oracle Order Management and Fulfillment Analytics Option Tables on page 349](#).
- **Tracking multiple products sold as one package.** This configuration allows the user to set up the Order Line Key ID to reference all products sold in a bundle. For more information on tracking multiple products sold as one package, see [About Tracking Multiple Products for Oracle Order Management and Fulfillment Analytics Option on page 353](#).
- **Adding dates to the Order Cycle Time table.** To add more dates, you need to understand how the Order Cycle Times table is populated. For more information on adding dates to the Cycle Time table load, see [Adding Dates to the Order Cycle Time Table for Post-Load Processing on page 354](#).

Checklist for Configuring Oracle Workforce Analytics

This section contains the mandatory and optional configuration points for Oracle Workforce Analytics application for Oracle 11i and post-load processing.

Checklist for Configuring Oracle Workforce Analytics for Oracle 11i

This section contains the configuration points for Oracle Workforce Analytics that are specific to Oracle 11i.

The following section contains mandatory configuration points for Oracle Workforce Analytics.

- **Configuring Workforce Operations.** Here, you have to configure the Employee and Job dimensions for U.S. Statutory Compliances. Apart from that, you also configure the Address and Phone Types for the Employee dimension. For more information on configuring the Workforce Operations, see [Process of Configuring Workforce Operations for Oracle 11i on page 364](#).
- **Configuring domain values and CSV worksheet files.** Here, you need to configure the CSV files in Oracle Workforce Analytics by mapping domain values to columns. For more information on Configuring Domain Values and CSV Worksheet Files, see [Configuring Domain Values and CSV Worksheet Files for Oracle Workforce Analytics on page 384](#).

The following section contains an optional configuration point for Oracle Workforce Analytics.

- **Configuring Workforce Payroll.** Here, you modify the Workforce Payroll Filters and improve ETL performance for Workforce Payroll. For more information on configuring the Workforce Payroll, see [Process of Configuring Workforce Payroll for Oracle 11i on page 374](#).

Checklist for Configuring Oracle Workforce Analytics for Post-Load Processing

This section contains the configuration points for Oracle Workforce Analytics that are specific to post-load processing.

The following section contain an optional configuration point for Oracle Workforce Analytics.

- **Aggregating the Payroll table.** Here, you aggregate the Payroll table in Oracle Workforce Analytics at the time-grain that you want (the default is monthly). If you want to consider a different time-grain aggregation (for example, Quarter or Year), you have to change the configuration (for more information on aggregation in the Payroll table, see [Aggregating the Payroll Table for Oracle Workforce Analytics on page 378](#)).

Checklist for Configuring Oracle Financial Analytics

This section contains the mandatory and optional configuration points for the Financial applications for Oracle 11i and PeopleSoft 8.4. The Financial applications consist of Oracle General Ledger and Profitability Analytics, Oracle Payables Analytics, Oracle Receivables Analytics, and Oracle General Ledger and Profitability Analytics.

Checklist for Configuring Oracle Financial Analytics for Oracle 11i

This section contains the configuration points for Oracle Financial Analytics applications that are specific to Oracle 11i.

The following section contains mandatory configuration points for Oracle Financial Analytics:

- **Extracting Data Posted at the Detail-Level.** By default, the Oracle Business Analytics Warehouse assumes that the posting from your journal to your General Ledger is done at summary level and that references are maintained in the General Ledger for AP and AR subledgers. Configuration is required if import references are not maintained in the General Ledger and the posting from subledgers is at the detail level. For more information on extracting data posted at the detail-level, see [Extracting Data Posted at the Detail-Level for Oracle 11i on page 317](#).
- **Mapping General Ledger Analytics account numbers to group account numbers.** You need to map General Ledger Analytics account numbers to group account numbers. For more information on mapping General Ledger accounts numbers to group account number, see [Mapping Oracle General Ledger and Profitability Analytics Account Numbers to Group Account Numbers on page 319](#).

The following section contains optional configuration points for Oracle Financial Analytics:

- **Configuring the Set of Books ID.** By default, the Oracle Business Analytics Warehouse extracts data for all set of books. Configuration is required to extract data for a certain set of books only.
- **Configuring posted or unposted transactions.** By default, Oracle Business Analytics Warehouse extracts posted transactions. You can configure the Oracle Business Analytics Warehouse to extract unposted transactions as well as posted transactions.
- **Configuring General Ledger account hierarchies.** Additional configuration is required to make sure the General Ledger hierarchical information is correct and to load the W_HIERARCHIES table.
- **Configuring the General Ledger Balance ID.** By default, the General Ledger Balance ID is maintained at the Set of Books and GL Code Combination ID level. If you want to maintain your General Ledger Balance at a different grain, you can redefine the GL Balance ID.
- **Configuring the AP Balance ID.** If you want to maintain your AP balance at a different grain, you can redefine the Balance ID value in the applicable mapplets.
- **Configuring the AR Balance ID.** If you want to maintain your AR balance at a different grain, you can redefine the Balance ID value in the applicable mapplets.
- **Configuring the AR Adjustments Extract.** If you want to extract additional types of AR adjustment entries, you can remove the filter in the Business Component mapplet. For more information on configuring the AR adjustments extract, see [Configuring the AR Adjustments Extract for Oracle Receivables Analytics on page 331](#).
- **Configuring the AR Schedules Extract.** If you want to extract additional types of AR schedule entries, you must remove the filter in the Business Component mapplet. For more information on configuring the AR schedules extract, see [Configuring the AR Schedules Extract on page 332](#).
- **Configuring the AR Cash Receipt Application Extract.** If you want to extract additional types of cash-receipt application entries, you can remove the filter in the Business Component mapplet. For more information on configuring the AR receipt application extract, see [Configuring the AR Cash Receipt Application Extract for Oracle Receivables Analytics on page 333](#).

- **Configuring the AR Credit-Memo Application Extract.** If you want to extract additional types of credit-memo application entries, you can remove the filter. For more information on configuring the AR receipt application extract, see [Configuring the AR Credit-Memo Application Extract for Oracle's Siebel Receivables Analytics on page 333](#).
- **Configuring the Customer Costs and Product Costs Fact Tables.** In Oracle General Ledger and Profitability Analytics, the Customer Costs and Product Costs 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. For more information on configuring the Customer Costs and Product Costs Fact Tables, see [Configuring the Customer Costs Lines and Product Costs Lines Tables for Oracle's Siebel Profitability Analytics on page 335](#).

Checklist for Configuring Oracle Financial Analytics for PeopleSoft 8.4

This section contains the configuration points for Oracle Financial Analytics that are specific to PeopleSoft 8.4.

The following section contains mandatory configuration points for Oracle Financial Analytics:

- **Configuring PeopleSoft Trees.** For PeopleSoft, the Financial Analytics application sources data from a data structure, called PeopleSoft Trees, to get information about the organization's General Ledger hierarchies, and so on. For a PeopleSoft environment with different tree names to the Financial Analytics application, you need to import these into the PowerCenter repository, and replace the old tree names with the new tree names. For more on PeopleSoft Trees, see [About PeopleSoft Trees in Oracle Financial Analytics on page 336](#).

Checklist for Configuring Supply Chain Analytics

This section contains the mandatory and optional configuration points for the Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) for Oracle 11i, universal source, and post-load processing.

Checklist for Configuring Supply Chain Analytics for Oracle 11i

This section contains the Supply Chain Analytics configuration points that are specific to Oracle11i.

The following section contains mandatory Supply Chain Analytics configuration points:

- **Deploying stored procedures.** Compile the Compute_Bounds_Ora11i.sql stored procedure in the target data warehouse. For more information on deploying stored procedures, see [Deploying Stored Procedures on page 286](#).
- **Configuring the Bill of Materials Explosion for Oracle 11.5.9 and above version.** This configuration is required to if your Oracle 11i application is version 11.5.9 and above. It's used by the Informatica Server to extract data from the BOM_EXPLOSION_TEMP table. See [Configuring the Bill of Materials Explosion Options on page 287](#).

The following section contains optional Supply Chain Analytics configuration points:

- Configuring the Bill of Materials (BOM) explosion option. This configuration allows you to choose an explosion option to load a BOM structure into the W_BOM_ITEMS table. For more information on configuring the BOM explosion option, see [Configuring the Bill of Materials Explosion Options on page 287](#).
- Configuring the left bound and right bound calculation. This configuration allows you to turn on or off the calculation of the left bound and right bound in the W_BOM_ITEMS table. For more information on configuring the left bound and right bound calculation, see [Configuring the Left Bound and Right Bound Calculation Option on page 292](#).
- Configuring the Quantity types for product transactions. 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. For more information on configuring the Quantity type for product transactions, see [Configuring Quantity Types for Product Transactions on page 294](#).
- Configuring the Country and State Region Name. This configuration allows you to load specific Region Names into the W_CODE_D table. For more information on configuring the Region Name, see [Configuring the Country Region and State Region Name on page 295](#).
- Configuring the State Name. This configuration allows you to load specific State Names into the W_CODE_D table. For more information on configuring the State Name, see [Configuring the State Name on page 296](#).
- Configuring the Country Name. This configuration allows you to load specific Country Names into the W_CODE_D table. For more information on configuring the Country Name, see [Configuring the Country Name on page 296](#).
- Configuring the Make-Buy Indicator. Your organization may require different indicator codes. If so, you can modify the indicator logic by reconfiguring the condition in the MPLT_SAI_PRODUCTS maplet. For more information on configuring the Make-Buy Indicator, see [Configuring the Make-Buy Indicator on page 297](#).
- Extracting particular purchase order records. By default, the filter condition is set to BLANKET or STANDARD. However, you can change this value to some conditional statement that only allows particular types of records to be extracted. For more information on extracting particular purchase order records, see [Extracting Particular Purchase Order Records on page 298](#).
- Configuring 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 PeopleSoft. See [About Configuring the Purchase Cycle Lines Aggregate Table on page 311](#).

Checklist for Configuring Supply Chain Analytics for Universal Source

This section contains configuration points for Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) that are specific to Universal Source.

The following section contains a mandatory Supply Chain Analytics configuration point:

- Configuring Expense Payment Types. The various expense types in the source data are mapped to Reimbursable Expenses (E), Expenses Prepaid (P), and Cash Advance (C).

The following section contains optional Supply Chain Analytics configuration points:

- Configuring the Preferred Merchant Flag. For more information on configuring the Preferred Merchant Flag, see [Configuring the Preferred Merchant Flag on page 306](#).
- Configuring the Customer Billable Indicator. For more information on configuring the Customer Billable Indicator, see [Configuring the Customer Billable Indicator on page 306](#).
- Configuring the Receipts Indicator. For more information on configuring the Receipts Indicator, see [Configuring the Receipts Indicator on page 307](#).
- Configuring the Default Expense Distribution Percentage. For more information on configuring the Default Expense Distribution Percentage, see [Configuring the Default Expense Distribution Percentage on page 308](#).
- Configuring Lookup Dates for Currency Conversion. The Supply Chain Analytics application uses the actual expiry date (ACTUAL_EXP_DATE) for looking up the exchange rate. You can configure the module to use a different date if required. For more information on configuring the Default Expense Distribution Percentage, see [Configuring Lookup Dates for Currency Conversion on page 309](#).

Checklist for Configuring Supply Chain Analytics for Post-Load Processing

This section contains the configuration points for Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) that are specific to post-load processing.

The following section contains optional Supply Chain Analytics configuration points:

- Configure the Inventory Monthly Balance table. This section allows you configure the Inventory Balance (W_INVENTORY_MONTHLY_BAL_F) table. For more information on configuring the Inventory Balance aggregate table, see [About Configuring the Inventory Monthly Balance Table on page 299](#).
- Configure the Product Transaction aggregate table. This section allows you configure the Product Transaction (W_PRODUCT_XACTS_A) aggregate table. For more information on configuring the Product Transaction aggregate table, see [About Configuring the Product Transaction Aggregate Table on page 301](#).
- Configuring the Purchase Receipts Aggregate Table. This section allows you configure the Purchase Receipts (W_PURCH_RCPT_F) aggregate table. For more information, see [About Configuring the Purchase Receipts Aggregate Table on page 309](#).
- Configuring the Purchase Cycle Lines Aggregate Table on page. This section allows you configure the Purchase Cycle Lines (W_PURCH_CYCLE_LINE_F) aggregate table. For more information, see [About Configuring the Purchase Cycle Lines Aggregate Table on page 311](#).

12

Configuring Oracle BI Applications Consumer

This chapter describes how to configure certain objects for the universal source to meet your business needs.

It contains the following topics:

- [Overview of Oracle Contact Center Telephony Analytics on page 263](#)
- [About the Dimension Key Resolution Process for Universal Source on page 264](#)
- [Loading the Dimension Table on page 264](#)
- [Loading the Fact Table on page 264](#)
- [About Configuring the Event Type Columns on page 265](#)
- [About Configuring Contact Status Type Column on page 268](#)
- [Configuring Dates and Times on page 275](#)
- [Setting Up the Representative Activities Table on page 269](#)
- [Setting Up the Contact Center Performance Table on page 271](#)
- [Setting Up the Benchmarks and Targets Table on page 272](#)
- [Configuring Flags for Oracle BI Applications Consumer on page 276](#)
- [Excluding Representative Data for Post-Load Processing on page 277](#)

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
- Integrated Voice Response History

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.

Loading 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.

Loading 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.

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.

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.

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 REP_ACTIVITY 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.

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.

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.

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 `ACW_ACTIVITY` 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.

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 BI Applications Consumer, 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 enquiry 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.

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.

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 date that is the same as the Activity end date and the Activity start local date the same as the Activity end local date. 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 46. 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 Microsoft WordPad or Notepad in the \$PMRoot\SrcFiles folder.
- 2 Type in your data to load the Representative Activities table.
- 3 Save and close the file.

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 Microsoft WordPad or Notepad in the `$PMRootDir\SrcFiles` folder.
- 2 Type in your data to load the Contact Center Performance table.
- 3 Save and close the file.

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 as shown in the following table:

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 [Customizing the Oracle Business Analytics Warehouse on page 171](#).
- 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 analytic 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 file.

To set up the Benchmarks and Targets table

- 1 Open the file_cntct_cntr_bnchmrk_tgt.csv file using Microsoft WordPad or Notepad in the \$PMRootDir\SrcFiles folder.
- 2 Type in your data to load the Benchmarks and Targets.
- 3 Save and close the file.

Logical Delete of Records From Fact Tables

If a fact record is deleted physically in the source OLTP system, you can logically delete that record from the data warehouse. To do this, a target staging table called <FACT table>_PE needs to be created for each fact table and a mapping needs to be created that can load this table.

For example in order to delete records from W_ACD_EVENT_F, a new staging table needs to be created as W_ACD_EVENT_F_PE where PE stands for primary extract. The structure of this table will be the 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 OLTP and in the data warehouse table W_ACD_EVENT_F, as follows:

Table 47. 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 48. 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 `_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 49. 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 50. 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

Configuring 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 local date columns for each fact table.

Table 51. 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.

Configuring Flags for Oracle BI Applications Consumer

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 52. 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.

Table 52. Configurable Flag Values and Descriptions

Flag	Flag Value	Description
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

Excluding Representative Data for Post-Load Processing

This section contains information about excluding representative data from contact representative and organization performance aggregate tables for post-load processing.

You may want to exclude certain contact representatives from the aggregation process for various reasons. For example, you may want to exclude representatives who are on vacation, or who are working on special projects.

The default configuration calculates contact-related information for all contact representatives in the enterprise. There are five aggregate tables supplied with the Oracle Contact Center Telephony Analytics application for improving the performance of the dashboards and reports:

- W_CNTCT_CNTRC_PERF_REP_HOUR_A
- W_CNTCT_CNTR_PERF_REP_DAY_A
- W_CNTCT_CNTR_PERF_REP_MONTH_A
- W_CNTCT_CNTR_PERF_ORG_HOUR_A
- W_CNTCT_CNTR_PERF_ORG_DAY_A

To exclude data about specific representatives from the aggregation calculation

- 1 In PowerCenter Designer, open the Configuration for Post Load Processing folder.
- 2 Open the PLP_ContactCenterPerformanceRepAggregate_Extract mapping.
- 3 Double-click the EXP_CNTCT_CNTR_PERF transformation.
- 4 Locate the EXCLUSION_IND_FLG port.
- 5 This port is preconfigured with a value of N indicating that all rows are included in the aggregates. Change this logic to include your logic to determine which groups of records you want to exclude.
- 6 Validate and save the mapping to the repository

NOTE: If you exclude data from an aggregate table, you also need to apply the same filter to the Logical Table Source corresponding to the W_CNTCT_CNTR_PERF_F base fact table in the repository metadata (Fact Service Contact Center Performance logical table). The metrics computed from the base fact tables are now consistent with those computed from the aggregate tables.

Configuring 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 the point in time 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 day and month level aggregates:

- W_REP_ACTIVITY_DAY_A
- W_REP_ACTIVITY_MONTH_A

To configure the alternate load plan to directly load aggregate table

- 1 Login to the DAC repository
- 2 Select the Universal container
- 3 Choose Design > Subject Area and query for the “Service – Rep Activity” subject area.
- 4 With this subject area selected, in the detail pane, click on Configuration Tags.
- 5 Uncheck the configuration point “ECC- Load into base fact and then aggregates”.
- 6 Check the configuration point “ECC – Load directly into aggregate facts”.
- 7 Redesign the subject area “Service – Rep Activity” and save.
- 8 Redesign the execution plan “Universal Contact Center Telephony Analytics”.

To provide hour level data

- 1 Open the file rep_activity_hour.csv using a text editor from \$PMRootDir\SrcFiles folder.
- 2 Enter the data for Representative activities at the hourly level.
- 3 Save and close the file.

Configuring Alternate Load Plan to Enable Direct Load to the Entry-Level Contact Performance Aggregates

The Contact Center Performance table stores information at a contact and representative grain. In case the data is not available at a contact grain, you can alternately choose to provide summarized data at the hourly level and load the Contact Center Performance hourly aggregate table, W_CNTCT_CNTR_PERF_REP_HOUR_A, directly.

This data will be used by the post-load process to load the following higher level aggregates.

W_CNTCT_CNTR_PERF_REP_DAY_A
W_CNTCT_CNTR_PERF_REP_MONTH_A
W_CNTCT_CNTR_PERF_ORG_HOUR_A
W_CNTCT_CNTR_PERF_ORG_DAY_A

To configure the alternate load plan to directly load aggregate

- 1 Login to the DAC repository
- 2 Select the Universal container
- 3 Go to Design -> Subject Area and query for the “Service - Contact Center Performance” subject area
- 4 With this subject area selected, in the detail pane, click on Configuration pointTags
- 5 Uncheck the configuration point “ECC- Load into base fact and then aggregates”
- 6 Check the configuration point “ECC – Load directly into aggregate facts”
- 7 Redesign the subject area “Service - Contact Center Performance and save.
- 8 Redesign the execution plan “Universal Contact Center Telephony Analytics”.

To provide hour level data

- 1 Open the file_rep_activity_hour.csv using wordpad or notepad from \$PMRootDir\SrcFiles folder.
- 2 Enter the data for Contact Center Performance at the hourly level.
- 3 Save and close the file.

NOTE: The Configuration Tag for the Direct Load to both Rep Activity and Contact Center Performance Hourly Aggregates is the same and hence either both can be loaded at hourly level or both can be loaded at point-in-time grain.

Formation of Integration_Id of the Enterprise Contact Center 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 adaptor 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 Adaptor

- 1** Login to the DAC repository and select the Universal container.
- 2** Go to the SDE_Universal task and click on the parameter tab.
- 3** Change the value of \$\$READ_INTEGRATION_ID_FROM_FILE parameter to FALSE.

The following table lists the expression of INTEGRATION_ID for each fact and the SDE task that creates it.

Table 53. Expressions and Task Names

Table Name	INTEGRATION_ID Expression	Task Name
W_ACD_EVENT_F	CALLID_NUM '~' EVENT_SEQUENCE_NUM	SDE_Universal_ACDEventFact
W_CNTCT_CNTR_PERF_F	CNTCT_NUM '~' CNTCT_REP_ID '~' CNTCT_START_DT	SDE_Universal_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

Setting Up the IVR Menu and IVR Navigation Profile Tables

The IVR Menu Dimension table W_IVR_MENU_D stores information about each menu point in the Interactive Voice Response System. The Integration_id in this table uniquely identifies a IVR MENU and is used as the dimension identifier IVR_MENU_ID in the ACD Event Fact fact table, W_ACD_EVENT_F.

If a call has n number of consecutive IVR events without any transfer to a contact representative, the NAV_PATH_ID for that call is identified as MENU1-MENU2-MENU3-.....-MENU_n where MENU_m is the IVR_MENU_ID of the mth event in that call. The NAV_PATH_ID is used to determine the navigation profile dimension key for a call.

It is recommended that the Integration_Id of the IVR MENU dimension W_IVR_MENU_D does not exceed 10 characters in order to keep the NAV_PATH_ID within the defined size limit of 255 characters.

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 54.

CallId	Event Type	Event Sequence Number	Menu Id
20060104-C1	IVR	1	MENU1
20060104-C1	IVR	2	MENU2
20060104-C1	IVR	3	MENU3
20060104-C1	REP	4	
20060104-C1	IVR	5	MENU4
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 55.

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.

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, results into a change in all the Navigation Profiles which have 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.

Viewing Duration Metrics in hh:mi:ss Format in Oracle BI Answers

The Oracle Business Intelligence Applications analytic repository (.rpd) has several duration metrics 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:mi:ss format. In order to view these duration metrics in hh:mi:ss format in an Oracle BI Answers report, do the following:

- 1 Select the duration in hh:mi:ss metric using the column selectors in the "Criteria" tab.
- 2 Click on "Format Column".
- 3 The Column Properties window will open. Go to Data format.
- 4 Check "Override Default data format".
- 5 Select Treat Number as "Custom" and give Custom Numeric Format as "#0#:##:##".

13

Configuring Oracle's Supply Chain Analytics family of products

This chapter describes how to configure certain objects for particular sources to meet your business needs.

It contains the following topics:

- [Overview of Oracle's Supply Chain Analytics family of products on page 283](#)
- [Process of Configuring Supply Chain Analytics for Oracle 11i on page 286](#)
- [Process of Aggregating Supply Chain Analytics Tables on page 299](#)
- [About Configuring Supply Chain Analytics for Universal Source on page 305](#)
- [Process of Configuring Supply Chain Analytics for a Universal Source on page 306](#)
- [Domain Values and CSV Worksheet Files for Supply Chain Analytics on page 313](#)

Overview of Oracle's Supply Chain Analytics family of products

Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) enable organizations to optimize their supply side performance by integrating data from across the enterprise value chain and enabling executives, managers, and frontline employees to make more informed and actionable decisions. Organizations using Supply Chain Analytics benefit from increased visibility into the complete Supply chain process, including comprehensive procurement and spend analysis, supplier performance analysis, inventory analysis, and supplier payables analysis. Through complete end-to-end insight into the factors that impact supply chain performance, organizations can significantly reduce costs, enhance profitability, increase customer satisfaction, and gain competitive advantage. Supply Chain Analytics also integrate with the other applications in Oracle BI Applications product line. They deliver this insight across the organization to increase the company's effectiveness in managing its customers, suppliers, and financial decisions. Supply Chain Analytics consists of the following modules:

- [Procurement and Spend Analytics Module on page 283](#)
- [Supplier Performance Analytics Module on page 284](#)
- [Inventory Analytics Module on page 285](#)

Procurement and Spend Analytics Module

Provides complete visibility into direct and indirect spend across the enterprise, payment, and employee expenses. Example analyses are spend by Commodity & Supplier, by Purchase Org, Cost Center, and expense by Employee, Buyer, etc.

The Procurement and Spend Analytics application is comprised of these subject areas:

- **Total Spend:** This is a summary subject area that provides the ability to do comparative analysis and report on requested spend, committed spend and actual spend across suppliers, company, products, commodities and associated hierarchies for both direct and indirect spend (indirect spend being MRO and employee expenses) in detail to allow complete visibility of spending across your organization.
- **Purchase Orders:** This is a detailed subject area that provides the ability to report on committed spend, and Purchase orders of the suppliers of an organization across suppliers, company, products, commodities and associated hierarchies at purchase order line level
- **Purchase Order Costs:** This is a detailed subject area that provides the ability to report on committed spend and Purchase orders of the suppliers of an organization across suppliers, company, products, and commodities and associated hierarchies at cost center (distribution line) level.
- **Purchase Cycle Lines:** This is a summary subject area that provides the ability to report cycle time performance such as Requisition to PO lead time, PO to Receipt lead time, P2P lead time of the Suppliers of an organization.
- **Purchase Schedules:** This is a detailed subject area that provides the ability to report on purchase order shipments of an organization across suppliers, company, products, commodities and associated hierarchies at purchase schedule line level
- **Purchase Requisitions:** This is a detailed subject area that provides the ability to report on requested spend and Purchase requisitions of the suppliers of an organization across suppliers, company, products, commodities and associated hierarchies at purchase requisition line level
- **Purchase Requisition Status:** This is a summary subject area that provides the ability to report on requisition status along the approval cycle of Purchase requisitions of the suppliers of an organization. It's populated only by Universal Adaptor.
- **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 Adaptor.

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.

Inventory Analytics Module

Incorporates analysis of inventory held by an organization; bill of materials; and inventory movements in, out, and through manufacturing plants, distribution centers, or storage locations. This enables companies to monitor inventory levels trend to Sales performance to improve cost exposure, increase turnover through inventory level reduction and increased velocity, properly deploy inventory at the right place / right time and better understand Customer and Supplier Returns to maintain quality.

The Inventory Analytics application is comprised of these subject areas:

- **Inventory Transactions.** The Inventory Transactions subject area allows you to analyze the various types of events and tasks that occur. Examples of these activities include tracking inventory by type of movement. For example, transfer, issues, receipts, returns, sales, and so on. It allows the user to understand the impact of these activities on business operations, and allows the identification of problematic trends early. For example, large quantities of product in-transit.
- **Inventory Balances.** The Inventory Balances subject area allows you to analyze the inventory held by an organization in relation to a number of different dimensions. For example, Product type, Product number, Storage location, Plant, Consigned Inventory, Restricted, and so on. It allows the user the ability to understand and determine the optimal distribution of assets as well as identify potential issues such as unnecessary build up of inventories.
- **Bill of Materials.** The Bill of Materials (BOM) subject area allows you to determine the profit margin of the components that comprise the finished goods. BOM allows you to keep up with the most viable vendors in terms of cost and profit, and to keep your sales organization aware of product delivery status, including shortages.
- **Customer and Supplier Returns.** The Customer and Supplier Returns subject area allows the user to specifically monitor the return of product by both Customers and Suppliers. At a Product level, it allows the user to identify early, potential, Customer-satisfaction issues relating to problematic Suppliers and Product.

Process of Configuring Supply Chain Analytics for Oracle 11i

This section contains configuration points for Supply Chain Analytics that are specific to Oracle 11i. It contains the following topics:

- [Deploying Stored Procedures on page 286](#)
- [Configuring the Bill of Materials Explosion for Oracle 11.5.9 and above version on page 287](#)
- [Configuring the Bill of Materials Explosion Options on page 287](#)
- [Configuring the Left Bound and Right Bound Calculation Option on page 292](#)
- [Configuring Quantity Types for Product Transactions on page 294](#)
- [Configuring the Country Region and State Region Name on page 295](#)
- [Configuring the State Name on page 296](#)
- [Configuring the Country Name on page 296](#)
- [Configuring the Make-Buy Indicator on page 297](#)
- [Extracting Particular Purchase Order Records on page 298](#)
- [Configuring the DAC parameter for Purchase Cycle Line on page 298](#)

Deploying Stored Procedures

This task is a step in the [Process of Configuring Supply Chain Analytics for Oracle 11i on page 286](#).

Stored procedures are a group of SQL statements that perform particular tasks on the database. For example, stored procedures can help to improve the performance of the database.

You can deploy stored procedures by copying the stored procedure files from your Oracle BI Analytics installation and deploying them to the target data warehouse.

NOTE: Some sessions may fail if these procedures are not compiled in the database before running the workflows.

To deploy stored procedures

- 1 Navigate to the OracleBI\dwrep\Informatica\Stored_Procedure_Scripts folder. copy the source codes Compute_Bounds_Ora11i.sql into the target data warehouse schema.
- 2 Compile the stored procedures in the target data warehouse database.

NOTE: If you have problems deploying the stored procedures, see your database reference guide, or contact your database administrator.

Configuring the Bill of Materials Explosion for Oracle 11.5.9 and above version

This task is a step in the [Process of Configuring Supply Chain Analytics for Oracle 11i on page 286](#).

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 11.5.9 and above

- 1 Click Start, click Run, type regedit, click OK.
- 2 Go to the following registry key:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PowerMart\Parameters\Configurati
ion
- 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 SDE_ORA1159_Adaptor or Oracle 11.5.9 Version and SDE_ORA11510_Adaptor for Oracle 11.5.10 version
- 5 Re-start the PowerCenter server (Informatica Service).

Configuring the Bill of Materials Explosion Options

This task is a step in the [Process of Configuring Supply Chain Analytics for Oracle 11i on page 286](#).

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 bomexpl.exploder_userexit stored procedure explodes the BOM structure. [Table 56](#) lists the variables for the bomexpl.exploder_userexit stored procedure.

Table 56. Variables for the bomexpl.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 Unimplemented
PLAN_FACTOR	2	1—Yes 2—No
EXPLODE_OPTION	2	1—All 2—Current 3—Current and Future
MODULE	2	1—Costing 2—BOM 3—Order Entry 4—ATO 5—WSM

Table 56. Variables for the bompexpl.exploder_userexit Stored Procedure

Input Variable	Preconfigured Value	Description
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 PowerCenter Designer, open the SDE_ORA115<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 box, and click the Port tab to display the EXPLODE_OPTION port.
- 4 Change the value to 1, and click Apply.
- 5 Validate the mapping, and save your changes to the repository.
- 6 Go to Mapplet Designer, and open mplt_BC_ORA_BOMHeaderDimension.
- 7 Double-click the SQL qualifier SQ_BOM_INVENTORY_COMPONENTS to open the Edit Transformations dialog box, and click on Properties tab, open value for SQL Query.
- 8 Modify the following default Where condition from:

```
((
/* CURRENT valid component changed */
INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY HH24:MI:SS')
AND (INV.EFFECTIVITY_DATE <= TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')
and (INV.DISABLE_DATE > TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS') OR
INV.DISABLE_DATE IS NULL))
```

```

OR

/* Component that became effective after last extract date and before today's
extract, for CURRENT Option*/

INV.EFFECTIVITY_DATE between TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS') and TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')

OR

/* Component that become disabled after last extract date and before today's
extract, for CURRENT and CURRENT-FUTURE Option*/

INV.DISABLE_DATE between TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS') and TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')

)

OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS'))

GROUP BY

```

To:

```

(INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY HH24:MI:SS')

OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS'))

GROUP BY

```

- 9 Click Apply, and Validate the mapping and save your changes to the repository.

To configure the BOM explosion to the Current and Future option

- 1 In PowerCenter Designer, open the SDE_ORA115<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 box, and click the Port tab to display the EXPLODE_OPTION port.
- 4 Change the value to 3, and click Apply.
- 5 Validate the mapping, and save your changes to the repository.
- 6 Go to Mapplet Designer, and open mplt_BC_ORA_BOMHeaderDimension.
- 7 Double-click the SQL qualifier SQ_BOM_INVENTORY_COMPONENTS to open the Edit Transformations dialog box, and click on Properties tab, open value for SQL Query.
- 8 Modify the following default Where condition from:

```

((

/* CURRENT valid component changed */

```

```

INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY HH24:MI:SS')
AND (INV.EFFECTIVITY_DATE <= TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')
and (INV.DISABLE_DATE > TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS') OR
INV.DISABLE_DATE IS NULL))

OR

/* Component that became effective after last extract date and before today's
extract, for CURRENT Option*/

INV.EFFECTIVITY_DATE between TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS') and TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')

OR

/* Component that became disabled after last extract date and before today's
extract, for CURRENT and CURRENT-FUTURE Option*/

INV.DISABLE_DATE between TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY HH24:MI:SS')
and TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')

)

OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS'))

GROUP BY

```

To:

```

((
INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY HH24:MI:SS')
AND ((INV.DISABLE_DATE > TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS') OR
INV.DISABLE_DATE IS NULL))

OR

INV.DISABLE_DATE between TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY HH24:MI:SS')
and TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')

)

OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS'))

GROUP BY

```

- 9 Click Apply, and Validate the mapping and save your changes to the repository.

To configure the BOM type

- 1 In PowerCenter Designer, open the SDE_ORA115<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 box, and click on Properties tab, open value for SQL Query.

- 4 Modify `BOM_ITEM_TYPE` section in Where statement.

Change the number to your BOM type. For example, change the number to 3 for a Planning BOM type.

```
Where INV.BOM_ITEM_TYPE = 3 AND
```

```
M.BOM_ITEM_TYPE = 3 AND
```

NOTE: You can also remove these two filters to extract all types of BOM.

- 5 Click Apply, and Validate the mapping and save your changes to the repository.

Configuring the Left Bound and Right Bound Calculation Option

This task is a step in the [Process of Configuring Supply Chain Analytics for Oracle 11i on page 286](#).

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.

Figure 26 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.

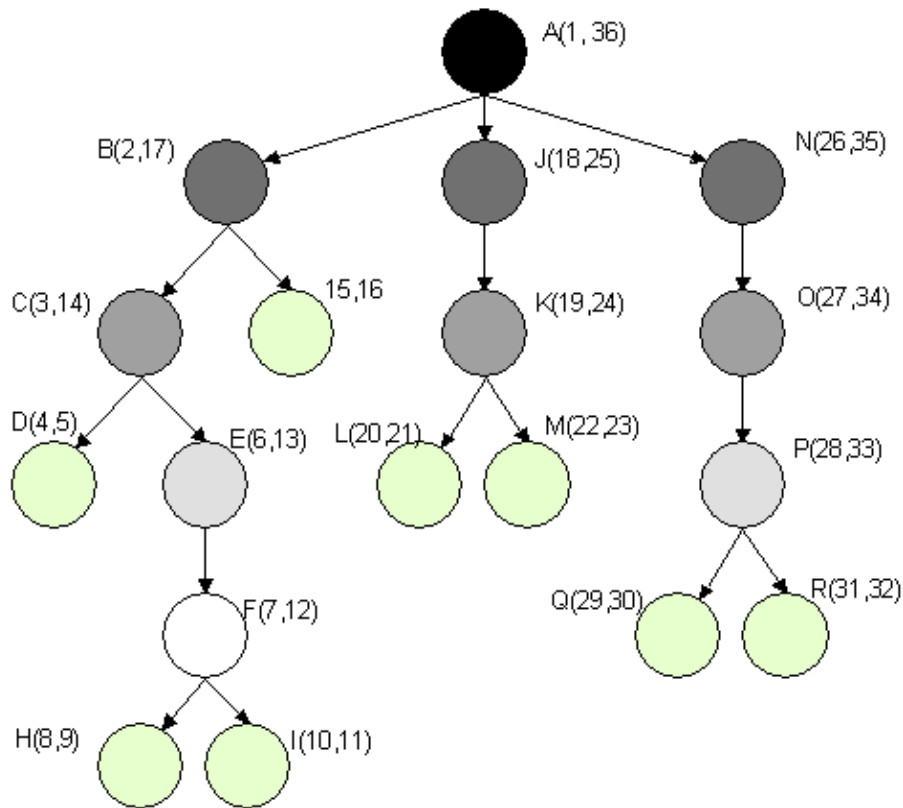


Figure 26. Sample BOM Structure

You can use the following procedure to turn on the left bound and the right bound calculation and populate the W_BOM_ITEM_F.LEFT_BOUNDS and W_BOM_ITEM_F.RIGHT_BOUNDS columns.

To configure the left bound and right bound calculation option

- 1 In PowerCenter Designer, open the SDE_ORA115<Ver>_Adaptor.
- 2 Open the SIL_BOMItemFact mapping.
- 3 Double-click the COMPUTE_BOUNDS stored procedure transformation to open the Edit Transformations dialog box, and click the Properties tab.
- 4 Change the value for Transformation Attributes Call Text to compute_bounds_ora11i(1).
- 5 Click Apply.
- 6 Validate the mapping, and save your changes to the repository.

Configuring Quantity Types for Product Transactions

This task is a step in the [Process of Configuring Supply Chain Analytics for Oracle 11i on page 286](#).

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.

All quantities extracted from the source system are always loaded into the Base quantity column (EXT_BASE_QTY). However, only the receipt quantity is loaded into the Goods Received quantity column (EXT_GR_QTY), and only delivered quantities are loaded into the Delivery quantity column (EXT_DELIVERY_QTY).

If your definition of goods received or delivery quantity is different from the prepackaged condition, then you can edit the condition to suit your business needs.

To configure the Quantity type

- 1 In PowerCenter Designer, open the SDE_ORA115<Ver>_Adaptor.
- 2 Open the mplt_SA_ORA_ProductTransactionFact mapplet.
- 3 Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT_GR_QTY and EXT_DELIVERY_QTY port.
- 4 Edit the quantity types by substituting your desired condition for the prepackaged expression.
- 5 Click Apply.
- 6 Validate the mapplet, and save your changes to the repository.

About Configuring the Handling of Currency Types

The Supply Chain Analytics applications, like all other applications, use the same method for handling currency conversions from document currency to local and group currencies. The only distinction is how currency is handled for product transactions. In a product transaction, if the document currency code is not supplied, the SLI uses the same code that it uses for the local currency code, and the local currency is derived using the Set of Books ID.

This guide provides a functional overview of how local and group currencies are derived depending on what data is supplied to the Oracle Business Analytics Warehouse. For more information on how to configure various components that relate to local, document, and group currencies, see [About Document, Local, and Global Currencies on page 227](#).

Configuring the Country Region and State Region Name

This task is a step in the [Process of Configuring Supply Chain Analytics for Oracle 11i on page 286](#).

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 [Codes Lookup on page 234](#).

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 PowerCenter Designer, open the Configuration for SDE_ORA1158_Adaptor 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_Plant
 - mplt_SA_ORA_BusinessLocationDimension_StorageLocation
- 3 Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT_COUNTRY_REGION port.
- 4 Edit the condition by removing the assigned value if you want the lookup to occur.
- 5 Click Apply.
- 6 Validate the mapplet, and save your changes to the repository.

To configure the State Region Name

- 1 In PowerCenter Designer, open the SDE_ORA115<Ver>_Adaptor.
- 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_Plant
 - mplt_SA_ORA_BusinessLocationDimension_StorageLocation
- 3 Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT_STATE_REGION port.

- 4 Edit the condition by removing the assigned value if you want the lookup to occur.
- 5 Click Apply.
- 6 Validate the mapplet, and save your changes to the repository.

Configuring the State Name

This task is a step in the [Process of Configuring Supply Chain Analytics for Oracle 11i on page 286](#).

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 [Codes Lookup on page 234](#).

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 PowerCenter Designer, open the SDE_ORA115<Ver>_Adaptor.
- 2 Open one of the following mapplets:
 - mplt_SA_ORA_SupplierDimension
 - mplt_SA_ORA_BusinessLocationDimension_Plant
 - mplt_SA_ORA_BusinessLocationDimension_StorageLocation
- 3 Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT_STATE_NAME port.
- 4 Edit the condition by removing the assigned value if you want the lookup to occur.
- 5 Click Apply.
- 6 Validate the mapplet and save your changes to the repository.

Configuring the Country Name

This task is a step in the [Process of Configuring Supply Chain Analytics for Oracle 11i on page 286](#).

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 [Codes Lookup on page 234](#).

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 PowerCenter Designer, open the SDE_ORA115<Ver>_Adaptor.
- 2** Open one of the following mapplets:
 - mplt_SA_ORA_SupplierDimension
 - mplt_SA_ORA_BusinessLocationDimension_Plant
 - mplt_SA_ORA_BusinessLocationDimension_StorageLocation
- 3** Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT_COUNTRY_NAME port.
- 4** Edit the condition by removing the assigned value if you want the lookup to occur.
- 5** Click Apply.
- 6** Validate the mapplet, and save your changes to the repository.

Configuring the Make-Buy Indicator

This task is a step in the [Process of Configuring Supply Chain Analytics for Oracle 11i on page 286](#).

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 PowerCenter Designer, open the SDE_ORA115<Ver>_Adaptor.

- 2 Open the mplt_SA_ORA_ProductDimension mapplet.
- 3 Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT_MAKE_BUY_IND port.
- 4 Edit the condition by replacing the prepackaged condition with your desired logic.
- 5 Click Apply.
- 6 Validate the mapplet, and save your changes to the repository.

Extracting Particular Purchase Order Records

This task is a step in the [Process of Configuring Supply Chain Analytics for Oracle 11i on page 286](#).

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 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.

Configuring the DAC parameter for Purchase Cycle Line

This task is a step in the [Process of Configuring Supply Chain Analytics for Oracle 11i on page 286](#).

To load the purchase cycle line table (W_PURCH_CYCLE_LINE_F), the ETL tasks needs to distinguish data coming from Oracle 11i applications from data from other applications such as SAP and PeopleSoft. This configuration explains the detail steps.

To configuring the DAC parameter for purchase cycle line

- 1** Login to your DAC repository
- 2** Go to the Tasks Tab, query for task SIL_PurchaseCycleLinesFact.
- 3** Click Parameters Child tab. Change the values for parameter \$\$ORA_DATASOURCE_NUM_ID_LIST from (2,4,5) to the list of Data Source NUM ID you defined for your Oracle connections
- 4** Repeat the same for task SIL_PurchaseCycleLinesFact_Extract
- 5** For more information on configuring DAC, please refer to *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

Process of Aggregating Supply Chain Analytics Tables

This section contains configuration points for Supply Chain Analytics for aggregating the Inventory Balance and Product Transaction tables.

It contains the following topics:

- [Configuring the Inventory Monthly Balance Table on page 300](#)
- [Configuring the Product Transaction Aggregate Table on page 303](#)

Related Topics

- [About Configuring the Inventory Monthly Balance Table on page 299](#)
- [About Configuring the Product Transaction Aggregate Table on page 301](#)

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 1. The value for the NUM_OF_PERIOD parameter is a positive integer, for example, 1, 2, 3, and so on.

Configuring the Inventory Monthly Balance Table

This task is a step in the [Process of Aggregating Supply Chain Analytics Tables on page 299](#).

You need to configure the parameterfileDW.txt parameters file, and run the initial ETL session or incremental ETL sessions to load the Inventory Monthly Balance table.

To configure the parameterfileDW.txt parameters file

- 1 Open the parameterfileDW.txt file using Microsoft WordPad or Notepad in the OracleBI\DAC\Informatica\parameters\input folder.
- 2 Replace the default parameter values with your new values.

For a list of values for each parameter see the [About Configuring the Inventory Monthly Balance Table on page 299](#).

The default values for the parameterfileDW.txt file are shown in the following table.

Table 57. Default values in the parameterfileDW.txt file are s

SESSION	Parameter	Value
PLP_InventoryMonthlyBalance	\$\$GRAIN	Month
PLP_InventoryDailyBalance_Trim	\$\$KEEP_PERIOD	'MONTH'
PLP_InventoryDailyBalance_Trim	\$\$NUM_OF_PERIOD	3

NOTE: You need to use single quotes for the values of the KEEP_PERIOD and GRAIN parameters.

- 3 Save and close the file.

To incrementally refresh the Inventory Monthly Balance table

- 1 Delete the records from the Monthly Balance (W_INVENTORY_MONTHLY_BAL_F) aggregate table for a certain time.

The GRAIN parameter determines the time period for the deletion. For example, if GRAIN=MONTH, and the date is May 15, 2005, then all records for April and the current month (May) are deleted in the Monthly Balance (W_INVENTORY_MONTHLY_BAL_F) table.

Running the PLP_InventoryMonthlyBalance workflow mapping implements this step.

- 2 Retrieve the records in the Inventory Balance (W_INVENTORY_DAILY_BALANCE_F) fact table and load the records to the Monthly Balance (W_INVENTORY_MONTHLY_BAL_F) table at a certain grain level.

For example, if GRAIN=MONTH, then the month end balance records in the W_INVENTORY_DAILY_BALANCE_F fact table are stored in and aggregated to the Monthly Balance (W_INVENTORY_MONTHLY_BAL_F).

Running the S_M_PLP_INV_BALANCE_A1_AGG session, and the M_PLP_INV_BALANCE_A1_AGG mapping implements this step. For the current month balance, balance records of the previous day (if it is in the same month) are deleted from W_INVENTORY_MONTHLY_BAL_F, and balance records of the current day will be loaded from W_INVENTORY_BALANCE_F to W_INVENTORY_MONTHLY_BAL_F.

Running the PLP_InventoryMonthlyBalance workflow implements this step.

- 3 Remove the old records from the W_INVENTORY_DAILY_BALANCE_F fact table.

To remove old records you need to use the KEEP_PERIOD and the NUM_OF_PERIOD parameters. For example, if KEEP_PERIOD=MONTH, NUM_OF_PERIOD=1, and the date is May 15, 2005, then the records for April and the current month (May) are kept and the older records are deleted.

Running the PLP_InventoryDailyBalance_Trim workflow implements this step.

NOTE: The trimming process is to reduce data size in the table. It is important to emphasize that you will not be able to see the old daily balance records. But you will still be able to see the month-end balance. Therefore, please make sure that you adjust the NUM_OF_PERIOD values to reflect your data volume and data recency requirement.

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.

You need to configure three parameters to aggregate the Product Transaction table for you initial run:

- GRAIN
- KEEP_PERIOD

■ NUM_OF_PERIOD

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.

You need to configure five parameters to aggregate the Product Transaction aggregate table for you initial run:

- GRAIN
- REFRESH_PERIOD
- NUM_OF_PERIOD (PLP_ProductTransactionAggregate)
- KEEP_PERIOD
- NUM_OF_PERIOD (PLP_ProductTransaction_Trim)

The GRAIN parameter has a preconfigured value of Month. The possible values for the GRAIN parameter are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The REFRESH_PERIOD parameter has a preconfigured value of Month. Values for the REFRESH_PERIOD parameter include:

- 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 for the PLP_ProductTransactionAggregate session 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.

The NUM_OF_PERIOD parameter for the PLP_ProductTransaction_Trim session 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.

Configuring the Product Transaction Aggregate Table

This task is a step in the [Process of Aggregating Supply Chain Analytics Tables on page 299](#).

You need to configure the parameterfileDW.txt parameters file, and run the initial ETL and then the incremental ETL to load the Product Transaction aggregate table.

To configure the parameterfileDW.txt parameters file

- 1 Open the parameterfileDW.txt file using Microsoft WordPad or Notepad in the OracleBI\DAC\Informatica\parameters\input folder.
- 2 Replace the default parameter values with your new values.

For a list of values for each parameter see the [About Configuring the Product Transaction Aggregate Table on page 301](#).

The default values are shown in the following table.

Table 58. Default values in the parameterfileDW.txt file

SESSION	PARAMETER	VALUE
PLP_ProductTransaction_Trim	\$\$KEEP_PERIOD	MONTH
PLP_ProductTransaction_Trim	\$\$NUM_OF_PERIOD	3
PLP_ProductTransactionAggregate	\$\$REFRESH_PERIOD	MONTH
PLP_ProductTransactionAggregate	\$\$GRAIN	MONTH
PLP_ProductTransactionAggregate	\$\$NUM_OF_PERIOD	3
PLP_ProductTransactionAggregate_Full	\$\$GRAIN	MONTH
PLP_ProductTransactionAggregate_Full	\$\$NUM_OF_PERIOD	3
PLP_ProductTransactionAggregate_Full	\$\$REFRESH_PERIOD	

NOTE: You need to use single quotes for the values of the KEEP_PERIOD, GRAIN, and REFRESH_PERIOD parameters. The KEEP_PERIOD value must be equal to or greater than the GRAIN value. The REFRESH_PERIOD value must equal the GRAIN value.

- 3 Save and close the file.

To configure the Product Transaction aggregate table for the initial ETL run

- 1 Retrieve the records in the Product Transaction fact (W_PRODUCT_XACT_F) table, and aggregate the records to the Product Transaction aggregate (W_PRODUCT_XACT_A) table at a certain grain level.

For example, if GRAIN=MONTH then the records in the W_PRODUCT_XACT_F fact table are retrieved and aggregated to the W_PRODUCT_XACT_A table at a monthly level.

Running the SPLP_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.

- 3 Remove the old records from the W_PRODUCT_XACT_FW_PRODUCT_XACT_A fact table.

To remove old records you need to use the KEEP_PERIOD and the NUM_OF_PERIOD parameters. For example, if KEEP_PERIOD=YEAR, NUM_OF_PERIOD=3, and the date is May 1, 2005, then the records for the years 2002, 2003, and 2004, and the current year (2005), are kept and the older records are deleted.

Running the PLP_ProductTransaction_Trim workflow implements this step.

NOTE: This workflow is not invoked at the initial ETL run, because data needs to be loaded to W_PRODUCT_XACT_F and then aggregated to W_PRODUCT_XACT_A. During the incremental ETL run, the old transactions are no longer important for the aggregations and can be trimmed to reduce data size in this table. It's important to emphasize that you will not be able to see the old transaction records. But you will still be able to see the aggregated summary records in the aggregate table W_PRODUCT_XACT_A. So please make sure you adjust the NUM_OF_PERIOD values to reflect your data volume and data recency requirement.

About Configuring Supply Chain Analytics for Universal Source

Expenses has one fact table (W_EXPENSE_F) that supports metrics and reports for examining employee expenses. Several mappings populate these tables to complete extracts, loads and updates; you may configure these to suit your organization's business rules. The following sections discuss decisions you must make before you begin adapting individual PowerCenter objects, and provide specific configuration procedures for the universal source.

Universal Source Adapter mapplets extract data from a flat file interface to populate the Oracle Business Analytics Warehouse. In this phase of your project, you can configure the following:

- **System Flags and Indicators.** You may configure various system flags to indicate record rejection settings, as well as to indicate if your employees are using your preferred vendors, if you can forward expenses to your customers, and if receipts are available for expensed items.
- **Currency and Payment Options.** You may configure the date used to establish your exchange rates, determine if you allow expenses to be distributed across multiple cost centers, and define payment types in your data warehouse.

Before you begin, you must make the following decisions:

- **Cash Advances.** Cash advance records have a unique expense item number. If your system allows multiple cash advance records for one expense report, each of these advances must have their own identifiers.
- **Violations.** Many organizations capture violations of company expense policies at the item level (for example, the line item airfare exceeds \$2000), cash advance level (for example, cash advance exceeds \$500) and at the expense report level (for example, the report's total expenses exceed \$5000). Currently the Oracle Business Analytics Warehouse stores item level violations within the corresponding item record, but the cash advance record stores both cash advance and report-level violations. Furthermore, each record has a VIOLATION_WID that can point to W_REASON_D, where violation details are stored. Depending on how you want your analytic system to perform, you must edit your universal business adapter file to reflect the violation counts and keys appropriately. For example:
 - If a requestor violates a cash advance policy, but there are no other violations at the report level, the VIOLATION_ID refers to the cash advance violation only. The violation count equals the cash advance violation counts.
 - If a requestor violates company policy with their expense report, but has not taken a cash advance, you must add a dummy record in the flat file for a cash advance and set the cash advance amount to zero, and enter the violation count as the total number of expense report violations. In this scenario, VIOLATION_ID refers to the expense report violation data only.
 - If a requestor violates a cash advance policy and an expense report policy, you must total the violation counts and enter them in your flat file record, and the VIOLATION_ID has no value. However, if your organization wants to prioritize the violations and have the VIOLATION_ID point to that which is most important, you may point it to the appropriate entry in W_REASON_D.

Process of Configuring Supply Chain Analytics for a Universal Source

Universal business adapters are built to source data from a flat file interface and populate the data warehouse base tables.

To configure Supply Chain Analytics for a universal source, perform the following tasks:

- [Configuring the Preferred Merchant Flag on page 306](#)
- [Configuring the Customer Billable Indicator on page 306](#)
- [Configuring the Receipts Indicator on page 307](#)
- [Configuring Expense Payment Types on page 307](#)
- [Configuring the Default Expense Distribution Percentage on page 308](#)
- [Configuring Lookup Dates for Currency Conversion on page 309](#)

Configuring the Preferred Merchant Flag

This task is a step in the [Process of Configuring Supply Chain Analytics for a Universal Source on page 306](#).

The Oracle Business Analytics Warehouse provides a preferred merchant flag to indicate whether the requestor used a preferred merchant for an expensed item. The flag can have only one value—Y (item acquired from a preferred merchant) or N (item acquired from a merchant not recorded). If you use custom logic to determine merchant status, you must include that logic in the expenses Source Adapter.

To configure the preferred merchant flag

- 1 In PowerCenter Designer, open the Configuration for Universal Source folder.
- 2 Open the SDE_Universal_ExpenseFact mapping.
- 3 Select the Expression transformation to open the Edit Transformations box and select the Port tab.
- 4 Select the Add Port icon to add the PREF_MERCHANT_FLAG port.
- 5 Enter preferred merchant flag logic.
- 6 Validate the mapplet and click OK to exit.
- 7 Save your changes to the repository.

Configuring the Customer Billable Indicator

This task is a step in the [Process of Configuring Supply Chain Analytics for a Universal Source on page 306](#).

The Oracle Business Analytics Warehouse provides a customer billable indicator that registers whether an expense item is billed to a customer or paid by your organization. The flag can have only one value—Y (cost is passed to the customer) or N (cost is paid by your organization). If you use custom logic to determine customer billable status, you must include that logic in the expenses Source Adapter.

To configure the customer billable indicator

- 1 In PowerCenter Designer, open the Configuration for Universal Source folder.
- 2 Open the SDE_Universal_ExpenseFact mapping.
- 3 Select the Expression transformation to open the Edit Transformations box, and select the Port tab.

Add a port called CUST_BILLABLE_FLG = <insert your expression here>.

- 4 Validate the mapplet and click OK to exit.
- 5 Save your changes to the repository.

Configuring the Receipts Indicator

This task is a step in the [Process of Configuring Supply Chain Analytics for a Universal Source on page 306](#).

The Oracle Business Analytics Warehouse provides a receipts indicator that registers whether requestors have submitted a receipt for a line item in their expense report. The flag can have only one value—Y (receipts are available) or N (receipts are not available). If you use custom logic to indicate receipt availability, you must include that logic in the expenses Source Adapter.

To configure the receipts indicator

- 1 In PowerCenter Designer, open the SDE_Universal_ExpenseFact mapping.
- 2 Open the MPLT_SAF_EXPENSES Source Adapter mapplet.
- 3 Select the Expression transformation to open the Edit Transformations box, and select the Port tab.

Add a port called RECEIPT_FLG = <insert your expression here>.

- 4 Validate the mapplet and click OK to exit.
- 5 Save your changes to the repository.

Configuring Expense Payment Types

This task is a step in the [Process of Configuring Supply Chain Analytics for a Universal Source on page 306](#).

The Oracle Business Analytics Warehouse supports analysis on three types of payment—Reimbursable Expense (type E), expenses prepaid by your company (type P), and cash advance (type C). All of your organization's payment types must be mapped to one of these types described earlier; do this by modifying MPLT_SAF_EXPENSES.

To configure additional payment types

- 1** In PowerCenter Designer, open the SDE_Universal_ExpenseFact mapping.
- 2** Open the MPLT_SAF_EXPENSES Source Adapter mapplet.
- 3** Select the Expression transformation to open the Edit Transformations box, and select the Port tab to display the EXPENSE_PAY_TYPE_ID port.
- 4** Copy and paste this port, rename it EXPENSE_PAY_TYPE_ID.
- 5** Select O to make your new port an output port.
- 6** Add a decode logic in the expression to decode source-supplied values to the Oracle Business Analytics Warehouse payment type of your choice.
- 7** Validate your mapping, and save your changes to the repository.

Configuring the Default Expense Distribution Percentage

This task is a step in the [Process of Configuring Supply Chain Analytics for a Universal Source on page 306](#).

At times, employee expenses may be distributed across multiple cost centers. For example, technical support associates frequently travel to work in an office with many cost centers; their expenses could be split between those who used their services. This cost center distribution is expected as a percentage from the source system or file; if it is not present a null value is returned. However, this prevents further calculations, so it is preferable to configure the default to be 100% if only one cost center is charged, rather than allow the system to return a null value.

To configure the default expense distribution percentage

- 1** In PowerCenter Designer, open the Configuration for Universal Source folder.
- 2** Open the SDE_Universal_ExpenseFact mapping.
- 3** Select the Expression transformation to open the Edit Transformations box and select the Port tab.
- 4** Add a port named DIST_PERCENTAGE = <expression that sets this to 100%>.
- 5** Validate your mapplet.
- 6** Save your changes.

Configuring Lookup Dates for Currency Conversion

The Oracle Business Analytics Warehouse supports conversion of currency to document (transactional, or source, currency) and group (corporate umbrella currency) for exchange rates. The Oracle Business Analytics Warehouse uses a specific lookup date to determine the rate of exchange on the date an expense was incurred (ACTUAL_EXP_DT). If you decide to use a different date as your currency conversion lookup, you must use the following procedure.

To configure the exchange rate lookup date

- 1 In PowerCenter Designer, open the Configuration for Universal Source folder.
- 2 Open the SDE_Universal_ExpenseFact mapping.
- 3 Select the Expression transformation to open the Edit Transformations box, and select the Port tab to display the EXT_XRATE_LKP_DATE port.

The expression contains the exchange rate lookup date logic.

- 4 Select the expression in the EXT_XRATE_LOOKUP_DATE port to open the Expression Editor box and edit the expression.
- 5 Edit the lookup date logic by substituting your lookup date for the prepackaged expression.
- 6 Validate the mapplet and click OK to exit.
- 7 Save your changes.

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.

Configuring the Purchase Receipts Aggregate Table

To load the Purchase Receipts aggregate table (W_PURCH_RCPT_A), you need to configure the post-load-processing parameter file and the source system parameter files, and run the initial workflow and then the incremental workflow.

To configure the parameterfiledw.txt parameters file

- 1** Open the parameterfiledw.txt file using Microsoft WordPad or Notepad in the OracleBI\DAC\Informatica\parameters\input folder.
- 2** Replace the default parameter values with your new values.
- 3** Save and close the file.

To configure the Purchase Receipts aggregate table for Universal Source

- 1 Open the parameterfileDW.txt file using Microsoft WordPad or Notepad in the OracleBI\DAC\Informatica\parameters\input folder.
- 2 Replace the default parameter values with your new values.
- 3 Save and close the file.

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.

Configuring the Purchase Cycle Lines Aggregate Table

To load the Purchase Cycle Lines aggregate table (W_PURCH_CYCLE_LINE_A), you need to configure the post-load-processing parameter file and the source system parameter files, and run the initial workflow and then the incremental workflow.

To configure the file_parameters_plp.csv parameters file

- 1 Open the parameterfiledw.txt file using Microsoft WordPad or Notepad in the OracleBI\DAC\Informatica\parameters\input folder.

- 2 Replace the default parameters values with your new values.
- 3 Save and close the file.

Domain Values and CSV Worksheet Files for Supply Chain Analytics

Table 59 lists the CSV worksheet files and the domain values for Supply Chain Analytics in the \$pmserver\LkpFiles folder.

For more information on configuring domain values with CSV worksheet files, see [About Domain Values on page 238](#) and [Configuring the Domain Value Set with CSV Worksheet Files on page 243](#).

Table 59. Domain Values and CSV Worksheet Files for Supply Chain Analytics

Worksheet File Name	Description	Session
domainValues_Status_Purch_Approve_ora11i.csv	Lists the Purchasing Approval Status column and the corresponding domain values for the Oracle 11i application.	S_M_I_STATUS_PURCH_APPROVE_LOAD
domainValues_Status_Purch_Cycle_ora11i.csv	Lists the Purchasing Cycle Status column and the corresponding domain values for the Oracle 11i application.	S_M_I_STATUS_PURCH_CYCLE_LOAD
domainValues_Xact_Types_Purch_Orders_ora11i.csv	Lists the Purchase Order Transaction Type column and the corresponding domain values for the Oracle 11i application.	S_M_I_XACT_TYPES_PURCH_ORDERS_LOAD
domainValues_Xact_Types_Purch_Requisitions_ora11i.csv	Lists the Purchase Requisition Transaction Type column and the corresponding domain values for the Oracle 11i application.	S_M_I_XACT_TYPES_PREQ_LOAD
domainValues_Xact_Types_PO_Line_Type_ora11i.csv	Lists the Purchasing Line Type and the corresponding domain Values for the Oracle 11i application	SDE_ORA_TransactionTypeDimension_PO_Line_Type

14 Configuring Oracle Financial Analytics

This chapter describes how to configure certain objects for particular sources to meet your business needs.

This chapter contains the following topics:

- [Overview of Oracle Financial Analytics on page 315](#)
- [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#)
- [About PeopleSoft Trees in Oracle Financial Analytics on page 336](#)

Overview of Oracle Financial Analytics

Oracle Financial Analytics consists of Oracle' Siebel General Ledger Analytics, Oracle Payables Analytics, Oracle Receivables Analytics, and Oracle General Ledger and Profitability Analytics, as follows:

- **Oracle General Ledger and Profitability Analytics.** The Siebel General Ledger Analytics application provides information to support your enterprise's balance sheet and provides a clearer understanding of the chart of accounts.

The default configuration for the Siebel 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.

- **Oracle Payables Analytics.** The Siebel 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 Siebel 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 Siebel 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 Siebel 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 Siebel 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 Siebel 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.

- **Oracle General Ledger and Profitability Analytics.** The Siebel 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 Siebel Profitability Analytics application pertains to data found in the revenue and expense account groupings of your financial statements and chart of accounts. The Siebel 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 Siebel 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 Siebel 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.

Process of Configuring Oracle Financial Analytics for Oracle 11i

To configure the Siebel Financial Analytics application for Oracle 11i, perform the following tasks:

- [About the Name of the Oracle 11i Folder in Informatica on page 317](#)
- [Extracting Data Posted at the Detail-Level for Oracle 11i on page 317](#)
- [Mapping Oracle General Ledger and Profitability Analytics Account Numbers to Group Account Numbers on page 319](#)
- [Configuring Oracle General Ledger and Profitability Analytics Transaction Extracts on page 321](#)
- [Configuring General Ledger Cost Of Goods Extract on page 322](#)
- [Configuring the General Ledger Account Hierarchies on page 322](#)
- [Configuring the General Ledger Balance ID on page 329](#)
- [Configuring AP Balance ID for Oracle Payables Analytics on page 330](#)
- [Configuring AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics on page 331](#)

- [Configuring the AR Adjustments Extract for Oracle Receivables Analytics on page 331](#)
- [Configuring the AR Schedules Extract on page 332](#)
- [Configuring the AR Cash Receipt Application Extract for Oracle Receivables Analytics on page 333](#)
- [Configuring the AR Credit-Memo Application Extract for Oracle's Siebel Receivables Analytics on page 333](#)
- [Configuring the Customer Costs Lines and Product Costs Lines Tables for Oracle's Siebel Profitability Analytics on page 335](#)

NOTE: Mandatory configurations are those that configure Group Account Number and the Fiscal Calendars.

Related Topic

- [About the Customer Costs Lines and Product Costs Lines Tables for Oracle's Siebel Profitability Analytics on page 334](#)

About the Name of the Oracle 11i Folder in Informatica

In Informatica, the Oracle11i adaptors are stored in the SDE_ORA11<ver>_Adaptor folder, where <ver> is the Applications version number. Examples folder names are SDE_ORA1158_Adaptor, SDE_ORA1159_Adaptor, and SDE_ORA11510_Adaptor.

Extracting Data Posted at the Detail-Level for Oracle 11i

This task is a step in the [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#).

By default, the Oracle Business Analytics Warehouse assumes that the posting from your journal to your Oracle General Ledger is done at the summary level, and that references are maintained in Oracle General Ledger for AP and AR subledgers. If import references are not maintained in Oracle General Ledger and the posting from AP and AR is at the detail-level, then modify the filter condition in SDE_ORA_GLJournals and disable the SDE_ORA_GLJournals_ImportReference_Extract session so that only the session S_SDE_ORA_GLJournals loads into the common table W_ORA_GLRF_DERV_F_TMP.

To modify the filter condition for posting at the detail level

- 1 In the PowerCenter Designer, open the appropriate Oracle Applications 11i folder (for example, SDE_ORA11510_Adaptor). For more information, see [About the Name of the Oracle 11i Folder in Informatica on page 317](#).
- 2 Open the SDE_ORA_GLJournals mapping.
- 3 Select the FIL_GL_XACTS_JOURNAL filter, and click the Properties tab to edit the filter condition. To load postings at the detail level, replace the 1=2 condition with 1=1.
- 4 Validate and save your changes to the repository.

To disable the session for Siebel General Ledger Analytics extract

- 1** In PowerCenter Workflow Manager, open the appropriate Oracle Applications 11i folder (for example, SDE_ORA11510_Adaptor). For more information, see [About the Name of the Oracle 11i Folder in Informatica on page 317](#).
- 2** Open the SDE_ORA_GLJournals_ImportReference_Extract workflow.
- 3** Double-click the SDE_ORA_GLJournals_ImportReference_Extract session to open the Edit Tasks window.
- 4** Click Disable this task, and click OK.

NOTE: As a best practice, you must move unused sessions to another folder to avoid error messages and preserve it for future use.

Configuring How GL Balances Is Populated

Oracle BI Applications provides two ways to populate the GL balances (stored in the W_GL_BALANCE_F table), as follows:

- 1** By extracting the GL balances directly from Oracle General Ledger, as follows:
 - a** In DAC, for the Subject Area 'Financials – General Ledger', in the 'Configuration Tag' tab, make sure that the tag 'Oracle – Extract GL Balance' is unchecked for the 'Inactive' checkbox.
 - b** Make sure the tag 'Financials – Calculate GL Balance' is checked in the 'Inactive' checkbox.
 - c** Click Assemble to redesign the subject area.
 - d** After the subject area has been redesigned, redesign the execution plans that contain this subject area.
- 2** By calculating the GL balances based on the records in the W_GL_OTHER_F table, which stores all journal lines, as follows:
 - a** In DAC, for the Subject Area 'Financials – General Ledger', in the 'Configuration Tag' tab, make sure that the tag 'Financials – Calculate GL Balance' is unchecked for the 'Inactive' checkbox.
 - b** Make sure the tag 'Oracle – Extract GL Balance' is checked in the 'Inactive' checkbox.
 - c** Click Assemble to redesign the subject area.
 - d** After the subject area has been redesigned, redesign the execution plans that contain this subject area.

Configuring 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 lengths:

- 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.

Mapping Oracle General Ledger and Profitability Analytics Account Numbers to Group Account Numbers

This task is a step in the [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#).

NOTE: It is critical that the General Ledger Account Numbers are mapped to the Group Account Numbers (or domain values) as the metrics in the General Ledger reporting layer uses these values. For a list of domain values for General Ledger Account Numbers, see *Oracle Business Analytics Warehouse Data Model Reference*.

You can categorize your Oracle General Ledger accounts into specific group account numbers. You may use this information during data extraction as well as front-end reporting. The GROUP_ACCT_NUM field denotes the nature of the Siebel General Ledger Analytics accounts. For example, Cash account, Payroll account, and so on. Refer to the master_code column in the file_group_acct_names_ora11i.csv file for values you can use. For a list of the Group Account Number domain values, see *Oracle Business Analytics Warehouse Data Model Reference*. The mappings to General Ledger Accounts Numbers are important for both Profitability Analysis (Income Statement) and General Ledger accounts.

The logic for assigning the accounts is located in the `file_group_acct_codes_ora11i.csv` file. Table 60 shows the layout of the `file_group_acct_codes_ora11i.csv` file.

Table 60. Layout of `file_group_acct_codes_ora11i.csv` File

SOB 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 Table 60, in the first row, all accounts within the account number range from 101010 to 101099 containing a Set of Books (SOB) ID equal to 1 are assigned to Current Asset. Each row maps all accounts within the specified account number range and with the given Set of Books ID.

If you need to create a new group of account numbers, you can create new rows in the `file_group_acct_names_ora11i.csv` file. You can then assign GL accounts to the new group of account numbers in the `file_group_acct_codes_ora11i.csv` file.

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_ora11i.csv` file.

To map Oracle General Ledger and Profitability Analytics Accounts to Group Account Numbers

- 1 Open the `file_group_acct_codes_ora11i.csv` file with Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

- 2 Edit the fields in the following table:

Field Name	Description
SOB ID	The set of books ID for the Siebel General Ledger Analytics accounts.
FROM ACCT and TO ACCT	The From Account and To Account specify the range of Oracle General Ledger and Profitability Analytics accounts for the mapping. The value you specify comes from the value of the natural account segment of the Siebel General Ledger Analytics account.
GROUP_ACCT_NUM	This field denotes the nature of the Siebel General Ledger Analytics accounts. For example, Cash account, Payroll account, and so on. Refer to the file_group_acct_names_ora11i.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.

Configuring Oracle General Ledger and Profitability Analytics Transaction Extracts

This task is a step in the [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#).

There are two separate transaction extracts for Oracle General Ledger and Profitability Analytics—General Ledger Revenue and General Ledger COGS. By default, the Siebel General Ledger Analytics application extracts only Completed revenue and COGS that have been posted to the general ledger. Completed revenue transactions are those where the RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG = Y. If you want to extract incomplete revenue transactions, you can remove the filter in the Business Component.

To modify the extract filter for Oracle General Ledger and Profitability Analytics Revenue

- 1 In PowerCenter Designer, open the appropriate Oracle Applications 11i folder (for example, SDE_ORA11510_Adaptor). For more information, see [About the Name of the Oracle 11i Folder in Informatica on page 317](#).
- 2 In Mapplet Designer, open the mp1t_BC_ORA_GLRevenueFact mapplet.
- 3 Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

In the User Defined Join field and in the SQL Query field, remove the statement:

```
AND RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG = Y
```

- 4 Validate and save your changes to the repository.

Configuring General Ledger Cost Of Goods Extract

This task is a step in the [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#).

By default, the Siebel General Ledger Analytics application extracts only COGS transactions that have been posted to the general ledger. 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.

To modify the extract filter for General Ledger COGS

- 1** In PowerCenter Designer, open the appropriate Oracle Applications 11i folder (for example, `SDE_ORA11510_Adaptor`). For more information, see [About the Name of the Oracle 11i Folder in Informatica on page 317](#).
- 2** In Mapplet Designer, open `mp1t_BC_ORA_GLCOGSFact`.
- 3** Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

In the User Defined Join field and in the SQL Query field, remove the statement:

`AND MTL_TRANSACTION_ACCOUNTS.GL_BATCH_ID <> -1`
- 4** In Mapplet Designer, open `mp1t_BC_ORA_GLCOGSFact_PRIMARY`.
- 5** Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

In the User Defined Join field and in the SQL Query field, remove the statement:

`AND MTL_TRANSACTION_ACCOUNTS.GL_BATCH_ID <> -1`
- 6** Validate and save your changes to the repository.

Configuring the General Ledger Account Hierarchies

There are two ways to set up hierarchies in Oracle Financial Analytics:

- Using General Ledger Accounting Flexfield Value Sets Definitions (for more information, see [Configuring General Ledger Account Hierarchies Using General Ledger Accounting Flexfield value sets definitions on page 323](#)).
- Using the Financial Statement Generator (FSG) Report Definition ([Configuring General Ledger Account Hierarchies Using Financial Statement Generator \(FSG\) Report Definition on page 327](#)).

Whichever method you choose to set up General Ledger Account hierarchies, you store the hierarchy information in the `W_HIERARCHY_D` table. To illustrate how hierarchy information is stored in the `W_HIERARCHY_D` table, see the following example using US Acct.

Figure 27. Example hierarchy US Acct

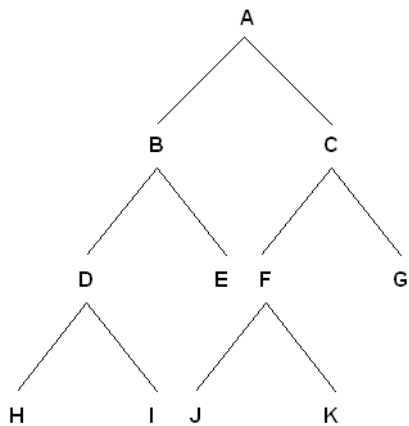


Table 61 shows how the hierarchy US Acct would be stored in the W_HIERARCHY_D table as follows:

Table 61. 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

Configuring General Ledger Account Hierarchies Using General Ledger Accounting Flexfield value sets definitions

Oracle EBS supports up to 30 segments in which to store accounting flex fields. Flex fields are flexible enough to support complex data configurations, for example:

- You can store data in any segments.
- You can use more or fewer segments per chart of account, as required.
- You can specify multiple segments for the same chart of account.

An Example Data Configuration for a Chart of Accounts

A single company might have a US Chart of Account and an APAC Chart of Account with the following data configuration:

Table 62. Example Chart of Accounts

Data 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.

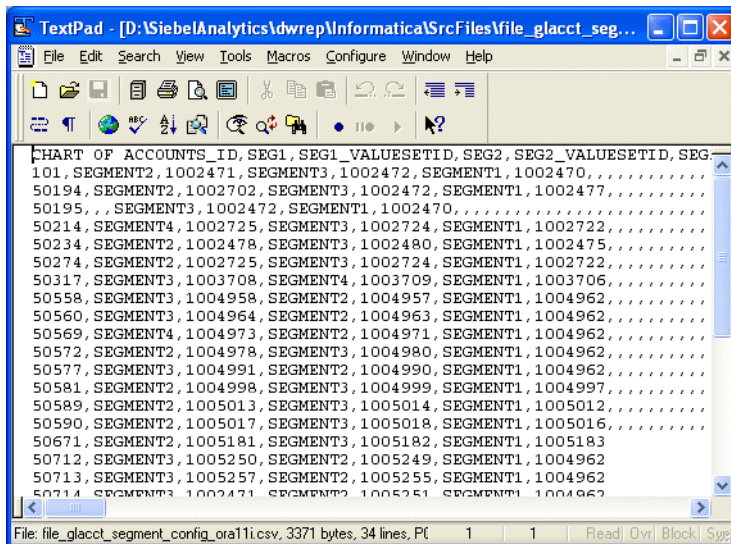
For example, we can store Company segments from COA 4256 and 4257 in segment1 column in W_GL_ACCOUNT_D; and Cost Center segments from COA 4256 and 4257 in segment2 column in W_GL_ACCOUNT_D, and so forth.

About Configuring the ETL Process for GL Accounting Flex fields

Before you run the ETL process for General Ledger Accounts, you need to specify the segments that you want to analyze. To specify the segments that you want to analyze, use the following ETL configuration file:

```
$PMSEVER/srcfile/file_glacct_segment_configur_ora11i.csv
```

Figure 28. Screen shot of file_glacct_segment_config_ora11i.csv File Opened in a Text Editor

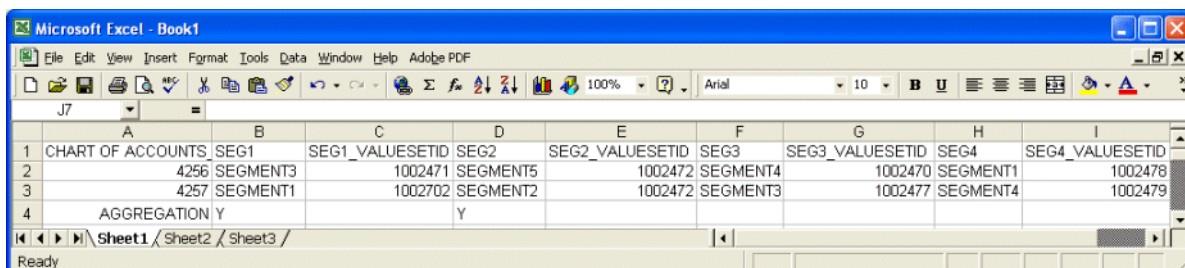


In the file_glacct_segment_config_ora11i.csv file, you need to specify the segments of the same type in the same column. For example, you might store all Cost Center segments from all chart of accounts in one column, and all Company segments from all chart of accounts in another column.

For example, you might want to do the following:

- analyze GL account hierarchies using only Company, Cost Center, Natural Account, and LOB. You are not interested in using Geography for hierarchy analysis.
- store all Cost Center segments from all COAs in ACCOUNT_SEG2_CODE column in W_GL_ACCOUNT_D.
- store all Natural Account segments from all COAs in ACCOUNT_SEG3_CODE column in W_GL_ACCOUNT_D.
- store all LOB segments from all COAs in ACCOUNT_SEG4_CODE column in W_GL_ACCOUNT_D.
- in W_GL_BALANCE_A (where you store GL account balances at aggregated level), you want to store GL account balances at Company and Cost Center level instead of at GL Code Combination level.

The screenshot below shows how the file_glacct_segment_config_ora11i.csv would be configured to implement the business requirements specified above.



To Set Up Hierarchies With General Ledger Accounting Flex fields

- 1 Use the /srcfile/file_glacct_segment_configur_ora11i.csv file to specify the segments that you want to analyze.

For more information about configuring the file_glacct_segment_configur_ora11i.csv file, see [About Configuring the ETL Process for GL Accounting Flex fields on page 324](#).

- 2 In the DAC, do the following:
 - a for the Subject Area 'Financials – General Ledger', in the 'Configuration Tag' tab, make sure that:
 - ❑ the tag 'Oracle – Extract Value Set Hierarchies' is unchecked for the 'Inactive' checkbox.
 - ❑ the tag 'Oracle – Extract FSG Hierarchies' is checked in the 'Inactive' checkbox.
 - b Click Assemble to redesign the subject area.

After the subject area has been redesigned, also redesign the execution plans which contain this subject area.

- 3 Run the ETL process for General Ledger Accounts.
- 4 In the RPD Physical Layer (using the Oracle BI Administration Tool), create additional aliases or change the names of the existing alias against the table W_HIERARCHY_D.

For example, if 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 In the Physical Layer of the Oracle BI Analytics Warehouse, use the Oracle BI Administration Tool to create joins in the physical layer from the new aliases that you created in the previous step, as follows:
 - Company Hierarchy will join to the segment column in W_GL_ACCOUNT_D that stores the Company segment.
 - ❑ Dim_Company_Hierarchy_D. HIER20_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG1_CODE
 - ❑ Dim_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 In the Business Model Layer of the Oracle BI Analytics Warehouse, use the Oracle BI Administration Tool to create additional dimensions using the new alias.

For more information about example dimensions that are installed with Oracle BI Applications, refer to the following dimensions in the Oracle Business Analytics Warehouse:

- Dim - GL ValueSetHierarchy1
- Dim - GL ValueSetHierarchy2
- Dim - GL ValueSetHierarchy3

- 8 In the Business Model Layer of the Oracle BI Analytics Warehouse, use the Oracle BI Administration Tool to create additional hierarchies using the dimensions that you created in the previous step.

For more information about example hierarchies that are installed with Oracle BI Applications, refer to the following hierarchies in the Oracle Business Analytics Warehouse:

- GL ValueSetHierarchy1
- GL ValueSetHierarchy2
- GL ValueSetHierarchy3

- 9 In the Presentation Layer of the Oracle BI Analytics Warehouse, use the Oracle BI Administration Tool to drag the new hierarchies into the presentation folder.

NOTE: You can rename the hierarchies in the Presentation Layer if required.

Configuring General Ledger Account Hierarchies Using Financial Statement Generator (FSG) Report Definition

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_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 calculate the amount in the previous row. An example of a columns calculation might be to calculate the difference between two other columns.

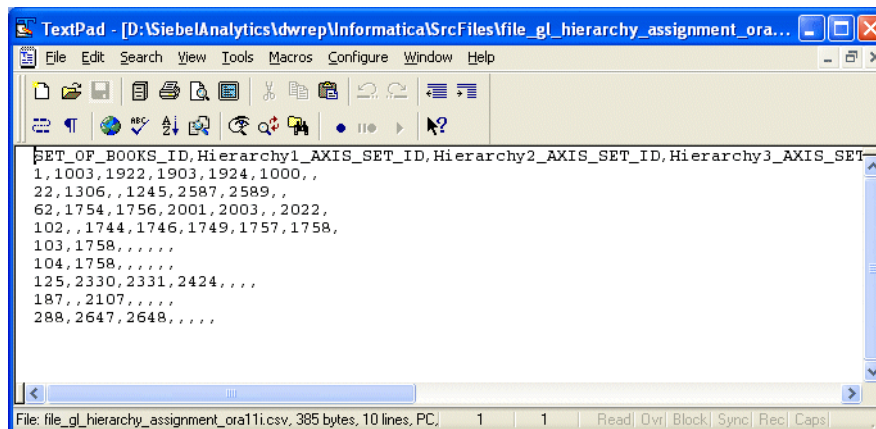
The hierarchies information is stored in the W_HIERARCHY_D table. Fact tables join to the W_HIERARCHY_D table via the General Ledger Account dimension table (W_GL_ACCOUNT_D). The W_HIERARCHY_D table contains six fields (HIER1, HIER2, HIER3, HIER4, HIER5, HIER6), which are foreign keys to the W_HIERARCHY_D.hier_key. Therefore, each General Ledger Code combination can participate in up to six different hierarchies.

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 hierarchies that you want to reference, use the following ETL configuration file:

\$PMSEVER/srcfile/file_glhierarchy_assignment_ora11i.csv

Figure 29. Screen shot of file_gl_hierarchy_assignment_ora11i.csv File Opened in a Text Editor



In the file_gl_hierarchy_assignment_ora11i.csv file, you specify the set of books ID. For each of the columns in W_HIERARCHY_D (for example, HIER1, HIER2, HIER3), you need to specify to KEY columns that reference each hierarchy. You specify hierarchies using the ID of the axis set that contains the hierarchy information.

The DATASOURCE_NUM_ID field specifies the data source to which the configurations apply. If you have multiple source systems, there might be a set of books across the multiple source systems with the same ID. Therefore, you need to use the DATASOURCE_NUM_ID value to distinguish between them.

To Set Up Hierarchies With Financial Statement Generator Report Definition

- 1 Use the /srcfile/file_glacct_segment_configur_ora11i.csv file to specify the segments that you want to analyze.

For more information about configuring the file_glacct_segment_configur_ora11i.csv file, see [About Configuring the ETL Process for Oracle Financial Statement Generator Report on page 328](#).
- 2 Run the ETL process for General Ledger Accounts.

Configuring the General Ledger Balance ID

This task is a step in the [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#). The General Ledger Balance ID controls the level at which the balance in W_GL_BALANCE_F is maintained. By default, the General Ledger Balance ID is maintained at the following granularity for Oracle 11i:

```
TO_CHAR(INP_SET_OF_BOOKS_ID) || '~' || TO_CHAR(INP_CODE_COMB_ID) || '~' || TO_CHAR(INP_ORG_ID)
```

However, if you want to maintain your General Ledger Balance ID at a different grain, you can redefine the GL Balance ID for any applicable mapplets.

To modify the General Ledger Balance ID

NOTE: To modify the General Ledger 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_ARTransactionFact_Adjust
 - mplt_SA_ORA_ARTransactionFact_ARScheduleDerive
 - mplt_SA_ORA_ARTransactionFact_CreditMemoApplication
 - mplt_SA_ORA_ARTransactionFact_ReceivableApplication
 - mplt_SA_ORA_GLCOGSFact
 - mplt_SA_ORA_GLOtherTransactionFact
 - mplt_SA_GLRevenueFact
- 1 In PowerCenter Designer, open the appropriate Oracle Applications 11i folder (for example, SDE_ORA11510_Adaptor). For more information, see [About the Name of the Oracle 11i Folder in Informatica on page 317](#).
 - 2 In Mapplet Designer, open the Source Adapter mapplet (for example, mplt_SA_ORA_APTransactionFact_LiabilityDistribution).

- 3 Double-click the Expression transformation to open the Edit Transformations window and select the Ports tab.
- 4 Edit the Balance ID definition in the EXT_GL_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.

Configuring AP Balance ID for Oracle Payables Analytics

This task is a step in the [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#). 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 11i.

By default, the Accounts Payable (AP) Balance ID is maintained at the following granularity:

GL_ACCOUNT_ID | '~' | VENDOR_SITE_ID | '~' | ORGANIZATION_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
- 1 In PowerCenter Designer, open the appropriate Oracle Applications 11i folder (for example, SDE_ORA11510_Adaptor). For more information, see [About the Name of the Oracle 11i Folder in Informatica on page 317](#).
 - 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 window.
 - 4 Click the Ports tab to edit the Balance ID definition in the EXT_NU_BALANCE_ID column.
 - 5 Validate and save your changes to the repository.
 - 6 Repeat steps 1 to 5 for each mapplet that is listed above.

Configuring AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics

This task is a step in the [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#). The AR Balance ID controls the level at which the balance in W_AR_BALANCE_F is maintained.

By default, the AR Balance ID is maintained at the following granularity:

```
set_of_books_id || '~' || code_combination_id || '~' || customer_id || '~' ||  
customer_site_use_id || '~' transaction_currency_code || '~' || org_id
```

However, if you want to maintain your AR balance at a different grain, you can redefine the Balance ID value in the applicable mapplets.

To modify the AR Balance ID

NOTE: To modify the AR Balance ID, you must modify the following mapplets:

- mplt_SA_ORA_ARTransactionFact_Adjust
 - mplt_SA_ORA_ARTransactionFact_ARScheduleDerive
 - mplt_SA_ORA_ARTransactionFact_CreditMemoApplication
 - mplt_SA_ORA_ARTransactionFact_ReceivableApplication
- 1 In PowerCenter Designer, open the appropriate Oracle Applications 11i folder (for example, SDE_ORA11510_Adaptor). For more information, see [About the Name of the Oracle 11i Folder in Informatica on page 317](#).
 - 2 In Mapplet Designer, open the mapplet (for example, mplt_SA_ORA_ARTransactionFact_Adjust).
 - 3 Double-click the Expression transformation to open the Edit Transformations window.
 - 4 Click the Ports tab to edit the Balance ID definition in the EXT_NU_AR_BALANCE_ID column.
 - 5 Validate and save your changes to the repository.
 - 6 Repeat steps 1 to 5 for each mapplet that is listed above.

Configuring the AR Adjustments Extract for Oracle Receivables Analytics

This task is a step in the [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#).

By default, Oracle Receivables Analytics extracts only approved adjustment entries against accounts receivable transactions. *Approved adjustments* are entries where the AR_ADJUSTMENTS_ALL.STATUS = A. If you want to extract additional types of AR adjustment entries, you can remove the filter in the Business Component mapplet. By modifying or removing the filter, you can extract other entries, such as those that require more research, those that are rejected, and those that are not accrued charges.

To modify the extract filter for Accounts Receivable adjustments

- 1 In PowerCenter Designer, open the appropriate Oracle Applications 11i folder (for example, SDE_ORA11510_Adaptor). For more information, see [About the Name of the Oracle 11i Folder in Informatica on page 317](#).
- 2 In Mapplet Designer, open the mplt_BC_ORA_ARTransactionFact_Adjust mapplet.
- 3 Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

In the SQL Query field and in the User Defined Join field, modify the statement:

`AND AR_ADJUSTMENTS_ALL.STATUS = A`
- 4 Validate and save your changes to the repository.
- 5 Repeat [Step 2](#) to [Step 4](#) for the mplt_BC_ORA_ARTransactionFact_AdjustPrimary mapplet.

Configuring the AR Schedules Extract

This task is a step in the [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#).

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.

You must modify both the regular mapplet (mplt_BC_ORA_ARTransactionFact_ARSchedules) as well as the primary extract mapplet (mplt_BC_ORA_ARTransactionFact_ARSchedulesPrimary). Repeat the following procedure for each mapplet.

To modify the extract filter for Accounts Receivable schedules

- 1 In PowerCenter Designer, open the appropriate Oracle Applications 11i folder (for example, SDE_ORA11510_Adaptor). For more information, see [About the Name of the Oracle 11i Folder in Informatica on page 317](#).
- 2 In Mapplet Designer, open the mplt_BC_ORA_ARTransactionFact_ARSchedules mapplet.
- 3 Double-click the Source Qualifier to open the Edit Transformations window.

In the User Defined Join field and in the SQL Query field, modify the statement:

`AND RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG(+) = Y`
- 4 Validate and save your changes to the repository.
- 5 Repeat [Step 2](#) to [Step 4](#) for the mplt_BC_ORA_ARTransactionFact_ARSchedulesPrimary mapplet.

Configuring the AR Cash Receipt Application Extract for Oracle Receivables Analytics

This task is a step in the [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#).

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 nonconfirmed applications.

You must modify both the regular mapplet (`mplt_BC_ORA_ARTransactionFact_ReceivableApplication`) as well as the primary extract mapplet (`mplt_BC_ORA_ARTransactionFact_ReceivableApplicationPrimary`).

To modify the extract filter for AR cash receipt application

- 1 In PowerCenter Designer, open the appropriate Oracle Applications 11i folder (for example, `SDE_ORA11510_Adaptor`). For more information, see [About the Name of the Oracle 11i Folder in Informatica on page 317](#).
- 2 In Mapplet Designer, open the `mplt_BC_ORA_ARTransactionFact_ReceivableApplication` mapplet.
- 3 Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

In the User Defined Join field and in the SQL Query field, modify the statement:

`AND NVL(AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG, 'Y') = Y`
- 4 Validate and save your changes to the repository.
- 5 Repeat [Step 2](#) to [Step 4](#) for the `mplt_BC_ORA_ARTransactionFact_ReceivableApplicationPrimary` mapplet.

Configuring the AR Credit-Memo Application Extract for Oracle's Siebel Receivables Analytics

This task is a step in the [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#).

By default, Oracle's 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 nonconfirmed, credit memos.

You must modify both the regular mapplet (mplt_BC_ORA_ARTransactionFact_CreditmemoApplication), as well as the primary extract mapplet (mplt_BC_ORA_ARTransactionFact_CreditmemoApplicationPrimary). Repeat the following procedure for each mapplet.

To modify the extract filter for Accounts Receivable Credit-Memo Application receipts

- 1** In PowerCenter Designer, open the appropriate Oracle Applications 11i folder (for example, SDE_ORA11510_Adaptor). For more information, see [About the Name of the Oracle 11i Folder in Informatica on page 317](#).
- 2** In Mapplet Designer, open the mplt_BC_ORA_ARTransactionFact_CreditmemoApplication mapplet.
- 3** Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

In the User Defined Join field and in the SQL Query field, modify the statement:

`AND NVL(AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG, 'Y') = Y`
- 4** Validate and save your changes to the repository.
- 5** Repeat [Step 2](#) to [Step 4](#) for the mplt_BC_ORA_ARTransactionFact_CreditmemoApplicationPrimary mapplet.

About the Customer Costs Lines and Product Costs Lines Tables for Oracle's Siebel Profitability Analytics

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.

Configuring the Customer Costs Lines and Product Costs Lines Tables for Oracle's Siebel Profitability Analytics

This task is a step in the [Process of Configuring Oracle Financial Analytics for Oracle 11i on page 316](#).

In Oracle's Siebel 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.

About PeopleSoft Trees in Oracle Financial Analytics

For PeopleSoft, the Siebel Financial Analytics application sources data from a data structure, called PeopleSoft Trees, to get information about the organization's General Ledger hierarchies, Profit hierarchies, Cost Centers' hierarchies, and so on.

PeopleSoft Trees are a flexible, generic way of constructing hierarchical summarization of a particular database fields in PeopleSoft for reporting purposes. Typically, entities such as Chart of Account fields (Account, Dept, and Project, and so on), items, locations, and so on, are organized into user-defined trees.

[Table 63](#) lists the PeopleSoft Trees the Siebel Financial Analytics application sources.

Table 63. PeopleSoft Trees for Oracle's Siebel Financial Analytics

PeopleSoft Data Structure	Repository Source Name	Description
PeopleSoft Tree	ACCTROLLUP	Account Rollup
PeopleSoft Tree	AR_BUSINESS_UNITS	AR Business Unit Roll-Up
PeopleSoft Tree	DEPARTMENTS	Manufacturing Departments
PeopleSoft Tree	OPERUNIT	Operating Unit

[Table 64](#) lists the mappings and sessions using PeopleSoft Trees.

Table 64. 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

15

Configuring Oracle Order Management and Fulfillment Analytics Option

This chapter describes how to configure certain objects for particular sources to meet your business needs.

It contains the following topics:

- [Overview of Oracle Order Management and Fulfillment Analytics Option on page 337](#)
- [Process of Configuring Oracle Order Management and Fulfillment Analytics Option for Oracle 11i on page 338](#)
- [Process of Aggregating Oracle Order Management and Fulfillment Analytics Option Tables on page 349](#)
- [About Tracking Multiple Products for Oracle Order Management and Fulfillment Analytics Option on page 353](#)
- [Adding Dates to the Order Cycle Time Table for Post-Load Processing on page 354](#)
- [About Configuring the Backlog Period Date for Oracle Order Management and Fulfillment Analytics Option on page 356](#)
- [Configuring the Backlog Period Date for Oracle Order Management and Fulfillment Analytics Option on page 358](#)
- [About the Grain at Which Currency Amounts and Quantities Are Stored on page 359](#)
- [About the Sales Order Dates on page 360](#)
- [Domain Values and CSV Worksheet Files for Oracle Order Management and Fulfillment Analytics Option on page 360](#)
- [Configuring Supply Chain Analytics for Oracle Order Management and Fulfillment Analytics Option on page 361](#)
- [Configuring Oracle's Siebel Financial Analytics for Oracle Order Management and Fulfillment Analytics Option on page 362](#)

Overview of Oracle Order Management and Fulfillment Analytics Option

The Oracle Order Management and Fulfillment Analytics Option application allows you to analyze the movement of sales opportunities through different stages of the sales cycle. This analysis includes insight into which items are booked, backlogged, and invoiced. It also provides you with information that allows you to evaluate the sales performance of individual sales representatives or departments. The Oracle Order Management and Fulfillment Analytics Option application contains the functional area, Orders and Revenue.

The Orders and Revenue functional area consists of orders, invoices, and backlog. Sales orders are the entry point for the sales process. Invoices are the exit point from the fulfillment process. *Backlogs* are points of congestion in your fulfillment process.

In the Oracle Order Management and Fulfillment Analytics Option application, two main types of backlog exist:

- Operational
- Financial

The scheduled, unscheduled, delinquent, and blocked backlogs belong to the Operational backlog. Two different sources can populate Bookings and Revenue:

- Oracle 11i
- Universal source

Orders and Revenue also requires post-load processing mappings to populate its tables.

Process of Configuring Oracle Order Management and Fulfillment Analytics Option for Oracle 11i

This section contains Oracle Order Management and Fulfillment Analytics Option configuration points that are specific to Oracle 11i. Of the three functional areas in the Oracle Order Management and Fulfillment Analytics Option application, Orders and Revenue is the only functional area that has prepackaged Oracle 11i business adapters to populate the warehouse tables; therefore, you can not find configuration points for the other two functional areas in this section.

To configure Oracle Order Management and Fulfillment Analytics Option for Oracle 11i, perform the following tasks:

- [Configuring Sales Order Lines Data Storage on page 339](#)
- [Tracking Multiple Attribute Changes in Bookings on page 342](#)
- [Configuring Sales Schedule Lines Data Storage on page 343](#)
- [Configuring Early and Late Tolerances for Shipments on page 345](#)
- [Configuring Sales Invoice Lines Data Storage on page 346](#)
- [Configuring Different Types of Backlog Calculations on page 346](#)
- [Accounting for Negative Values in Orders, Invoices, and Picks on page 348](#)
- [Process of Configuring the Order Lines Execution Plan and OTB ETL on page 349](#)

Related Topic

- [About Tracking Attribute Changes in Bookings on page 341](#)

Configuring Sales Order Lines Data Storage

This task is a step in the [Process of Configuring Oracle Order Management and Fulfillment Analytics Option for Oracle 11i](#) on page 338.

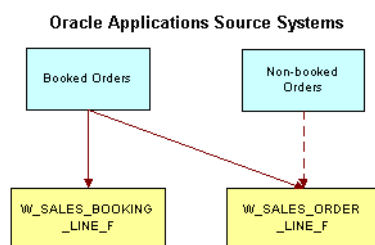
Sales order lines are the itemized lines that make up a sales order. This information is stored in the W_SALES_ORDER_LINE_F table. This topic describes how to modify the type of information stored in this table.

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 11i source system as shown in [Figure 30](#). 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, 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 30. Handling Booked and Nonbooked Orders

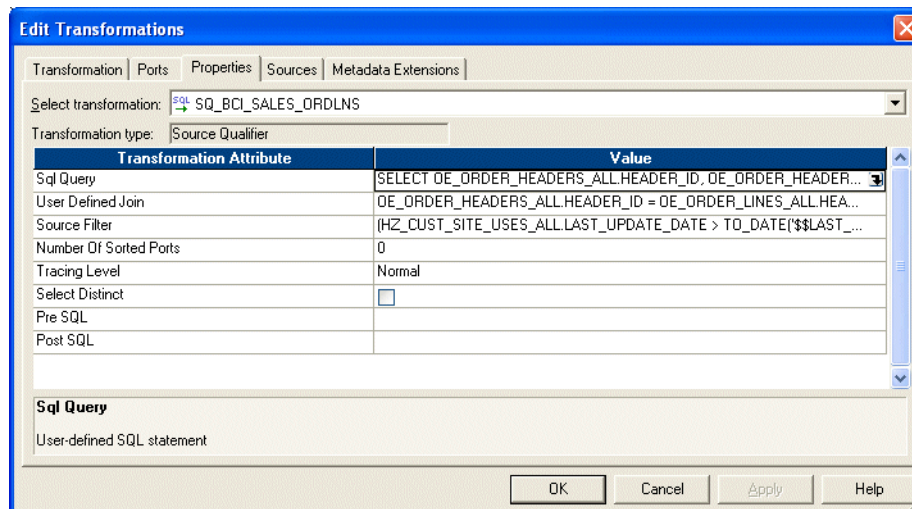


By default, only booked orders are loaded into the Sales Order Lines (W_SALES_ORDER_LINES_F) and Sales Booking Lines (W_SALES_BOOKING_LINE_F) tables. However, you can also load non-booked orders in Sales Order Lines (W_SALES_ORDERS_LINES_F).

To include nonbooked orders in the Sales Order Lines tables

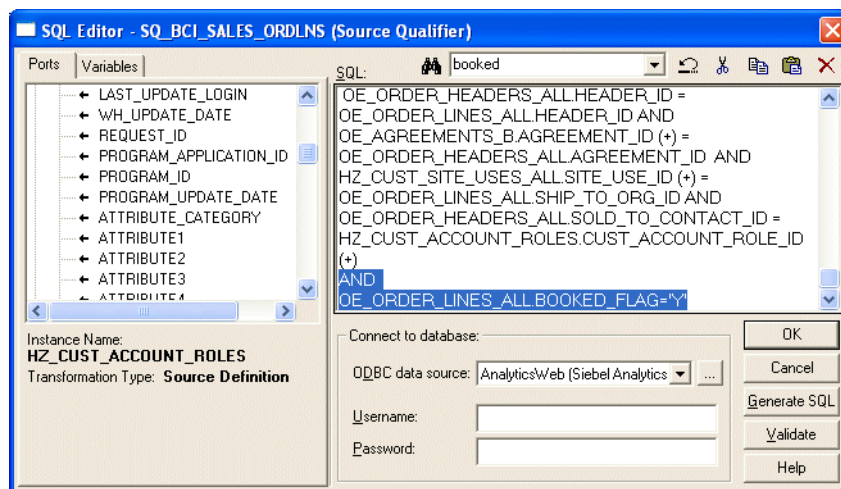
- 1 In PowerCenter Designer, open the SDE_ORA115<ver>_Adaptor folder.
- 2 Open the mplt_BC_ORA_SalesOrderFact mapplet in the Mapplet Designer.
- 3 Double-click the SQ_BCI_SALES_ORDLNS source qualifier to open the Edit Transformations box.

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_ORAC_SalesOrderLinesFact_Primary mapping.

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)
)|| TO_CHAR(INP_SALESREP_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.

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:

- a** In Informatica Designer, open the mplt_SA_ORA_SalesOrderLinesFact mapplet in the Mapplet Designer.
- b** Double click the MAPI_SALES_ORDLNS transformation to open the Edit Transformation box.
- c** Display the Ports tab.
- d** Select the EXP_SALES_ORDLNS transformation.
- e** 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 65](#).

Table 65. Oracle 11i: 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 65](#) is entered into the IA Bookings table (W_SALES_BOOKING_LINE_F) as shown in [Table 66](#).

Table 66. Oracle 11i: W_SALES_BOOKING_LINE_F Table Row After Day One Activity

SALES_ORDER_NUM	SALES_ORDER_ITEM	SALESREP_ID	SALES_QTY	NET_DOC_AMT	BOOKING_ID	BOOKED_ON_DT
1	1	1001	100	2500	1001	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 67](#).

Table 67. Oracle 11i: 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 68](#).

Table 68. Oracle 11i: W_SALES_BOOKING_LINE_F Table Row After Day Two Activity

SALES_ORDER_NUM	SALES_ORDER_ITEM	SALESREP_ID	SALES_QTY	NET_DOC_AMT	BOOKING_ID	BOOKED_ON_DT
1	1	1001	100	2500	1001	1-June-2000
1	1	1001	-100	-2500	1001	2-June-2000
1	1	1002	100	2500	1002	2-June-2000

Tracking Multiple Attribute Changes in Bookings

This task is a step in the [Process of Configuring Oracle Order Management and Fulfillment Analytics Option for Oracle 11i on page 338](#).

When you modify the default VAR_BOOKING_ID column, the SQL statement is configured as follows for Oracle 11i:

```
TO_CHAR(INP_LINE_ID)|| '~' || TO_CHAR(INP_INV_ITEM_ID)|| '~' || to_char(INP_WAREHOUSE_ID  
)
```

However, if you want to track changes based on more than one attribute, in the SQL statement you must concatenate the attribute column IDs in the VAR_BOOKING_ID column. For example, if you want to track changes in Salespersons and Customer, then concatenate the technical name IDs in the VAR_BOOKING_ID column as follows:

```
TO_CHAR(INP_LINE_ID)|| '~' || TO_CHAR(INP_INV_ITEM_ID)|| '~' || TO_CHAR(INP_WAREHOUSE_ID  
)|| '~' || TO_CHAR(INP_SALESREP_ID)|| '~' || TO_CHAR(INP_CUSTOMER_ID)
```

To track dimensional attribute changes in bookings

- 1** In PowerCenter Designer, open the SDE_ORA115<ver>_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.

Configuring Sales Schedule Lines Data Storage

This task is a step in the [Process of Configuring Oracle Order Management and Fulfillment Analytics Option for Oracle 11i on page 338](#).

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.

About Loading Bookings at the Schedule Line Level

As initially configured for Oracle 11i, 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 [Figure 31](#). There are two subject areas in the SDE_ORA115<ver>_Adaptor containers:

- Enterprise Sales - Booking Lines & Order Lines

■ Enterprise Sales - Booking Lines & Schedule Lines



Figure 31. Oracle 11i: Bookings at the Order Line Level

The execution plans that are installed with Oracle BI Applications use the Enterprise Sales - Booking Lines & Order Lines subject area by default. If you want to load booking lines at the schedule line level, create a new execution plan and include the Enterprise Sales - Booking Lines & Schedule Lines subject area instead of Enterprise Sales - BookingLines & Order Lines.

Bookings may be recorded at the Sales Schedule Line level instead of the Sales Order Line level. At the Sales Schedule Line level, bookings provide a more granular view, as the orders are segmented by schedule line. Bookings recorded at the Schedule Line level provide one row in the Bookings table for each schedule line, as shown in [Figure 32](#). 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.

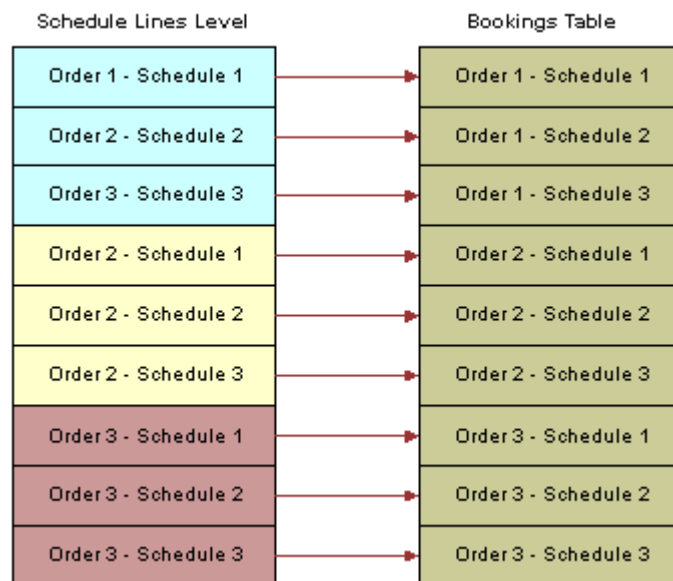


Figure 32. Oracle 11i: Bookings at the Schedule Line Level

Configuring Early and Late Tolerances for Shipments

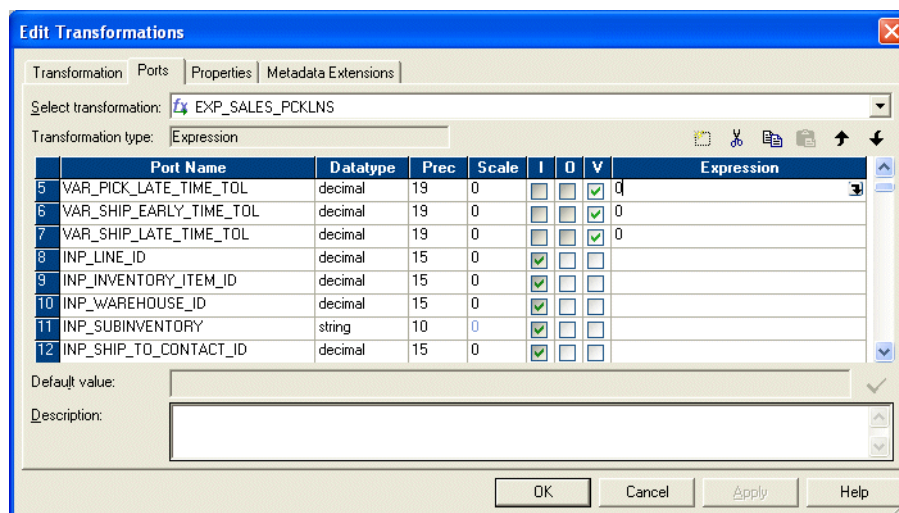
This task is a step in the [Process of Configuring Oracle Order Management and Fulfillment Analytics Option for Oracle 11i](#) on page 338.

You configure the definition of early and late shipments by editing the EXP_SALES_PCKLNS expression in the mplt_SA_ORA_SalesPickLinesFact mapplet. The mplt_SA_ORA_SalesPickLinesFact mapplet is used by the SDE_ORASalesPickLinesFact mapping.

This mapplet compares the pick date and ship date with the scheduled ship date to determine whether or not orders are late.

To configure early and late tolerances for shipments

- 1 In PowerCenter Designer, open the SDE_ORA115<ver>_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.

Configuring Sales Invoice Lines Data Storage

This task is a step in the [Process of Configuring Oracle Order Management and Fulfillment Analytics Option for Oracle 11i on page 338](#).

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.

Configuring the Sales Invoice Extract

By default, the Oracle Order Management and Fulfillment Analytics Option application is configured to extract completed sales invoices when performing the Sales Invoice data extract. Oracle 11i uses 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.

To extract incomplete sales invoices, as well as complete invoices, remove the extract filter statement.

To remove the extract filter for sales invoices

- 1 In PowerCenter Designer, open the SDE_ORA115<ver>_Adaptor folder.
- 2 Open the mplt_BC_ORA_SalesInvoiceLinesFact maplet in the Maplet 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 maplet.

Configuring Different Types of Backlog Calculations

This task is a step in the [Process of Configuring Oracle Order Management and Fulfillment Analytics Option for Oracle 11i on page 338](#).

Backlog information is stored in the W_SALES_BACKLOG_LINE_F and W_SALES_BACKLOG_HISTORY_F tables. This topic describes how to modify the type of information stored in these tables. Many types of backlog exist in the Oracle Order Management and Fulfillment Analytics Option 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.

Adding Closed Orders to Backlog Calculations

By default, the Oracle Order Management and Fulfillment Analytics Option application only extracts open sales orders from the Sales Order Lines (W_SALES_ORDER_LINE_F) table and Sales Schedule Lines table (W_SALES_SCHEDULE_LINE_F) for backlog calculations to populate the Backlog tables. *Open sales orders* are defined as orders that are not canceled or not complete. The purpose in extracting only open orders is that in most organizations those orders that are closed are no longer a part of backlog. However, if you want to extract sales orders that are marked as closed, you may remove the default filter condition from the extract mapping.

For example, assume your customer orders ten items. Six items are invoiced and shipped, but four items are placed on operational and financial backlog. This backlog status continues until one of two things happens:

- The items are eventually shipped and invoiced.
- The remainder of the order is canceled.

If you choose to extract sales orders that are flagged as closed, you must remove the condition in the Backlog flag. To do so, use the following procedure.

The BACKLOG_FLAG in the W_SALES_ORDER_LINE_F table is also used to identify which sales orders are eligible for backlog calculations. By default, all sales order types have their Backlog flag set to Y. As a result, all sales orders are included in backlog calculations.

To remove open order extract filters

- 1** In PowerCenter Designer, open the SDE_ORA115<ver>_Adaptor folder.
- 2** Open the mplt_BC_ORA_SalesOrderLinesFact maplet in the Maplet Designer.
- 3** Double click on the EXP_SALES_ORDLNS, and display the Ports tab.
- 4** Edit the VAR_OPR_BACKLOG_FLG and remove the OPEN_FLAG='Y'.
- 5** Edit the VAR_FIN_BACKLOG_FLG and remove the OPEN_FLAG='Y'.
- 6** Open the mplt_BC_ORA_SalesScheduleLinesFact maplet in the Maplet Designer.
- 7** Double click on the EXP_SALES_SCHLNS, and display the Ports tab.
- 8** Edit the VAR_OPR_BACKLOG_FLG and remove the OPEN_FLAG='Y'.
- 9** Edit the VAR_FIN_BACKLOG_FLG and remove the OPEN_FLAG='Y'.
- 10** Validate and save your changes to the repository.

Accounting for Negative Values in Orders, Invoices, and Picks

This task is a step in the [Process of Configuring Oracle Order Management and Fulfillment Analytics Option for Oracle 11i](#) on page 338.

By default, the Oracle Business Analytics Warehouse does not use negative values in the quantity or amount columns for the W_SALES_INVOICE_LINE_F table or the W_SALES_ORDER_LINE_F table. However, you can make these values negative using a column called VAR_NEGATIVE_SIGN. By default, this column has the value 1.0. To make the values negative, modify the column value to be -1.

For example, to account for a negative return value for a Return Material Authorization (RMA) or for a negative value in a credit memo, you can use a conditional statement to define the VAR_NEGATIVE_SIGN column.

Assume that the S14 column in SO_LINES_ALL table has been configured in Oracle 11i to have the value 30 if the order line is a return. You can use this identifier for returned orders as a condition for setting the VAR_NEGATIVE_SIGN column to be -1. To do this, you can modify the VAR_NEGATIVE_SIGN column's definition in the MPLT_SAO_SALES_ORDLNS as follows:

```
DECODE(INP_LINES_S14, 30, -1, 1)
```

For Oracle 11i, the VAR_NEGATIVE_SIGN column is available in the following Source Adapters:

- mplt_SA_ORA_SalesInvoiceLinesFact
- mplt_SA_ORA_SalesOrderLinesFact
- mplt_SA_ORA_SalesPickLinesFact
- mplt_SA_ORA_SalesScheduleLinesFact

In Oracle 11i the VAR_NEGATIVE_SIGN column's value is set based on the type of order line.

To configure mapplets to account for negative values

- 1** In PowerCenter Designer, open the SDE_ORA115<ver>_Adaptor folder.
- 2** Open the applicable Source Adapter mapplet (for example, mplt_SA_ORA_SalesInvoiceLinesFact).
- 3** Double-click the Expression transformation to open the Edit Transformation box.

For example, if you are editing the mplt_SA_ORA_SalesInvoiceLinesFact mapplet, the Expression transformation is EXP_SALES_PCKLNS.

- 4** In the Ports tab, edit Expression value for the VAR_NEGATIVE_SIGN port.

For example, if the S14 column in SO_LINES_ALL table has been configured in Oracle 11i to have the value 30 if the order line is a return, then you can use this identifier for returned orders as a condition for setting the VAR_NEGATIVE_SIGN column to be -1. To do so, you would set the VAR_NEGATIVE_SIGN column's definition as follows:

```
DECODE(INP_LINES_S14, 30, -1, 1)
```

- 5** Validate and save your changes to the repository.

Process of Configuring the Order Lines Execution Plan and OTB ETL

The Backlogs and Cycle Lines (OTB ETL) component assumes that the Oracle Order Line Tables have been updated with shipping and invoicing information (for example, using Oracle programs). If the Oracle Order Line Tables have not been updated with shipping and invoicing information, you need to update the OTB ETL and Execution Plan as follows:

To configure the Order Lines Execution Plan and OTB ETL

- 1** In PowerCenter Designer, open the PLP folder.
- 2** Open the PLP_SalesCycleLinesFact maplet in the Maplet Designer.
- 3** Double-click the SQ_W_SALES_ORDER_LINE_F source qualifier to open the Edit Transformations box.
- 4** Display the Properties tab.
- 5** For the SQL Query Transformation Attribute, select the down arrow in the Value field to display the SQL Editor box.
- 6** In the SQL box, replace the SQL text 'X.TOTAL_SHIPPED_QTY' with 'PICKLINE.TOTAL_SHIPPED_QTY'.
- 7** In the SQL box, replace the SQL text 'X.TOTAL_INVOICED_QTY' with 'IVCLINE.TOTAL_INVOICE_QTY'.
- 8** Validate and save your changes to the repository.
- 9** Log into the DAC (for more information, see [To log in to the DAC on page 92](#)).
- 10** In the Design tab, expand the Configuration Points node, and select the Sales PLP Optional Tasks.
- 11** Activate the appropriate Subject Areas.

Process of Aggregating Oracle Order Management and Fulfillment Analytics Option Tables

This section contains Oracle Order Management and Fulfillment Analytics Option 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:

- [Configuring the Sales Invoice Lines Aggregate Table on page 351](#)
- [Configuring the Sales Order Lines Aggregate Table on page 353](#)

Related Topics

- [About Configuring the Sales Invoice Lines Aggregate Table on page 350](#)
- [About Configuring the Sales Order Lines Aggregate Table on page 351](#)

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 and the image capture phase.

You need to configure two parameters to aggregate the Sales Invoice Lines table for your incremental run:

- TIME_GRAIN
- PHASE

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 Sales Invoice Lines aggregate table is fully loaded from the base table in the initial ETL run. The table can grow to millions of records. Thus, the Sales Invoice aggregate table is not fully reloaded from the base table after each incremental ETL run. Oracle Business Analytics Warehouse minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. This process is described below.

- Oracle Business Analytics Warehouse finds the records to be deleted in the base table since the last ETL run, and loads them into the W_SALES_ORDER_LINE_TMP table. The measures in these records are multiplied by (-1). The mapping responsible for this task is SIL_SalesInvoiceLinesAggregate_Derive_PreSoftDeleteImage, which is run before SIL_SalesInvoiceLinesFact_SoftDelete deletes the records from the base table.

- Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the `W_SALES_ORDER_LINE_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is `SIL_SalesInvoiceLinesFact_Derive_PreLoadImage`, which is run before `SIL_SalesInvoiceFact` deletes the records from the base table.
- Oracle Business Analytics Warehouse finds the inserted or updated records in the base table since the last ETL run, and loads them into the `W_SALES_ORDER_LINE_TMP` table, without changing their sign. The mapping responsible for this task is `SIL_SalesInvoiceLinesFact_Derive_PreLoadImage`, which is run before `PLP_SalesInvoiceLinesFact_Derive_PostLoadImage` updates or inserts records in the base table.
- Oracle Business Analytics Warehouse aggregates the `W_SALES_ORDER_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`).

Configuring the Sales Invoice Lines Aggregate Table

This task is a step in the [Process of Aggregating Oracle Order Management and Fulfillment Analytics Option Tables on page 349](#).

To load the Sales Invoice Lines aggregate table (`W_SALES_INVOICE_LINE_A`), you need to configure the `parameterfileDW.txt` file, and run the initial and then the incremental workflows.

To configure the parameterfileDW.txt parameters file

- 1 Open the `parameterfileDW.txt` file in a text editor.
This file is located in the `OracleBI\DAC\Informatica\parameters\input` folder.
- 2 Replace the default values with your new values.
For a list of values for each parameter see the [About Configuring the Sales Invoice Lines Aggregate Table on page 350](#).
- 3 Save and close the file.

About Configuring the Sales Order Lines Aggregate Table

The Sales Order Lines aggregate table (`W_SLS_ORDLNS_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.

The aggregation processes use the following Teradata parameters:

- Hint_Tera_Pre_Cast
- Hit_Tera_Post_Cast

For the incremental ETL run, you need to configure the time aggregation level and image capture phase.

You need to configure two parameters to aggregate the Sales Order Lines table for your incremental run:

- TIME_GRAIN
- PHASE

The TIME_GRAIN parameter has a preconfigured value of Month. The possible values for the GRAIN parameter are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

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_SalesInvoiceLinesAggregate_Derive_PreSoftDeleteImage, which is run before SIL_SalesInvoiceLinesFact_SoftDelete deletes the records from the base table.
- Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the W_SALES_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 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).

Configuring the Sales Order Lines Aggregate Table

This task is a step in the [Process of Aggregating Oracle Order Management and Fulfillment Analytics Option Tables on page 349](#).

To load the Sales Order Lines aggregate table (W_SLS_ORDLNS_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 file_parameters_plp.csv parameters file

- 1 Open the parameterfileDW.txt file in a text editor.

This file is located in the OracleBI\DAC\Informatica\parameters\input folder.

- 2 Replace the default values with your new values.

For a list of values for each parameter see the [About Configuring the Sales Order Lines Aggregate Table on page 351](#).

- 3 Save and close the file.

About Tracking Multiple Products for Oracle Order Management and Fulfillment Analytics Option

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 69](#). 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 69. 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 69](#) 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 70](#).

Table 70. 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

Adding Dates to the Order Cycle Time Table for Post-Load Processing

It is recommended that you do not change the structure of any of the data warehouse tables, with the exception of the Order Cycle Time table—W_SALES_CYCLE_LINE_F_HOLD_TMP. In particular, the Order Cycle Time table may not store every sales-related date. As a result, you want to create a new date field in this table to accommodate your additional dates.

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_HOLD_TMP), then you have to modify the M_PLP_SALES_CYCHDR_LOAD mappings that take the dates from the W_* tables and load them into the Cycle Time table.

NOTE: Be sure that the date is already being extracted and stored in IA and OD data warehouse tables.

To add dates to the Cycle Time table load

- 1 In the PowerCenter Designer, open the Configuration for Post Load Processing folder.
- 2 In Warehouse Designer, modify the table definition for the target table to verify that it has a field to store this date.

For example, if you are loading the Validated on Date in the W_SALES_CYCLE_LINE_F_HOLD_TMP table, then you need to create a new column, VALIDATED_ON_DT, and modify the target definition of the W_SALES_CYCLE_LINE_F_HOLD_TMP 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_HOLD_TMP source table.

- 4 Create the table in the database with the new table structure.

TIP: If you have already loaded data in the W_SALES_CYCLE_LINE_F_HOLD_TMP table, then make sure that you backup the data before you recreate this table.

- 5 Modify the M_PLP_SALES_CYCLNS_INCR_LOAD mapping to select the new column from any of the following source tables, and load it to the W_SALES_CYCLE_LINE_F_HOLD_TMP target table:

- W_SALES_ORDER_LINE_F
- W_SALES_INVOICE_LINE_F
- W_SALES_PICK_LINE_F
- W_SALES_SCHEDULE_LINE_F

- 6 Modify The Source Qualifier SQL Override for the mapping, and map the column in the Transformation to map it to the target table.

About Configuring the Backlog Period Date for Oracle Order Management and Fulfillment Analytics Option

The Backlog table (W_BLGLNS) stores backlog data for the current month. In contrast, the Backlog History table (W_BLGHIS) 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.

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 71](#).

Table 71. Oracle 11i: 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 72](#).

Table 72. Oracle 11i: 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~~ 02/01/2001	02/28/2001	~~10~~ 5

No further activity happens until February 28. On February 28, 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 73](#).

Table 73. 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	~~02/01/2001~~ 02/02/2001	02/28/2001	~~10~~ 5
2	02/28/2001	02/28/2001	50

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 74](#).

Table 74. Oracle 11i: 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	~~02/01/2001~~ 02/02/2001	02/28/2001	5
2	02/28/2001	02/28/2001	50
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 74](#), you can see that sales order number 1 and 2 ended up with 5 and 50 financial backlogged items respectively. You do not have visibility into what the initial financial backlogged item quantities were for both of these 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 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 75](#) provides a list of all applicable mappings and their corresponding Expression transformations that you must modify.

Table 75. Oracle 11i: Backlog History Applicable Mappings and Expression Transformations

Mapping	Expression Transformation
M_I_SALES_BLGLNS_LOAD	EXP_SALES_BLGLNS

The backlog history period is monthly by default. The default SQL statement in the Expression transformation of the listed mappings is as follows:

```
trunc(DATE_DIFF(LAST_DAY(CAL_DAY_DT), to_date('01-JAN-1900', 'DD-MON-YYYY'), 'DD')) + 2415021
```

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).

Configuring the Backlog Period Date for Oracle Order Management and Fulfillment Analytics Option

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.

To configure the backlog period date

- 1 In PowerCenter Designer, open the SDE_ORA115<ver>_Adaptor folder.
- 2 Open the applicable Sales Backlog Lines mapping.
- 3 Double-click the Expression transformation to display the BACKLOG_PERIOD_DK port.
The SQL statement in this port's expression contains the backlog period date.
- 4 In the Ports tab, modify the default SQL statement for the BACKLOG_PERIOD_DK.

For example, if you want to store backlog history at the weekly level, replace the existing statement:

```
trunc(DATE_DIFF(LAST_DAY(CAL_DAY_DT), to_date('01-JAN-1900', 'DD-MON-YYYY'), 'DD')) +
2415021
```

with this new statement:

```
trunc(DATE_DIFF(CAL_WEEK_END_DT, to_date('01-JAN-1900', 'DD-MON-YYYY'), 'DD')) +
2415021
```

```
TO_CHAR(CAL_WEEK_END_DT, 'J')
```

NOTE: The CAL_WEEK_END_DT is not prepackaged to be extracted from W_DATES; to use this calculation you have to extract this data from W_DATES and pass it to the Expression transformation.

- 5 Validate and save your changes to the repository.

About the Grain at Which Currency Amounts and Quantities Are Stored

Quantities and currency amounts may be stored at different grains in the Order Cycle Time (W_SALES_CYCLE_LINE_F_HOLD_TMP) table, because of their unique ability to be rolled up.

Consider as an example a situation where a customer orders one package, which includes a computer, scanner, printer, and two speakers. In addition to the package, the customer also orders one monitor, which is not included in the package deal. In this case, the sales quantities are listed for the parent line item as well as for each child line items. However, the currency amounts are only listed for the parent line items; they are not listed for the individual child line items. [Table 76](#) illustrates this example.

Table 76. Storing Currency Amounts at the Parent Line Level in Order Cycle Time table

Key_ID	SALES_ORDER_NUM	PRODUCT_ID	ORDHD_KEY_ID	ORDLN_KEY_ID	SALES_QTY	Currency Amount	Relation
Line_1	1000	Package	1000	Line_1	1	\$1000	Parent A
Line_2	1000	Computer	1000	Line_1	1	-	Child A1
Line_3	1000	Scanner	1000	Line_1	1	-	Child A2
Line_4	1000	Printer	1000	Line_1	1	-	Child A3
Line_5	1000	Speaker	1000	Line_1	2	-	Child A4
Line_6	1000	Monitor	1000	Line_5	1	\$400	Parent B (no children)

Consider another example. In this example, a customer orders the same package, which includes a computer, scanner, printer, and two speakers. In addition to the package, the customer also orders one monitor, which is not included in the package deal. In this case, the quantities are provided for the parent and child line item levels. In addition, the currency amounts are also listed for both the parent and child line item levels. [Table 77](#) illustrates this example.

Table 77. Storing Currency Amounts at the Child Line Level in Order Cycle Time table

Key ID	Order #	Product	ORDHD_KEY_ID	ORDLN_KEY_ID	Relationship	Qty	Currency Amount
Line_1	1000	Package	1000	Line_1	Parent A	1	-
Line_2	1000	Computer	1000	Line_1	Child A1	1	725
Line_3	1000	Scanner	1000	Line_1	Child A2	1	75
Line_4	1000	Printer	1000	Line_1	Child A3	1	150
Line_5	1000	Speaker	1000	Line_1	Child A4	2	50
Line_6	1000	Monitor	1000	Line_5	Parent B (no children)	1	\$400

For more information on parent and child relationships, see [About Tracking Multiple Products for Oracle Order Management and Fulfillment Analytics Option on page 353](#).

About the Sales Order Dates

The Order Cycle Time table (W_SALES_CYCLE_LINE_F) store many sales-related dates, which are sourced from the following tables:

- W_SALES_ORDER_LINE_F
- W_SALES_INVOICE_LINE_F
- W_SALES_PICK_LINE_F
- W_SALES_SCHEDULE_LINE_F
- W_SALES_CYCLE_LINE_F_HOLD_TMP

Domain Values and CSV Worksheet Files for Oracle Order Management and Fulfillment Analytics Option

[Table 78](#) lists the CSV worksheet files and the domain values for Oracle Order Management and Fulfillment Analytics Option in the \$pmserver\LkpFiles folder.

For more information on configuring domain values with CSV worksheet files, see [About Domain Values on page 238](#) and [Configuring the Domain Value Set with CSV Worksheet Files on page 243](#).

Table 78. Domain Values and CSV Worksheet Files for Oracle Order Management and Fulfillment Analytics Option

Worksheet File Name	Description	Session
domainValues_InvoiceTypes_ora11i.csv	Lists the Invoice Document Type column and the corresponding domain values for the Oracle 11i application.	SDE_ORA_TransactionTypeDimension_SalesInvoiceLines
domainValues_PickTypes_ora11i.csv	Lists the Picking Document Type column and the corresponding domain values for the Oracle 11i application.	SDE_ORA_TransactionTypeDimension_SalesPickLines
domainValues_OrderTypes_ora11i.csv	Lists the Order Document Type column and the corresponding domain values for the Oracle 11i application.	SDE_ORA_TransactionTypeDimension_SalesOrderLines
domainValues_PickStatuses_ora11i.csv	Lists the Picking Status Code and the Status Desc columns, and the corresponding domain values for the Oracle 11i application.	SDE_ORA_StatusDimension_SalesPickLines
domainValues_InvoiceStatuses_ora11i.csv	Lists the Invoice Status Code and the Status Desc columns, and the corresponding domain values for the Oracle 11i application.	SDE_ORA_StatusDimension_SalesPickLines
domainValues_PayMethodCode_sapr3.csv	Lists the method code column and the corresponding domain value for the SAP R/3 application.	SDE_ORA_PaymentMethodDimension

Configuring Supply Chain Analytics for Oracle Order Management and Fulfillment Analytics Option

The Oracle Order Management and Fulfillment Analytics Option application uses tables that are also used in Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics).

For Oracle 11i you need to use the following configuration steps for Supply Chain Analytics to configure Oracle Order Management and Fulfillment Analytics Option:

- [Configuring Quantity Types for Product Transactions on page 294](#)
- [Configuring the Country Region and State Region Name on page 295](#)
- [Configuring the State Name on page 296](#)
- [Configuring the Country Name on page 296](#)

- [Configuring the Make-Buy Indicator on page 297](#)

Configuring Oracle's Siebel Financial Analytics for Oracle Order Management and Fulfillment Analytics Option

The Oracle Order Management and Fulfillment Analytics Option application uses tables that are also used in the Siebel Financial Analytics application.

For Oracle 11i, you need to use the following configuration steps for Siebel Financial Analytics to configure Oracle Order Management and Fulfillment Analytics Option:

- [Extracting Data Posted at the Detail-Level for Oracle 11i on page 317](#)
- [Configuring How GL Balances Is Populated on page 318](#)
- [Configuring the AP/AR Aging Tables on page 318](#)
- [Mapping Oracle General Ledger and Profitability Analytics Account Numbers to Group Account Numbers on page 319](#)
- [Configuring Oracle General Ledger and Profitability Analytics Transaction Extracts on page 321](#)

16 Configuring Oracle Workforce Analytics

This chapter describes how to configure certain objects for particular sources to meet your business needs. It contains the following topics:

- [Overview of Oracle Workforce Analytics on page 363](#)
- [Process of Configuring Workforce Operations for Oracle 11i on page 364](#)
- [Process of Configuring Workforce Payroll for Oracle 11i on page 374](#)
- [About Aggregating the Payroll Table for Oracle Workforce Analytics on page 377](#)
- [Aggregating the Payroll Table for Oracle Workforce Analytics on page 378](#)
- [Domain Values and CSV Worksheet Files for Oracle Workforce Analytics on page 379](#)
- [Configuring Domain Values and CSV Worksheet Files for Oracle Workforce Analytics on page 384](#)

Overview of Oracle Workforce Analytics

Oracle Workforce Analytics contains information for workforce operations, employee events, and payroll.

The Oracle Workflow Analytics application has the following functional areas:

- **Compensation.** Workforce 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 Workforce Compensation area provides the information your Workforce 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, workforce 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 help Human Resources departments prepare government-required reports.
- **Workforce Profile.** The Workforce Profile functional area provides you with the tools to separate sensitive from nonsensitive information, and to restrict access to sensitive data. Sensitive information includes such data as ethnicity, age, native language, marital status, and performance ratings. Nonsensitive information includes information such as job title, work location, and position status.

About Configuring Workforce Operations for Oracle 11i

There are two dimensions in the U.S. Statutory Compliance configuration for the Oracle 11i folder for which there are mandatory changes to the configuration information—the Employees dimension and the Jobs dimension. Failure to configure the following mandatory configuration information can result in incorrect data being fed to reports and other ETL processes.

This section also provides the necessary information on configuring the Workforce Profile functional area for Oracle 11i—configuring address types, configuring phone types, modifying the derive flag, and modifying the snapshot extract date.

NOTE: Currently, no configuration changes are required for Oracle 11i for the Retention functional area.

Process of Configuring Workforce Operations for Oracle 11i

This section contains Workforce Operations configuration information that is specific to Oracle 11i.

To configure Workforce Operations for Oracle 11i, perform the following tasks:

- [Configuring the Employees Dimension for U.S. Statutory Compliance on page 364](#)
- [Configuring the Jobs Dimension for U.S. Statutory Compliance on page 370](#)
- [Configuring Address Types for Workforce Profile on page 372](#)
- [Configuring Phone Types for Workforce Profile on page 373](#)

Configuring the Employees Dimension for U.S. Statutory Compliance

This task is a step in the [Process of Configuring Workforce Operations for Oracle 11i on page 364](#).

Within the Employees dimension, there are mandatory changes to the configuration information about 3 sets of columns, as follows:

- Ethnic Group Code (W_ETHNIC_GRP_CODE)

- Ethnic Group Description (W_ETHNIC_GRP_DESC)
- Veteran Status Code (W_VETERAN_STATUS_CODE)
- Veteran Status Description (W_VETERAN_STATUS_DESC)
- Sex Code (W_SEX_MF_CODE)
- Sex Description (W_SEX_MF_DESC)

Configuring Ethnic Group Codes

The domain values for Ethnic Group Code and Ethnic Group Description in Oracle Business Analytics Warehouse are shown in [Table 79](#).

Table 79. Domain Values for Ethnic Group Code and Ethnic Group Description

Ethnic Group Code	Ethnic Group Description
1	White
2	Black
3	Asian
4	American Indian/Alaskan Native
5	Native Hawaiian or Other Pacific Islander
6	Hispanic or Latino (White)
7	Hispanic or Latino (All Other Races)
8	Race Unknown
9	Others

It is recommended that you do not change the pre-packaged domain values. If you do so, there will be impacts on the Dashboards and the reporting metadata. You should limit yourselves to just 'mapping' the source supplied values against the packaged set of domain values. A sample set of domain value 'maps' is provided as a CSV lookup file – you should be modifying that file according to values that your source provides. An example of the CSV lookup is shown in [Table 80](#).

Table 80. Sample Source Values for Ethnic Group Code and Ethnic Group Description

ETHNIC_CODE	W_ETHNIC_GRP_CODE	W_ETHNIC_GRP_DESC
1	1	White
2	2	Black
8	2	Black
3	7	Hispanic or Latino (all other races)
10	7	Hispanic or Latino (all other races)
4	3	Asian
11	3	Asian
5	5	Native Hawaiian or Other Pacific Islander
12	5	Native Hawaiian or Other Pacific Islander
6	4	American Indian/Alaskan Native
7	4	American Indian/Alaskan Native
9	6	Hispanic or Latino (White)
BA	2	Black
BC	2	Black
BO	2	Black
C	3	Asian
I	3	Asian
O	9	Others
P	3	Asian
W	1	White

NOTE: This example CSV lookup file specifies that the Ethnic Code "7" signifies the group 'American Indian/Alaskan Native'. However, you should use the Ethnic Code used by your source system.

To modify the Ethnic Group code

- 1 Edit the file "domainValues_Employee_Ethnic_Group_Code_ora11i.csv" in the \$pmserver\LkpFiles directory using in a text editor.

- 2 Modify the W_ETHNIC_CODE column (the source code) as required so that it maps correctly in the corresponding domain values in the table above.
- 3 Save the file.

Configuring Ethnic Group Descriptions

For the Ethnic Group Description, make sure that the description used with the source-specified code matches the description defined in the domain values. For example, if the source system uses the description, African American, instead of Black for code 2, the description must be modified to reflect the descriptive value of Black.

To modify the Ethnic Group description

- 1 Edit the file "domainValues_Employee_Ethnic_Group_Code_ora11i.csv" in the \$pmserver\LkpFiles directory using in a text editor.
- 2 Modify the W_ETHNIC_GRP_DESC column (the domain value description) as required (that is, as specified by your source system).
- 3 Save the file.

NOTE: You might need to change the metric names in the reporting data to use more suitable names.

Configuring Veteran Status Codes

The steps to resolve the Veteran Status Code and the Veteran Status Description requirements are very similar to those for Ethnic Group. The domain values for Veteran Status Code and Veteran Status Description are shown in [Table 81](#).

Table 81. Domain Value for Veteran Status Code and Veteran Status Description

Veteran Status Code	Veteran Status Description
1	Special Disabled Veterans
2	Vietnam Era Veterans
3	Other Protected Veterans
4	Newly Separated Veterans

It is recommended that you do not change the pre-packaged domain values. Changing the pre-packaged domain values impacts on the Dashboards and the reporting metadata. You should limit yourselves to just 'mapping' the source supplied values against the packaged set of domain values. A sample set of domain value 'maps' is provided as a CSV lookup file. You should modify that file according to values that your source provides. An example of the CSV lookup file is shown in [Table 82](#).

Table 82. Sample Source-Supplied Veteran Status Code and Veteran Status Description

VETERAN_STATUS_CODE	W_VETERAN_STATUS_CODE	W_VETERAN_STATUS_DESC
VIETVET	2	Vietnam Era Veterans
VIETVETDIS	1	Special Disabled Veterans

NOTE: This example CSV lookup file specifies that the Veteran Status Code "VIETVETDIS" signifies the status 'Special Disabled Veteran'. However, you should use the Veteran Status Code used by your source system.

To modify Veteran Status code

- 1 Edit the file "domainValues_Employee_Veteran_Status_Code_ora11i.csv" in the \$pmserver\LkpFiles directory using in a text editor.
- 2 Modify the VETERAN_STATUS_CODE column (the source code) as required so that it maps correctly in the corresponding domain values in the table above.
- 3 Save the file.

NOTE: You might need to change the metric names in the reporting data to use more suitable names.

Configuring Veteran Status Descriptions

As with the Ethnic Group Description, make sure that the Veteran Status Description used with the source-specified code matches the description defined in the domain values. For example, in the sample source data in [Table 82](#), the source system uses the description Other Veterans instead of Other Protected Veterans for source-supplied code VET.

To modify Veteran Status description

- 1 Edit the file "domainValues_Employee_Veteran_Status_Code_ora11i.csv" in the \$pmserver\LkpFiles directory using in a text editor.
- 2 Modify the W_VETERAN_STATUS_DESC column (the domain value description) as required so that it maps correctly in the corresponding domain values in the table above.
- 3 Save the file.

Configuring Sex Codes

The steps for configuring Sex Codes and Sex Descriptions are similar to those for configuring Ethnic Groups. The domain values for Sex Code and Sex Description are shown in [Table 83](#).

Table 83. Domain Values for Sex Code and Sex Description

Sex Code	Sex Description
F	Female
M	Male
U	Unknown

It is recommended that you do not change the pre-packaged domain values. If you do so, there will be impacts on the Dashboards and the reporting metadata. You should limit yourselves to just 'mapping' the source supplied values against the packaged set of domain values. A sample set of domain value 'maps' is provided as a CSV lookup file – you should be modifying that file according to values that your source provides. An example of the CSV lookup is shown in [Table 80](#).

Table 84. Sample Source Values for Sex Code and Sex Description

SEX	W_SEX_MF_CODE	W_SEX_MF_DESC
F	F	Female
M	M	Male
	U	Unknown

NOTE: This example CSV lookup file specifies that the Sex Code "F" signifies the group 'Female'. However, you should use the Sex Code used by your source system.

To modify the Sex Code

- 1 Edit the file "domainValues_Employee_Sex_MF_ora11i.csv" in the \$pmserver\LkpFiles directory using in a text editor.
- 2 Modify the Sex column (the source code) as required so that it maps correctly in the corresponding domain values in the table above.
- 3 Save the file.

Configuring Sex Descriptions

As with the Ethnic Group Description, make sure that the Sex Description used with the source-specified code matches the description defined in the domain values. For example, the source system might use the description Unspecified instead of Unknown for source-supplied code NULL (in other words, not supplied).

To modify the Sex Descriptions

- 1** Edit the file “domainValues_Employee_Sex_MF_ora11i.csv” in the \$pmserver\LkpFiles directory using in a text editor.
- 2** Modify the W_SEX_MF_DESC column (the domain value description) as required so that it maps correctly in the corresponding domain values in the table above.
- 3** Save the file.

NOTE: You might need to change the metric names in the reporting data to use more suitable names.

Configuring the Jobs Dimension for U.S. Statutory Compliance

This task is a step in the [Process of Configuring Workforce Operations for Oracle 11i on page 364](#).

Within the Jobs dimension there are mandatory changes to the configuration information about two sets of columns, as follows:

- EEO Job Category Code (W_EEO_JOB_CAT_CODE)
- EEO Job Category Description (W_EEO_JOB_CAT_DESC)
- FLSA Status Code (W_FLSA_STATUS_CODE)
- FLSA Status Description (W_FLSA_STATUS_DESC)

Configuring EEO Job Category Codes

The domain values for EEO Job Category Code and EEO Job Category Description in Oracle Business Analytics Warehouse are shown in [Table 85](#).

Table 85. Domain Values for EEO Job Category Code and EEO Job Category Description

EEO Job Category Code	EEO Job Category Description
1	Officials and Managers
2	Professionals
3	Technicians
4	Sales Workers
5	Office and Clerical
6	Craft Workers
7	Operatives
8	Laborers
9	Service Workers

It is recommended that you do not change the pre-packaged domain values. If you do so, there will be impacts on the Dashboards and the reporting metadata. You should limit yourselves to just 'mapping' the source supplied values against the packaged set of domain values. A sample set of domain value 'maps' is provided as a CSV lookup file – you should be modifying that file according to values that your source provides. An example of the CSV lookup file is shown in [Table 86](#).

Table 86. Sample Source-Supplied EEO Job Category Code and EEO Job Category Description

EEO_JOB_CAT_CODE	W_EEO_JOB_CAT_CODE	W_EEO_JOB_CAT_DESC
1:EEO1CODE	1	Laborers (Unskilled)
2:EEO1CODE	2	Office-Clerical
3:EEO1CODE	3	Technicians
4:EEO1CODE	4	Sales Workers
5:EEO1CODE	5	Professionals
6:EEO1CODE	6	Craft Workers
7:EEO1CODE	7	Operatives (Semi-Skilled)
8:EEO1CODE	8	Officials and Managers
9:EEO1CODE	9	Service Workers

NOTE: This example CSV lookup file specifies that the EEO Job Category Code "1:EEO1CODE" signifies the group 'Laborers (Unskilled)'. However, you should use the EEO Job Category Code used by your source system.

To modify EEO Job Category code

- 1 Edit the file "domainValues_Job_Eeo_Cat_Code_ora11i.csv" in the \$pmserver\LkpFiles directory using in a text editor.
- 2 Modify the EEO_JOB_CAT_CODE column (the source code) as required so that it maps correctly in the corresponding domain values in the table above.
- 3 Save the file.

Configuring EEO Job Category Descriptions

As with the Ethnic Group Description and Veteran Status Description, make sure that the EEO Job Category Description used with the source-specified code matches the description defined in the domain values. For example, in the sample source data in [Table 86](#), the source system uses the description Office-Clerical instead of Office and Clerical for source-supplied Code 2.

To modify the EEO Job Category description

- 1 Edit the file "domainValues_Job_Eeo_Cat_Code_ora11i.csv" in the \$pmserver\LkpFiles directory using in a text editor.

- 2 Modify the W_EEO_JOB_CAT_DESC column (the domain value description) as required so that it maps correctly in the corresponding domain values in the table above.
- 3 Save the file.

NOTE: You might need to change the metric names in the reporting data to use more suitable names.

Configuring Address Types for Workforce Profile

This task is a step in the [Process of Configuring Workforce Operations for Oracle 11i on page 364](#).

There are three address fields in the Employee dimension table:

- Permanent address
- Mail address
- Work address

For each of these, we use only the primary ones. In addition, the following logic is used to determine the various types of addresses:

- Permanent: Address Type = 'H' (This is also the home address, in other words)
- Mail: Address Type = 'M'. If this is not available, use Permanent address (which can be the best alternate for mailing address).
- Work: Address Type = 'Default'. If this is not available, use Permanent address (which can be the best alternate for mailing address).

You can modify this logic if required. For example, if your system tracks work address with a special address type 'W', then you should be able to modify the existing logic. Or, if you do not want to assume that the mail address (M) is same as the home address (H), you may want to remove the null evaluation check there.

To configure Address Type

- 1 In PowerCenter Designer, open the SDE_ORA1158_Adapter folder.
- 2 In Mapplet Designer, open the mplt_SA_ORA_EmployeeDimension mapplet.

- 3** Locate the expression transformation Exp_SA_Employees_Validate and scroll down the ports to get to the variables “HOME_ADDRESS_ID_VAR”, “MAIL_ADDRESS_ID_VAR” and “DEFAULT_ADDRESS_ID_VAR”.

For example, if you have a specific address type for work addresses, (for example 'W'), you would modify the expression for the variable port DEFAULT_ADDRESS_ID_VAR as follows:

From: :LKP.Lkp_Address_Id(INP_PERSON_ID, INP_EFFECTIVE_END_DATE, 'Y', 'Default',
INP_DATASOURCE_NUM_ID)

To: :LKP.Lkp_Address_Id(INP_PERSON_ID, INP_EFFECTIVE_END_DATE, 'Y', 'W',
INP_DATASOURCE_NUM_ID)

The output Address ID values are the next three ports “PERM_ADDRESS_ID”, “MAIL_ADDRESS_ID” and “WORK_ADDRESS_ID”. If you do not want to assume that the mail address (M) is same as the home address (H) in the event of mail address not being available, then you would modify the logic in the outgoing port “MAIL_ADDRESS_ID” as follows:

From: IIF(ISNULL(MAIL_ADDRESS_ID_VAR), HOME_ADDRESS_ID_VAR,
MAIL_ADDRESS_ID_VAR)

To: MAIL_ADDRESS_ID_VAR

- 4** Validate and save changes to the repository. If you are using the version controlling for the Informatica repository, you will have to check in your changes as well.

Configuring Phone Types for Workforce Profile

This task is a step in the [Process of Configuring Workforce Operations for Oracle 11i on page 364](#).

There are four phone related fields in the Employee dimension table, as follows:

- Fax
- Work phone
- Pager
- Mobile phone

The following logic is used to arrive at the various types of addresses:

- Fax: Phone Type = 'WF' (Work Fax)
- Work Phone: Phone Type = 'W1' (First work phone, if there are more than one)
- Pager: Phone Type = 'P' (Pager)
- Mobile: Phone Type = 'M' (Mobile)

You can modify this logic if required. For example, if your system tracks the primary work phone with a special phone type 'WP', instead of W1, then you should be able to modify the existing logic. The same applies for other phone types as well.

To configure Phone Type

- 1 In PowerCenter Designer, open the SDE_ORA1158_Adapter folder.
- 2 In 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.

Process of Configuring Workforce Payroll for Oracle 11i

This section contains configuration instructions for Payroll specific to Oracle 11i.

To configure Workforce Payroll for Oracle 11i, perform the following tasks:

- [Modifying the Workforce Payroll Filters on page 374](#)
- [Improving ETL Performance for Workforce Payroll on page 376](#)

Modifying the Workforce Payroll Filters

This task is a step in the [Process of Configuring Workforce Payroll for Oracle 11i on page 374](#).

Oracle Business Intelligence Applications is preconfigured to extract the input value name of Pay Value. Oracle Business Analytics Warehouse does not extract classification elements such as Information, Balance, or Employer. To modify the Payroll filter perform the following procedure.

NOTE: If you change the Payroll filter, you need to also change the Oracle BI metadata, so your reports are run correctly.

To modify the Payroll filters

- 1 Open the parameterfile0LTP.txt file using Notepad in the installation directory in \OracleBI\DAC\Informatica\parameters\input.

- 2 Locate the sections [SDE_ORA_PayrollFact] and [SDE_ORA_PayrollFact_Full].

For example, the file might contain the following sections:

```
[SDE_ORA_PayrollFact]
$$DATASOURCE_NUM_ID=2
$$TENANT_ID=DEFAULT
$DBConnection_OLAP=PARAM_OLAP
$DBConnection_OLTP=PARAM_OLTP_ORA11i
mplt_BC_ORA_PayrollFact.$$hint1=/* + RULE*/
mplt_BC_ORA_PayrollFact.$$INITIAL_EXTRACT_DATE=01/01/1970
mplt_BC_ORA_PayrollFact.$$INITIAL_EXTRACT_DATE1=01/01/1970
mplt_BC_ORA_PayrollFact.$$LAST_EXTRACT_DATE=01/01/1970
mplt_BC_ORA_PayrollFact.$$RUN_RESULT_FILTER1= AND PAY_INPUT_VALUES_F.NAME='Pay
Value'
mplt_BC_ORA_PayrollFact.$$RUN_RESULT_FILTER2= AND CLASSIFICATION_NAME NOT
LIKE '% Information%'
mplt_BC_ORA_PayrollFact.$$RUN_RESULT_FILTER3= AND CLASSIFICATION_NAME NOT
LIKE '% Employer%'
mplt_BC_ORA_PayrollFact.$$RUN_RESULT_FILTER4= AND CLASSIFICATION_NAME NOT
LIKE '% Balance%'

[SDE_ORA_PayrollFact_Full]
$$DATASOURCE_NUM_ID=2
$$TENANT_ID=DEFAULT
$DBConnection_OLAP=PARAM_OLAP
$DBConnection_OLTP=PARAM_OLTP_ORA11i
mplt_BC_ORA_PayrollFact.$$hint1=/* + RULE*/
mplt_BC_ORA_PayrollFact.$$INITIAL_EXTRACT_DATE=01/01/1970
mplt_BC_ORA_PayrollFact.$$INITIAL_EXTRACT_DATE1=01/01/1970
mplt_BC_ORA_PayrollFact.$$LAST_EXTRACT_DATE=01/01/1970
mplt_BC_ORA_PayrollFact.$$RUN_RESULT_FILTER1= AND PAY_INPUT_VALUES_F.NAME='Pay
Value'
mplt_BC_ORA_PayrollFact.$$RUN_RESULT_FILTER2= AND CLASSIFICATION_NAME NOT
LIKE '% Information%'
mplt_BC_ORA_PayrollFact.$$RUN_RESULT_FILTER3= AND CLASSIFICATION_NAME NOT
LIKE '% Employer%'
```

```
mplt_BC_ORA_PayrollFact.$$RUN_RESULT_FILTER4= AND CLASSIFICATION_NAME NOT  
LIKE '%Balance%'
```

- 3 Change the following sections, as required (highlighted in bold above):

```
PAY_INPUT_VALUES_F.NAME='Pay Value'
```

```
CLASSIFICATION_NAME NOT LIKE '%Information%'
```

```
CLASSIFICATION_NAME NOT LIKE '%Employer%'
```

```
CLASSIFICATION_NAME NOT LIKE '%Balance%'
```

- 4 Save and close the file.

Improving ETL Performance for Workforce Payroll

This task is a step in the [Process of Configuring Workforce Payroll for Oracle 11i on page 374](#).

Oracle Business Intelligence Applications is preconfigured to use the Rule Base Optimization to improve the performance of the Payroll extract mapping. You can change this by using the following procedure.

To improve ETL performance for Payroll

- 1 Open the parameterfile0LTP.txt file using Notepad in the installation directory in \OracleBI\DAC\Informatica\parameters\input.

- 2 Locate the sections [SDE_ORA_PayrollFact] and [SDE_ORA_PayrollFact_Full].

For example, the file might contain the following sections:

```
[SDE_ORA_PayrollFact]
$$DATASOURCE_NUM_ID=2
$$TENANT_ID=DEFAULT
$DBConnection_OLAP=PARAM_OLAP
$DBConnection_OLTP=PARAM_OLTP_ORA11I
mplt_BC_ORA_PayrollFact.$$hint1= / * + RULE* /
...

[SDE_ORA_PayrollFact_Full]
$$DATASOURCE_NUM_ID=2
$$TENANT_ID=DEFAULT
$DBConnection_OLAP=PARAM_OLAP
$DBConnection_OLTP=PARAM_OLTP_ORA11I
mplt_BC_ORA_PayrollFact.$$hint1= / * + RULE* /
...
```

- 3 Change the value of the database hint, as required (highlighted in bold above).
- 4 Save and close the file.

About Aggregating the Payroll Table for Oracle Workforce 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 Workforce Analytics minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. Oracle Business Analytics 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 Workforce 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”.

Aggregating the Payroll Table for Oracle Workforce Analytics

To load the Payroll aggregate table (W_PAYROLL_A), you need to configure the post-load processing parameter file, parameterfileDW.csv, and run the initial workflow and then the incremental workflow.

To configure the parameterfileDW.txt parameters file

- 1 Open the parameterfileDW.txt file using Microsoft WordPad or Notepad in the \OracleBI\DAC\Informatica\parameters\input.
- 2 Locate the sections [PLP_PayrollAggregate_Extract] and [PLP_PayrollAggregate_Load_Full].

For example:

```
[PLP_PayrollAggregate_Extract]
$$ETL_PROC_WID=1
$$GRAIN=MONTH
$DBConnection_OLAP=PARAM_OLAP
[PLP_PayrollAggregate_Load_Full]
$$GRAIN='MONTH'
$$Hint_Tera_Post_Cast= as decimal(18,3))
$$Hint_Tera_Pre_Cast= cast (
$DBConnection_OLAP=PARAM_OLAP
MPLT_GET_ETL_PROC_WID.$$ETL_PROC_WID=1
```

- 3 Change the value of the aggregation time grains, as required (highlighted in bold above).

NOTE: You need to use single quotes for the GRAIN parameter for the section "PLP_PayrollAggregate_Load_Full".

- 4 Save and close the file.

Domain Values and CSV Worksheet Files for Oracle Workforce Analytics

Table 87 lists the CSV worksheet files and the domain values for Oracle Workforce Analytics in the \$pmsserver\LkpFiles folder.

For more information on configuring domain values with CSV worksheet files, see [About Domain Values on page 238](#) and [Configuring the Domain Value Set with CSV Worksheet Files on page 243](#).

Table 87. Domain Values and CSV Worksheet Files for Oracle Human Resources Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_Employee_Ethnic_Group_Code_ora11i.csv	W_EMPLOYEE_D.W_ETHNIC_GRP_CODE	Lists the Ethnic codes and their corresponding domain values of "Ethnic Group Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Sex_MF_ora11i.csv	W_EMPLOYEE_D.W_SEX_MF_CODE	Lists the Sex codes and their corresponding domain values of "Sex Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Veteran_Status_Code_ora11i.csv	W_EMPLOYEE_D.W_VETERAN_STATUS_CODE	Lists the Veteran codes and their corresponding domain values of "Veteran Status Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employment_Cat_ora11i.csv	W_EMPLOYMENT_D.W_EMPLOYEE_CAT_CODE	Lists the User Person Types and their corresponding domain values of "Employment Category Code" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full

Table 87. Domain Values and CSV Worksheet Files for Oracle Human Resources Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_Employment_Exempt_Flg_ora11i.csv	W_EMPLOYMENT_D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of "Exempt Flag" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Full_Time_Flg_ora11i.csv	W_EMPLOYMENT_D.W_FULL_TIME_FLG	Lists the Employment Categories and their corresponding domain values of "Full Time Flag" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Status_ora11i.csv	W_EMPLOYMENT_D.W_EMPLOYMENT_STAT_CODE	Lists the Per System Statuses and their corresponding domain values of "Employment Status" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_EventTypes_ora11i.csv	W_EVENT_TYPE_D.W_EVENT_CLASS, W_EVENT_TYPE_D.W_EVENT_GRP_CODE, W_EVENT_TYPE_D.W_EVENT_REASON_CODE, W_EVENT_SUBG_CODE	Lists the Event Types, Event Codes and Meanings and their corresponding domain values of "Event Group", "Event Sub-Group" and "Event Reason" for the Oracle 11i Application.	SDE_ORA_EventTypeDimension_AbsenceAttendance, SDE_ORA_EventTypeDimension_AbsenceAttendance_Full, SDE_ORA_EventTypeDimension_AdditionalEvents_FromFile, SDE_ORA_EventTypeDimension_OtherHREvents, SDE_ORA_EventTypeDimension_OtherHREvents_Full
domainValues_HRPosition_Active_Pos_Flg_ora11i.csv	W_HR_POSITION_D.W_ACTIVE_POSITION_FLAG	Lists the Position Statuses and their corresponding domain values of "Active Position Flag" for the Oracle 11i Application.	SDE_ORA_HRPositionDimension, SDE_ORA_HRPositionDimension

Table 87. Domain Values and CSV Worksheet Files for Oracle Human Resources Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_HRPosition_Exempt_Flg_ora11i.csv	W_HR_POSITION_D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of "Exempt Flag" for the Oracle 11i Application.	SDE_ORA_HRPosition Dimension, SDE_ORA_HRPosition Dimension
domainValues_Job_Eeo_Cat_Code_ora11i.csv	W_JOB_D.W_EEO_JOB_CAT_CODE	Lists the EEO Job Categories and their corresponding domain values of "EEO Job Category" for the Oracle 11i Application.	SDE_ORA_JobDimension, SDE_ORA_JobDimension_Full
domainValues_Job_Flsa_Stat_Code_ora11i.csv	W_JOB_D.W_FLSA_STAT_CODE	Lists the FLSA Statuses and their corresponding domain values of "FLSA Status Code" for the Oracle 11i Application.	SDE_ORA_JobDimension, SDE_ORA_JobDimension_Full
domainValues_Pay_Type_Grp_Code_ora11i.csv	W_PAY_TYPE_D.W_PAY_TYPE_GRP_CODE	Lists the Classification Names, Element Names and their corresponding domain values of "Pay Type Group Code" for the Oracle 11i Application.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full
domainValues_Pay_Type_Flg_ora11i.csv	W_PAY_TYPE_D.W_PAY_TYPE_FLG	Lists the Costing Debit or Credit values and their corresponding domain values of "Pay type Flag" for the Oracle 11i Application.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full
domainValues_Employee_Ethnic_Group_Code_ora11i.csv	W_EMPLOYEE_D.W_ETHNIC_GRP_CODE	Lists the Ethnic codes and their corresponding domain values of "Ethnic Group Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full

Table 87. Domain Values and CSV Worksheet Files for Oracle Human Resources Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_Employee_Sex_MF_ora11i.csv	W_EMPLOYEE_D.W_SEX_MF_CODE	Lists the Sex codes and their corresponding domain values of "Sex Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Veteran_Status_Code_ora11i.csv	W_EMPLOYEE_D.W_VETERAN_STATUS_CODE	Lists the Veteran codes and their corresponding domain values of "Veteran Status Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employment_Cat_ora11i.csv	W_EMPLOYMENT_D.W_EMPLOYEE_CAT_CODE	Lists the User Person Types and their corresponding domain values of "Employment Category Code" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Exempt_Flg_ora11i.csv	W_EMPLOYMENT_D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of "Exempt Flag" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Full_Time_Flg_ora11i.csv	W_EMPLOYMENT_D.W_FULL_TIME_FLG	Lists the Employment Categories and their corresponding domain values of "Full Time Flag" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Status_ora11i.csv	W_EMPLOYMENT_D.W_EMPLOYMENT_STAT_CODE	Lists the Per System Statuses and their corresponding domain values of "Employment Status" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full

Table 87. Domain Values and CSV Worksheet Files for Oracle Human Resources Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_EventTypes_ora11i.csv	W_EVENT_TYPE_D.W_EVENT_CLASS, W_EVENT_TYPE_D.W_EVENT_GRP_CODE, W_EVENT_TYPE_D.W_EVENT_REASON_CODE, W_EVENT_SUBG_CODE	Lists the Event Types, Event Codes and Meanings and their corresponding domain values of "Event Group", "Event Sub-Group" and "Event Reason" for the Oracle 11i Application.	SDE_ORA_EventTypeDimension_AbsenceAttendance, SDE_ORA_EventTypeDimension_AbsenceAttendance_Full, SDE_ORA_EventTypeDimension_AdditionalEvents_FromFile, SDE_ORA_EventTypeDimension_OtherHREvents, SDE_ORA_EventTypeDimension_OtherHREvents_Full
domainValues_HRPosition_Active_Pos_Flg_ora11i.csv	W_HR_POSITION_D.W_ACTIVE_POSITION_FLAG	Lists the Position Statuses and their corresponding domain values of "Active Position Flag" for the Oracle 11i Application.	SDE_ORA_HRPositionDimension, SDE_ORA_HRPositionDimension
domainValues_HRPosition_Exempt_Flg_ora11i.csv	W_HR_POSITION_D.W_EXEMPT_FLAG	Lists the FLSA Statuses and their corresponding domain values of "Exempt Flag" for the Oracle 11i Application.	SDE_ORA_HRPositionDimension, SDE_ORA_HRPositionDimension
domainValues_Job_EEO_Cat_Code_ora11i.csv	W_JOB_D.W_EEO_JOB_CAT_CODE	Lists the EEO Job Categories and their corresponding domain values of "EEO Job Category" for the Oracle 11i Application.	SDE_ORA_JobDimension, SDE_ORA_JobDimension_Full
domainValues_Job_FLSA_Stat_Code_ora11i.csv	W_JOB_D.W_FLSA_STAT_CODE	Lists the FLSA Statuses and their corresponding domain values of "FLSA Status Code" for the Oracle 11i Application.	SDE_ORA_JobDimension, SDE_ORA_JobDimension_Full

Table 87. Domain Values and CSV Worksheet Files for Oracle Human Resources Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_Pay_Type_Grp_Code_ora11i.csv	W_PAY_TYPE_D.W_PAY_TYPE_GRP_CODE	Lists the Classification Names, Element Names and their corresponding domain values of "Pay Type Group Code" for the Oracle 11i Application.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full
domainValues_Pay_Type_Flg_ora11i.csv	W_PAY_TYPE_D.W_PAY_TYPE_FLG	Lists the Costing Debit or Credit values and their corresponding domain values of "Pay type Flag" for the Oracle 11i Application.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full
domainValues_Employee_Ethnic_Group_Code_ora11i.csv	W_EMPLOYEE_D.W_ETHNIC_GRP_CODE	Lists the Ethnic codes and their corresponding domain values of "Ethnic Group Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Sex_MF_ora11i.csv	W_EMPLOYEE_D.W_SEX_MF_CODE	Lists the Sex codes and their corresponding domain values of "Sex Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full

Configuring Domain Values and CSV Worksheet Files for Oracle Workforce Analytics

You configure the CSV files in Oracle Workforce Analytics by mapping domain values to columns. The CSV worksheet files for Oracle Workforce Analytics are in the \$pmserver\LkpFiles folder.

For more information on configuring domain values with CSV worksheet files, see [About Domain Values on page 238](#) and [Configuring the Domain Value Set with CSV Worksheet Files on page 243](#).

This section contains the following tasks:

- To configure the domainValues_Employee_Ethnic_Group_Code_ora11i.csv on page 385
- To configure the domainValues_Employee_Sex_MF_ora11i.csv on page 385
- To configure the domainValues_Employee_Veteran_Status_Code_ora11i.csv on page 386

- To configure the domainValues_Employment_Cat_ora11i.csv on page 386
- To configure the domainValues_Employment_Exempt_Flg_ora11i.csv on page 387
- To configure the domainValues_Employment_Full_Time_Flg_ora11i.csv on page 387
- To configure the domainValues_Employment_Status_ora11i.csv on page 388
- To configure the domainValues_EventTypes_ora11i.csv on page 388
- To configure the domainValues_HRPosition_Active_Pos_Flg_ora11i.csv on page 389
- To configure the domainValues_HRPosition_Exempt_Flg_ora11i.csv on page 389
- To configure the domainValues_Job_Eeo_Cat_Code_ora11i.csv on page 390
- To configure the domainValues_Job_Flsa_Stat_Code_ora11i.csv on page 390
- To configure the domainValues_Pay_Type_Grp_Code_ora11i.csv on page 391
- To configure the domainValues_Pay_Type_Flg_ora11i.csv on page 391

To configure the domainValues_Employee_Ethnic_Group_Code_ora11i.csv

- 1 Identify the Ethnic Group Codes in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT PER_INFORMATION1 FROM PER_ALL_PEOPLE_F  
  
WHERE PER_INFORMATION1 in  
( '1','2','3','4','5','6','7','8','9','10','11','12','BA','BC','BO','C','I','O','P','W')  
  
ORDER BY 1
```

- 2 Open the domainValues_Employee_Ethnic_Group_Code_ora11i.csv file using Microsoft WordPad or Notepad 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.

For more information on Ethnic Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

- 5 Save and close the file.

To configure the domainValues_Employee_Sex_MF_ora11i.csv

- 1 1)Identify the Sex Codes in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT SEX FROM PER_ALL_PEOPLE_F ORDER BY 1
```
- 2 Open the domainValues_Employee_Sex_MF_ora11i.csv file using Microsoft WordPad or Notepad 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.
For more information on Sex Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.
- 5 Save and close the file.

To configure the domainValues_Employee_Veteran_Status_Code_ora11i.csv

- 1 Identify the Veteran Status Codes in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT PER_INFORMATION5 FROM PER_ALL_PEOPLE_F  
WHERE PER_INFORMATION5 in ('NOTVET', 'OTEDV', 'VET', 'VETDIS', 'VIETVET', 'VIETVETDIS')  
ORDER BY 1
```
- 2 Open the domainValues_Employee_Veteran_Status_Code_ora11i.csv file using Microsoft WordPad or Notepad 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.
For more information on Veteran Status Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.
- 5 Save and close the file.

To configure the domainValues_Employment_Cat_ora11i.csv

- 1 Identify the User Person Types in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT SYSTEM_PERSON_TYPE, USER_PERSON_TYPE FROM  
PER_PERSON_TYPES  
WHERE SYSTEM_PERSON_TYPE IN  
( 'EMP', 'OTHER', 'EMP_APL', 'EX_EMP', 'EX_EMP_APL', 'RETIREE', 'PRTN' )  
ORDER BY 1, 2
```


NOTE: If you have modified the Payroll filter, you need to also modify the SQL. For more information on modifying Payroll filters, see [Modifying the Workforce Payroll Filters on page 374](#).
- 2 Open the domainValues_Employment_Cat_ora11i.csv file using Microsoft WordPad or Notepad in the \$pmserver\lkpfiles folder.

- 3 Copy the User Person Types to the USER_PERSON_TYPE column in the file. The data must be copied starting from the 6th line.
- 4 Map each User Person Type to one domain value. System Person Types are also extracted with User Person Type to help you map the domain values. Do not copy the System Person types in the CSV file.

For more information on Employment Category domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

- 5 Save and close the file.

To configure the domainValues_Employment_Exempt_Flg_ora11i.csv

- 1 Identify the FLSA Statuses in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS  
ORDER BY 1
```

- 2 Open the domainValues_Employment_Exempt_Flg_ora11i.csv file using Microsoft WordPad or Notepad 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.

For more information on Exempt Flag domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

- 5 Save and close the file.

To configure the domainValues_Employment_Full_Time_Flg_ora11i.csv

- 1 Identify the Employment Categories in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT EMPLOYMENT_CATEGORY FROM PER_ALL_ASSIGNMENTS_F ORDER BY 1
```

- 2 Open the domainValues_Employment_Full_Time_Flg_ora11i.csv file using Microsoft WordPad or Notepad 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.

For more information on Full Time Flag domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

- 5 Save and close the file.

To configure the domainValues_Employment_Status_ora11i.csv

- 1 Identify the Per System Statuses in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT PER_SYSTEM_STATUS FROM PER_ASSIGNMENT_STATUS_TYPES  
ORDER BY 1
```

- 2 Open the domainValues_Employment_Status_ora11i.csv file using Microsoft WordPad or Notepad 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.

For more information on Employment Status domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

- 5 Save and close the file.

To configure the domainValues_EventTypes_ora11i.csv

- 1 Identify the Event Types in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT LOOKUP_TYPE, LOOKUP_CODE, MEANING  
FROM FND_LOOKUP_VALUES  
WHERE LOOKUP_TYPE IN  
( 'EMP_ASSIGN_REASON',  
  'LEAV_REAS',  
  'PROPOSAL_REASON')  
ORDER BY 1, 2, 3
```

- 2 Open the domainValues_EventTypes_ora11i.csv file using Microsoft WordPad or Notepad 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.

For more information on Event Type domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

- 5 Save and close the file.

To configure the domainValues_HRPosition_Active_Pos_Flg_ora11i.csv

- 1 Identify the Position Statuses in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT STATUS FROM HR_ALL_POSITIONS_F  
  
ORDER BY 1
```

- 2 Open the domainValues_HRPosition_Active_Pos_Flg_ora11i.csv file using Microsoft WordPad or Notepad 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.

For more information on Active Position Flag domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

- 5 Save and close the file.

To configure the domainValues_HRPosition_Exempt_Flg_ora11i.csv

- 1 Identify the FLSA Statuses in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS  
  
ORDER BY 1
```

- 2 Open the domainValues_HRPosition_Exempt_Flg_ora11i.csv file using Microsoft WordPad or Notepad 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.

For more information on Exempt Flag domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

- 5 Save and close the file.

To configure the domainValues_Job_Eeo_Cat_Code_ora11i.csv

- 1 Identify the EEO Job Categories in your Orace11i source system by using the following SQL:

```
SELECT DISTINCT JOB_INFORMATION1 FROM PER_JOBS  
  
ORDER BY 1
```

- 2 Open the domainValues_Job_Eeo_Cat_Code_ora11i.csv file using Microsoft WordPad or Notepad 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.

For more information on Equal Employment Opportunity (EEO) domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

- 5 Save and close the file.

To configure the domainValues_Job_Flsa_Stat_Code_ora11i.csv

- 1 Identify the FLSA Statuses in your Orace11i source system by using the following SQL:

```
SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS  
  
ORDER BY 1
```

- 2 Open the domainValues_Job_Flsa_Stat_Code_ora11i.csv file using Microsoft WordPad or Notepad 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.

For more information on Fair Labor Standards Act (FLSA) Status Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

- 5 Save and close the file.

To configure the domainValues_Pay_Type_Grp_Code_ora11i.csv

- 1 Identify the Pay Elements in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT CLASSIFICATION_NAME, ELEMENT_NAME
FROM
PAY_ELEMENT_TYPES_F,
PAY_ELEMENT_CLASSIFICATIONS
WHERE
PAY_ELEMENT_CLASSIFICATIONS.CLASSIFICATION_ID =
PAY_ELEMENT_TYPES_F.CLASSIFICATION_ID AND
CLASSIFICATION_NAME NOT LIKE '%Information%' AND
CLASSIFICATION_NAME NOT LIKE '%Employer%' AND
CLASSIFICATION_NAME NOT LIKE '%Balance%'
ORDER BY 1, 2
```

- 2 Open the domainValues_Pay_Type_Grp_Code_ora11i.csv file using Microsoft WordPad or Notepad 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.
- 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".

For more information on Pay Type Group Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

- 5 Save and close the file.

To configure the domainValues_Pay_Type_Flg_ora11i.csv

- 1 Identify the Costing (Debit or Credit) in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT COSTING_DEBIT_OR_CREDIT FROM PAY_ELEMENT_CLASSIFICATIONS
ORDER BY 1
```

- 2 Open the domainValues_Pay_Type_Flg_ora11i.csv file using Microsoft WordPad or Notepad 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.

For more information on Pay Type Flag domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

- 5 Save and close the file.

17

Using Oracle Business Analytics Warehouse Exception Reports With Siebel CRM Sources

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:

- [Understanding Oracle Business Analytics Warehouse Exceptions on page 393](#)
- [Executing Oracle Business Analytics Warehouse Exception Reports on page 396](#)

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.

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 88](#).

Table 88. List of Values Exceptions (Example)

Type	Name	Low	High
MY_TYPE	< 100	10	100
MY_TYPE	80 – 200	80	200

Table 88. List of Values Exceptions (Example)

Type	Name	Low	High
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

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 89](#).

Table 89. 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.

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.

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 transactional database. 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 transactional database. If any other types have been defined, they are not handled without some customization.

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 transactional database. If the depth of hierarchies extends beyond this number, results become inconsistent or incomplete.

Circular Hierarchy Exceptions

Circular Hierarchies arise when the parent-child relationship has circular references. See [Table 90](#) for an example.

Table 90. 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 91](#) for an example.

Table 91. Circular Hierarchy Example 2

Child	Parent
A1	A2
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.

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 transactional database, repeat these actions until the exception report is empty:

- In the DAC, run the Exception Reports execution plan.
- In Oracle's Siebel application that you are using, navigate to Analytics Administration > 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 [Cleansing Data](#).

Cleansing Data

Use a combination of the Exception Reports and the Diagnostic views to assess changes that need to be made external to the transactional database, and changes to the transactional database directly.

The ETL Exception Reports list contains one record for each exception flagged in the transactional database. 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.

Using the List of Values View

Use the List of Values view, shown in [Figure 33](#), to visually compare how the list of values data extracted from the transactional database 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.

ETL History List of Values Exchange Rates Cost Lists						
Query Clear Cache 1 - 7 of 7+						
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

List Of Values (Data Warehouse)						
Query 1 - 7 of 7+						
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	

Figure 33. List of Values List

The top List of Values list shows values from the transactional database and the bottom List of Values (Data Warehouse) list shows the data that is to be used in ETL process. You can edit the transactional database 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 [Configuring Email Recipients in the DAC Client on page 103](#)).

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 transactional database.
- 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.

Using the Cost List View

Use the Cost List view to display the cost lists from the transactional database 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.

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.

Additional Exceptions

The above mentioned exceptions are not an exhaustive list of all possible exceptions. Other exceptions are:

- The Analysis start and end date in the DAC Source System Parameters must span the entire period of time during which the transactions have occurred. For example, you may want to choose an early and late date range to cover the entire time period you are analyzing. These dates in the DAC Source System Parameters are crucial for the building of Day Dimension, flattening of Exchange Rates, Cost Lists, and KPI (Key Performance Indicator fact) calculations.

- The DAC Source System Parameters — ETL Date Format, ETL Analysis Start, ETL Analysis End parameters, and the List of Values — ETL_UNSPEC_DATE must be defined in the same data format. If one is changed, the others must be changed accordingly.
- List of Values must be defined appropriately. If there is no appropriate entry in List of Values, the strings that depend on List of Values in the transactional database 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.

18 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:

- [Oracle Business Analytics Warehouse Installation and Initial Configuration Issues on page 402](#)
- [Informatica and Loading Issues on page 403](#)
- [Unicode Issues with Informatica and the Oracle Business Analytics Warehouse on page 408](#)
- [UNIX Installation Issues with the Oracle Business Analytics Warehouse on page 409](#)
- [Oracle Business Analytics Warehouse Upgrade Issues on page 413](#)
- [Oracle BI Interactive Dashboard Issues on page 415](#)

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)

Oracle Business Analytics Warehouse Installation and Initial Configuration Issues

Table 92 provides information about problems and solutions related to the installation and initial configuration of the Oracle Business Analytics Warehouse.

Table 92. 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 "IBM DB2 UDB-Specific Database Guidelines for Oracle Business Analytics Warehouse" on page 32.
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 transactional database from Informatica.	Make sure that the server running the Informatica Server software has an ODBC connection to the transactional database 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 <i>Siebel eBusiness Third-Party Bookshelf</i> 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 Changing the Default Currency in Analytics Applications on page 425.

Informatica and Loading Issues

Table 93 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 93. Informatica and Loading Issues

Symptom / Error Message	Probable Cause / Solution
Double-clicking the workflow yields a Workflow Manager error message: "The system cannot find the file specified."	The session log files are not set up properly. You also may need to change the text editor.
Using Oracle, some mappings hang while running when performance statistics are switched on.	When running some Informatica mappings for loading the Oracle Business Analytics Warehouse, turning on the Performance Statistics can cause the mapping to hang. The only workaround is to increase the values of the LMSharedMemory and MaxSessions variables in Informatica. The risk of increasing the LMSharedMemory too much is that it may start to have a serious effect on overall performance of the machine that the Informatica server is running on.
When you execute a workflow on the Informatica Workflow Manager, Informatica returns the following error message: "Request to start workflow (workflow name) on server (server name) not successful."	This can happen due to a server time-out property that is usually set to 20 or 40 seconds. When you try to run a large workflow, every session in that workflow is fetched into the server's memory. If this takes longer than the server time-out property, the server returns a message that the workflow was unable to run. However, the workflow is running, but the server just needs time to complete fetching the sessions into memory. Double-click the workflow to view the log file details.
When running Full_Extract_Siebel_DW or Refresh_Extract_Siebel_DW, Informatica returns errors similar to: TE_7007 Transformation Evaluation Error; current row skipped... TE_7007 [<<Transformation Error>> [to_date]: Date function error to_date('19010101', 'DD-MON-YYYY')]	Incorrect date entry in the SME Date Format field in the System Preferences. The format is YYYYMMDD.

Table 93. Informatica and Loading Issues

Symptom / Error Message	Probable Cause / Solution
When running Full_Load_Siebel_DW, Informatica returns errors similar to: CMN_1014 Error creating semaphore... TM_6006 Error initializing DTM for session... TM_6006 [s_CR18a1. Load W_PROG_DM_TMP - Program Records]	Insufficient semaphores allocated. Allocate more semaphores on the Informatica Server. The change becomes effective when you reboot.
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 93. 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: <pre> DROP INDEX OPB_OBJECT_LOCKS.OPB_OBJ_LOCKS_IDX DROP INDEX OPB_OBJECT_LOCKS.OPB_OBJ_LOCKS_IDX2 DROP INDEX OPB_OBJECT_LOCKS.OPB_OBJ_LOCKS_IDX3 </pre> 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 Under Services > Informatica> Logon, make sure the login (domain\username) and password are correct. 2 Under Control Panel > Mail (it may also be called Mail and Fax or Exchange) > Services > Show Profiles, make sure the mail profile is correct. 3 Under Programs> Informatica Server > Informatica Server Setup> Miscellaneous, 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.

Table 93. Informatica and Loading Issues

Symptom / Error Message	Probable Cause / Solution
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.
Oracle 8i sessions running in bulk mode fail and return an error message similar to: <pre>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]</pre>	This problem is an Oracle 8i defect. It has been resolved in Oracle 9i. The workaround is to run the session in Normal mode. To do so, in Workflow Manager navigate to the Targets window, and change the Target Load type to Normal.
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: <pre>SEVERE: [IBM][CLI Driver][DB2] SQL0191N Error occurred because of a fragmented MBCS character. SQLSTATE=22504 103 SEVERE Tue May 11 21:37:29 CDT 2004 [IBM][CLI Driver][DB2] SQL0191N Error occurred because of a fragmented MBCS character. SQLSTATE=22504</pre>	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: - 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. - Make a copy of the file and rename it to IMB05348.ucs.

Table 93. Informatica and Loading Issues

Symptom / Error Message	Probable Cause / Solution
When an ETL process is running and tasks fail, Informatica returns an error similar to the following Error while running Workflow Description: ERROR : TM_6292 : (3040 4868) Session task instance REP_12400 [Repository Error ([REP_51055] Repository agent connection failed. [System Error (errno = 121): The semaphore timeout period has expired. . (Cannot read message. Read 5824 bytes.)]]]	This issue is related to the network. The workaround is to increase the Timeout parameter values in the Informatica Repository Server Administration Console. 1 In the left pane of the Repository Server Administration Console window, right click your repository and select Edit Connection. 2 In the Network tab, enter 9 as the value for the parameters MessageReceiveTimeout and MessageSendTimeout. 3 Stop and start the Informatica Repository Server. 4 Start the Informatica Server.
[IBM][CLI Driver] CLI0102E Invalid conversion. SQLSTATE=07006 sqlstate = 07006 [IBM][CLI Driver][DB2/AIX64] SQL0303N A value cannot be assigned to a host variable in the SELECT, VALUES, or FETCH statement because the data types are not compatible. SQLSTATE=42806 sqlstate = 42806 This issue comes up when you use 'Unicode' data movement in Informatica.	The target table that the mapping is inserting into is a non-unicode table. Make sure the table is created as an Unicode table.

Unicode Issues with Informatica and the Oracle Business Analytics Warehouse

Table 94 provides information about problems and solutions related to Unicode issues with Informatica and the Oracle Business Analytics Warehouse.

Table 94. 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 Mapping 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 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.
When Configuring the Informatica Server or Informatica Repository Server, there are major truncations in some of the tabs. For example, in the Server tab and the Repository tab in the Configure Informatica Service screen (Start > Programs > Informatica Server > Informatica Server Setup > Configure Informatica Service), the field names are truncated.	Informatica 6.x does not support non-English messages and resources. Delete the following files to display all messages in English. PC/PM client folder*411.dll PMServer folder\bin*411.dll PMRepServer folder\bin*411.dll
Unable to start the Informatica repository with the Informatica Repository Server on UNIX. The Oracle_BI_DW_Base (MS Windows Japanese and superset of Shift-JIS) is not compatible with that of pmrepagent (UTF-8 encoding of Unicode).	Make sure Japanese locales are installed on the UNIX machine. The environment variables on the UNIX machine are not complete. Set the environment variable PM_CODEPAGENAME to MS932. For other languages, set the variable appropriately.

Table 94. Unicode Issues with Informatica and Oracle Business Analytics Warehouse

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 through the Informatica Workflow Manager. 1 Start the Workflow Manager. 2 Connect to the Oracle_BI_DW_Base using Administrator as the user. 3 Click Menu Connection > Select Relational. 4 Select the appropriate OLTP connection (DB2_OLTP for DB2, ORA_OLTP for Oracle, MSSQL_OLTP or ODBC_OLTP for MSSQL). 5 For the code page select the UTF-8 encoding of Unicode or the appropriate code page, and then click OK. 6 Select the OLAP connection (DB2_OLAP for DB2, ORA_OLAP for Oracle, MSSQL_OLAP or ODBC_OLTP for MSSQL). 7 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 <code>locale</code> at the UNIX command prompt. Make sure the correct locales are installed on the UNIX machine. Set the environment variable <code>PM_CODEPAGENAME</code> correctly (For example set it to <code>MS932</code> for Japanese).

UNIX Installation Issues with the Oracle Business Analytics Warehouse

Table 95 provides information about issues related to installing the Oracle Business Analytics Warehouse on UNIX.

Table 95. UNIX Installation Issues with Oracle Business Analytics Warehouse

Symptom / Error Message	Probable Cause / Solution
When starting Informatica in pmserver.log, the following error messages appear: FATAL ERROR: LM_36023 [Tue Mar 11 23:12:41 2003]: (9968 1) The IP address [172.0.129.141] for this machine did not match the IP address [172.20.94] in the repository for server [ORACLE_BI_DW_SERVER]. FATAL ERROR: SF_34014 [Tue Mar 11 23:13:41 2003]: (9968 1) Server initialization failed. INFO: SF_34014 [Tue Mar 11 23:13:41 2003]: (9968 1) Server shut down.	Launch Workflow Manager, then right-click on Oracle_BI_DW_Base and click More. Make sure the Host Name is the machine name where the Informatica Repository Server resides.
When connecting to the Informatica Repository Server from the Repository Administration Console, the following error messages appear: ■ Oracle databases running on Solaris or HP: Database Connection Error: Database driver event...Error occurred loading library [Bad magic number for shared library: /export/home/oracle/9.0.1.3/lib/libclntsh.sl] Database driver event...Error occurred loading library [libpmora8.sl] ■ DB2 databases running on AIX: (17764 1) Database Connection Error: Database driver event...Error occurred loading library [No such file or directory] Database driver event...Error occurred loading library [libpmdb2.a]	Use Oracle 32-bit library. - a Set the environment variable as follows: LD_LIBRARY_PATH=/export/home/oracle/9.0.1.3/lib32 - b Restart the Informatica Repository Server service. ■ DB2 databases running on AIX: (17764 1) Database Connection Error: Database driver event...Error occurred loading library [No such file or directory] Database driver event...Error occurred loading library [libpmdb2.a] Set the environment variable as follows: ■ For DB2 v8: setenv LIBPATH \${LIBPATH}:/usr/opt/db2_08_01/lib ■ For DB2 v7 setenv LIBPATH \${LIBPATH}:/usr/lpp/db2_07_01/lib Restart the Informatica Repository Server service, after setting the variable.

Table 95. 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. ■ For Unicode UTF8, set the environment variable to American_America.UTF8. ■ 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.

Table 95. UNIX Installation Issues with Oracle Business Analytics Warehouse

Symptom / Error Message	Probable Cause / Solution
A session fails (any session), and the following error message appears: TM_6227 Error: codepage incompatible in session [SDE_RecordExtractStart]. The source database DB2_OLTP and server ORACLE_BI_DW_SERVER do not have compatible code pages. (One way compatibility is required).	For Informatica servers running on UNIX, in the pmserver.cfg file, set ValidateDataCodePages to No. For Informatica servers running on Windows, navigate to Informatica Server setup > Configuration, and remove the check from the Validate Data CodePages check box.
When trying to start the Informatica Server, the following error message appears: <pre>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.</pre>	Set the environment parameters as follows: <pre>setenv PATH /export/home/informatica/ pm:.\${PATH} setenv LD_LIBRARY_PATH /export/home/ informatica/ pm:\${LD_LIBRARY_PATH}</pre>

Oracle Business Analytics Warehouse Upgrade Issues

[Table 96](#) provides information about problems and solutions related to issues with upgrading the Oracle Business Analytics Warehouse.

Table 96. 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: <code>set(col)=rtrim(col)</code> - 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: <pre> SELECT DISTINCT CASE WHEN G1.VAL IS NULL THEN RTRIM(LTRIM(DS.AREA)) ELSE G1.VAL END AS AREA, CASE WHEN G1.NAME IS NULL THEN RTRIM(LTRIM(DS.AREA)) ELSE G1.NAME END AS AREA_I, CASE WHEN G2.VAL IS NULL THEN RTRIM(LTRIM(DS.SUB_AREA)) ELSE G2.VAL END AS SUB_AREA CASE WHEN G2.NAME IS NULL THEN RTRIM (LTRIM(DS.SUB_AREA)) ELSE G2.NAME END AS SUB_AREA_I CASE WHEN G3.VAL IS NULL THEN RTRIM(LTRIM(DS.SEV_CD)) ELSE G3.VAL END AS SEV_CD CASE WHEN G3.NAME IS NULL THEN RTRIM (LTRIM(DS.SEV_CD)) ELSE G3.NAME END AS SEV_CD_I CASE WHEN G4.VAL IS NULL THEN RTRIM(LTRIM(DS.PRIO_CD)) ELSE G4.VAL END AS PRIO_CD CASE WHEN G4.NAME IS NULL THEN RTRIM (LTRIM(DS.PRIO_CD)) ELSE G4.NAME END AS PRIO_CD_I FROM W_SRVREQ_DS DS, W_LOV_G G1, W_LOV_G G2, W_LOV_G G3, W_LOV_G G4 WHERE W_SRVREQ_DS LEFT OUTER JOIN V_LOV_G G1 ON DS.AREA = G1.NAMEVAL AND G1.TYPE = 'SR_AREA' LEFT OUTER JOIN W_LOV_G G2 ON DS.SUB_AREA = G2.NAMEVAL AND G2.TYPE = 'SR_AREA' LEFT OUTER JOIN W_LOV_G G3 ON DS.SEV_CD = G3.NAMEVAL AND G3.TYPE = 'SR_SEVERITY' LEFT OUTER JOIN W_LOV_G G4 ON DS.PRIO_CD = G4.NAMEVAL AND G4.TYPE = 'SR_PRIORITY'} </pre>

Oracle BI Interactive Dashboard Issues

Table 97 provides information about problems and solutions related to Oracle BI Interactive Dashboards.

Table 97. Oracle BI Dashboard Issues

Symptom / Error Message	Probable Cause / Solution
Error Codes: OPR4ONWY:U9IM8TAC:OI2DL65P File: ./project/webodbcaccess/odbcstatementimpl.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,killcas:killstba) at OCI call OCISstmtFetch. [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).

A

Installing the Usage Accelerator Application

This appendix provides instructions for installing the optional Usage Accelerator application.

NOTE: This step is applicable to all versions of the Usage Accelerator application.

If an execution plan in DAC metadata contains a Usage Accelerator Subject Area, it should include a preceding task "Create View For Usage Accelerator". All out of the box execution plans that contain Usage Accelerator subject area are configured with this preceding task. This step is required if you want to create a new execution plan or modify an existing execution plan to include "Usage Accelerator Subject area". This step is also required if you are using any customized subject area containing Usage accelerator warehouse tables for your execution plans. These tables are prefixed with "W_UA*".

NOTE: The following instructions in this appendix are required only if you are planning to use the Usage Accelerator analytics application integrated with Siebel CRM 7.5.3 applications. If you licensed the Usage Accelerator application versions 7.7 or 7.7.1, you should follow the installation instructions in Oracle Business Intelligence Platform Installation and Configuration Guide. No additional steps are required to install these versions of the Usage Accelerator application.

This appendix contains the following topics:

- [Usage Accelerator Application Integration with Siebel 7.5.3 CRM Application on page 417](#)
- [Usage Accelerator-Analytics Adapter Files on page 420](#)

Usage Accelerator Application Integration with Siebel 7.5.3 CRM Application

NOTE: The following steps are required only if are planning to use the Usage Accelerator in an integrated environment with a Siebel Business application version 7.5.3.

The Usage Accelerator installation process consists of installing the Usage Accelerator application and importing seed data into the transactional database. You install the Usage Accelerator application using the standard Oracle Business Intelligence installer. The installation options that are available to you depend on the license key you enter.

To integrate the Usage Accelerator application

- 1 Import repository objects to enable embedded Analytics.

Use Siebel tools to imports "SRW_753_UA_Integration_objects.sif" file into tools repository.

- 2 Lock the following projects:

- Analytics

- ERM
- Siebel Field Service
- Siebel Marketing Enterprise
- Siebel Channel
- Siebel Sales Enterprise
- Siebel Service Enterprise
- Siebel Universal Agent

NOTE: Do not apply the schema in the transactional database after importing these objects in tools repository. You might see some tables created in this step. These tables would have been applied in the transactional database (for more information, see [Updating Siebel Transactional Database Schema Definitions on page 138](#)).

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 > 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 > Administration - Integration > 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 98. 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.

- 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 99. 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 the Usage Accelerator reports.

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 100.

File	Description
seed_753_UA.dat	Siebel Seed Data file containing language-independent seed data records for the following tables: S_WI_SYMURL, S_WE_SYMURL_ARG, S_RESP, S_APP_VIEW, S_APP_VIEW_RESP.
seed_753_db2_UA.inp 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.

B

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:

- [Process of Maintaining Translation Tables for Oracle BI on page 421](#)
- [About Translating Presentation Services Strings on page 425](#)
- [About Installing Oracle Business Intelligence Localization Packs on page 425](#)
- [Changing the Default Currency in Analytics Applications on page 425](#)

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 Presentation Services Administration Guide*.

This process includes the following tasks:

- [Upgrading Oracle Business Intelligence Seed Data for Non-English Locales on page 422](#)
- [Externalizing Metadata Strings on page 423](#)
- [Adding Custom Translations to the W_LOCALIZED_STRING_G Table on page 424](#)

Upgrading Oracle Business Intelligence Seed Data for Non-English Locales

This topic is part of the ["Process of Maintaining Translation Tables for Oracle BI."](#)

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_LOCALISED_STRING_G`. This data allows the Analytics applications metadata to be viewed in non-English languages. Because this seed data is normally part of the Siebel operational application suite, you would have loaded part of this data when you installed the Siebel operational application. The additional task described in ["Importing Locale Seed Data Into The Translation Table \(W_LOCALIZED_STRING_G\)"](#) updates the seed data and makes it current for the Oracle Business Intelligence Applications. This process must be performed once for each language in which your data is to be viewed.

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).

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 procedure.

NOTE: This procedure can be performed only by a BI Administrator.

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 the connection parameters shown in the following table with the values appropriate to your database environment.

Import parameters:
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 database connectivity to the machine on which the Oracle BI Server resides.

NOTE: Unicode connectivity can be used to access databases that do not support Unicode.

Externalizing Metadata Strings

This topic is part of the [“Process of Maintaining Translation Tables for Oracle BI.”](#)

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 101](#).

Table 101. Example of W_LOCALIZED_STRING_G Translation Table

MSG_NUM	MSG_TEXT	LANG_ID
CN_Customer_Satisfaction	Customer Satisfaction	ENU
CN_Customer_Satisfaction	Kundenzufriedenheit	DEU
CN_Customer_Satisfaction	Satisfação do cliente	PTB

By default, the Oracle Business Intelligence repository is configured to run in English only. To deploy in any other language, you must externalize the metadata strings, as described in the following procedure.

To externalize metadata strings in the Oracle Business Intelligence repository

- 1 Stop the Oracle BI Server.
- 2 Open OracleBIAnalyticsApps.rpd using the Oracle BI Administration Tool in offline mode.
- 3 Select the entire Presentation layer and right-click the mouse to display the menu.
 - From the pop-up menu, select Externalize Display Names. (A check mark appears next to this option the next time you right-click on the Presentation layer.)

- Unselect the Presentation layer.

NOTE: When Externalize Display Names is checked, all metadata strings are read from the W_LOCALIZED_STRING_G table in the data warehouse.

- 4 In the Physical layer, select the Externalized Metadata Strings database icon. Expand the tree.

- 5 Double-click Internal System Connection Pool.

In the Connection Pool window General tab, the field Data source name should point to the data warehouse.

- 6 Click OK and exit the Oracle BI Administration Tool.

- 7 Restart the Oracle BI Server.

Adding Custom Translations to the W_LOCALIZED_STRING_G Table

This topic is part of the [“Process of Maintaining Translation Tables for Oracle BI.”](#)

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).

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.
- 3 Obtain the Message Key from the Oracle BI Administration Tool as follows:
 - In the Oracle BI Administration Tool, right-click on the new Presentation layer metadata object and select Properties from the menu.
 - The Message key is displayed in the dialog box under Custom Display Name. The Message key is the part that starts with CN_.

For example, double-click the Pipeline catalog folder in the Presentation layer. The Custom Display name is Valueof(NQ_SESSION.CN_Pipeline). CN_Pipeline is the Message Key.
- 4 Enter your deployment language in the new record.

- 5 Enter the Message Type required (for example, Metadata, FINS_Metadata).
- 6 Select the Message Level *AnalyticsNew*, then do the following:
 - In the Message Text column, add the translation of the object.
 - Check the flags (set to Yes) for the Translate and Active columns.
 - Set the Error Message # column to 0.
- 7 Enter the required Message Facility (for example, HMF, FIN).
- 8 Repeat [Step 3](#) through [Step 7](#) for each new metadata object string.
- 9 Exit the database administration tool, then restart the Oracle BI Server.

About Translating Presentation Services Strings

The translations for such Presentation Services objects as report and page names are stored in the captions.xml file. 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 Presentation Services Administration Guide*.

About Installing Oracle Business Intelligence Localization Packs

During your Siebel operational application installation, you also should have installed a Siebel operational application language pack. The language pack installation process populates the W_LOCALIZED_STRING_G table from seed data shipped with the Siebel operational application software. The languages available in the W_LOCALIZED_STRING_G table are the same as the language packs that you install. For more information on the Siebel operational application language pack installation, see the server installation guide for the operating system you are using.

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\OracleBIData\Web\config\
- UNIX:

\$INSTALLDIR/OracleBIData/Data/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.

C

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:

- [Importing Oracle's Siebel Industry Applications Seed Data on page 427](#)
- [Completing the Initialization in the Siebel Operational Application on page 428](#)

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 \$INSTALLDIR\SeedData\ 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 siebsrvr\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 Presentation Services Administration Guide*.

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:
 - [Updating the Siebel Operational Application on page 428](#)
 - [Reapplying Customized Style Sheets on page 429](#)
 - [Process of Configuring Oracle Business Intelligence with Oracle's Siebel Web Extension \(SWE\) on page 430](#)
 - [Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services on page 431](#)
 - [Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE on page 432](#)
 - [Process of Configuring Oracle BI Action Links on page 432](#)
 - [Testing Action Links on page 432](#)
- Customizing the operational application home page, dashboards, or content, or add views to a dashboard. See the following topics:
 - [Process of Integrating Analytics and Oracle's Siebel Operational Applications on page 433](#)
 - [Accessing Optional Analytics Applications on page 433](#)
 - [Configuring the Browser Settings for Operational Applications on page 434](#)
 - [Process of Customizing Oracle BI Application Home Page and Dashboards on page 435](#)
 - [Customizing Oracle BI Content on the Siebel Operational Application Home Page on page 435](#)
 - [Determining the Oracle BI Report Path Argument on page 437](#)
 - [Adding Views for Custom Oracle BI Interactive Dashboards on page 438](#)
 - [In Oracle Business Intelligence, set up the dashboards to your requirements. on page 441](#)
 - [Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client on page 440](#)
 - [Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client on page 440](#)

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 > 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: The following additional information is applicable only to UNIX platforms.

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.

Reapplying Customized Style Sheets

For Oracle's Siebel Business Analytics versions 7.7 and later, new styles have been appended to the following style sheets:

- Go.css
- PortalBanner.css
- PortalContent.css
- Views.css

These new classes are identified in the style sheets. In this version of Oracle Business Intelligence, new styles and files must be added to the underlying style sheet (for example, to `s_Siebel7`). For complete functionality, any custom styles require similar updating. In addition, `views2.css` and some other files have been added to the `s_` directory. For custom column formatting to work properly, references to font sizes and families should also be removed from the `td` styles in `PortalBanner.css`, `PortalContent.css`, `Views.css`.

Because of these changes to styles, review your deployment's customizations manually, reapply them, and test them thoroughly to ensure that there are no problems.

After you have tested them, but before you copy the default `views.css` files back to the implementation server, perform the following task on the server to clear the server caches.

To clear the server caches and restore your default views

- 1** Shut down Oracle Business Intelligence Server, Oracle BI Presentation Service and IIS.

- 2 Remove your custom views.css from the directory where it has been installed.

For example:

```
$INSTALL\Web\App\Res\s_Siebel7\b_mozilla_4
```

or

```
$INSTALL\OracleBIData\Web.
```

- 3 Clear the Oracle BI Presentation Services Server Cache.

In the C:\WINNT\Temp directory, delete the nQs_*.temp files.

- 4 Clear the Browser Cache.

From the Internet Explorer menu, navigate to Tools > Internet Options > Settings > 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.

Process of Configuring Oracle Business Intelligence with Oracle's Siebel Web Extension (SWE)

Whenever you run Oracle Business Intelligence and Siebel Web Extension (SWE) on separate machines, you must perform additional configuration steps in order for action links and interactive charts to work. If, for example, you plan to run the SWE and Oracle Business Intelligence Presentation Services on different Web servers, you must use some kind of networking or load balancing mechanism to create a single logical domain (or virtual IP address) for the two machines.

CAUTION: When Oracle BI Applications Server and Oracle BI Presentation Services are installed on different machines and load balancing software is used to create a single virtual IP address, the action links feature works only if you log onto the Siebel application using a virtual Web browser. Action links fail on a dedicated client if the SWE and Oracle BI Presentation Services servers are on separate machines. Either use a virtual Web browser, or install the Oracle BI Applications Web Server and Oracle BI Presentation Services on the same machine.

When one virtual IP address is created for two machines, the Web browser accesses one IP address and is still routed to different physical machines, based on the port accessed. From the browser, it appears that both servers are running on the same IP address.

You can use any of several physical methods to create a single logical domain, such as running SWE and Oracle Business Intelligence Presentation Services on a single machine if you are not load balancing the SWE, or using a router to do the mapping, or using load balancing software. Your company must determine the best mechanism to accomplish this routing given the topology being used.

Configuring Oracle Business Intelligence to work with Siebel Web Extension includes the following tasks:

- [Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services on page 431](#)

- Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE on page 432

Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services

This task is part of the “[Process of Configuring 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.

Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE

This task is part of the [“Process of Configuring 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 [“Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services”](#) on page 431.

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.

Process of Configuring 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 Sales Analytics dashboard to a specific record in a Siebel 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 Presentation Services 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:

- [Testing Action Links on page 432](#)

Testing Action Links

This task is part of the [“Process of Configuring 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 logons. Go to Service Analytics screen > Employee Analytics > 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** Now click any ROW_ID action link.
You should be in the Employee screen in Oracle's Siebel Call Center operational application.

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:

- [Accessing Optional Analytics Applications on page 433](#)
- [Configuring the Browser Settings for Operational Applications on page 434](#)
- [Process of Customizing Oracle BI Application Home Page and Dashboards on page 435](#)

Accessing Optional Analytics Applications

This task is part of "[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 102](#) shows the additional options for Oracle Business Intelligence.

Table 102. Oracle Business Intelligence Applications Options

Siebel Application	Options
Sales	Sales Analytics, 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 Sales Analytics with your Siebel 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 > 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.

Web Group Option
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.

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 ["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 > Internet options > 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.

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:

- [Customizing Oracle BI Content on the Siebel Operational Application Home Page on page 435](#)
- [Determining the Oracle BI Report Path Argument on page 437](#)
- [Adding Views for Custom Oracle BI Interactive Dashboards on page 438](#)
- [Verifying NQHOST Setup for Pharma Disconnected Analytics Client on page 438](#)
- [Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client on page 440](#)
- [Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client on page 440](#)

For a list of Pharma Disconnected Analytics components that are installed with Oracle BI Applications, see [Pharma Disconnected Analytics Administration - Supplemental Information on page 539](#).

Customizing Oracle BI Content on the Siebel Operational Application Home Page

This task is part of the [“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 [In Oracle Business Intelligence, set up the dashboards to your requirements. on page 441.](#)

The figure below shows example Inline and IFrame symbolic URL arguments.

Figure 34. 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	Login	✓		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 > 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 [“Determining the Oracle BI Report Path Argument” on page 437](#) to determine the path to the Analytics report.

Determining the Oracle BI Report Path Argument

This task is part of the [“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 > 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

Name	Type	Path Argument Value	Append	Sequence #
Syndicate	Constant	Siebel	Y	3
nQUser	Command	UseSiebelLoginId	Y	4
nQPassword	Command	UseSiebelLoginPassword	Y	5
PostRequest	Command	PostRequest	Y	6

Adding Views for Custom Oracle BI Interactive Dashboards

This task is part of the [“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 > Integration Administration > 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.

Verifying NQHOST Setup for Pharma Disconnected Analytics Client

This task is part of the [“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`

Disconnected Client Symbolic URL Name
ePharmaHomePageAnalytics1
ePharmaHomePageAnalytics2
ePharmaHomePageAnalytics3
ePharmaHomePageAnalytics4
SISMLSDistrictManagerDashboard1
SISMLSSalesRepDashboard1
SiebelAnswers
SiebelDelivers

The following table shows the symbolic URL path arguments for SiebelAnswers and SiebelDelivers.

Name	Type	Path Argument Value	Append	Sequence #
Cmd	Constant	Answers for SiebelAnswers	Y	1
		Delivers for SiebelDelivers		
nQUser	Command	UseSiebelLoginId	Y	2
nQPassword	Command	UseSiebelLoginPassword	Y	3

Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client

This task is part of the [“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.
- 4 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.
- 5 Set the UseCachedExternalContent parameter to FALSE.
- 6 Save and close the file.

Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client

This task is part of the [“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.
- 4 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.

- 5 Set the EnableFQDN parameter to FALSE.
- 6 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

- 7 In Oracle Business Intelligence, create the dashboards.
NOTE: Only an Oracle BI Presentation Services administrator can perform this step.
- 8 In Oracle Business Intelligence, set up the dashboards to your requirements.

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.

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 Sales 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.

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

D

Integrated Security for Oracle BI Applications

NOTE: This appendix gives an overview of the configuration of data-level and object-level security groups for areas within Oracle Business Intelligence applications integrated with Siebel (CRM) and Oracle E-Business Suite (operational) applications. The specific processes for configuring Oracle Business Intelligence repositories and Oracle Business Intelligence Presentation Services are described in the *Oracle Business Intelligence Server Administration Guide* and the *Oracle Business Intelligence Presentation Services Administration Guide*, respectively. This book assumes that you are familiar with *Security Guide for Siebel eBusiness Applications* (on Oracle's Siebel Bookshelf), especially the topics on user authentication, basic access control, and data-level security.

This appendix includes the following main topics:

- [Types of Security in Oracle BI on page 443](#)
- [Data-Level Security in CRM Analytics Applications on page 448](#)
- [Implementing Data-Level Security in the Oracle BI Repository on page 452](#)
- [Initialization Blocks and Security-Related Information in Oracle BI on page 453](#)
- [Filters for Oracle BI Security Groups on page 454](#)
- [Configuring Oracle BI Repository Table Joins for Security on page 455](#)
- [Responsibilities and Data-Level Visibility in Sales Analytics Applications on page 456](#)
- [Changing the Security Group from Position-Based to Organization-Based in Oracle BI on page 458](#)
- [Metadata Object-Level Security in Oracle BI Applications on page 458](#)
- [Oracle Partner Analytics Security Settings on page 460](#)
- [Consumer Sector Analytics Security Settings on page 468](#)
- [Communications, Media, and Energy \(CME\) Analytics Security Settings on page 468](#)
- [Financial Services Analytics Security Settings on page 469](#)
- [Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics Security Settings on page 472](#)
- [Usage Accelerator Analytics Security Settings on page 474](#)
- [Usage Accelerator Analytics Security Settings on page 474](#)

Types of Security in Oracle BI

Two different levels of security configuration are supported by Oracle Business Intelligence:

- Data-level security
- Object-level security

Oracle Business Intelligence applications security is based on Responsibilities, Positions, and Oracle Business Intelligence repository groups. A Repository group and a Presentation Services group are created for each Oracle Business Intelligence-specific responsibility defined in the Siebel (CRM) or Oracle E-Business Suite operational application. The group names are exactly the same as the responsibility name.

In a Siebel (CRM) or Oracle E-Business Suite operational application, you assign users to the appropriate Analytics responsibilities, and the membership is automatically reflected in the repository as well as the in Presentation Services. The user inherits the appropriate permissions from the responsibilities.

NOTE: Users should always be created in the operational application databases or directory services such as LDAP, never in the Oracle Business Intelligence repository. For details on integration with Oracle E-Business Suite please refer to the section [Implementing Security With Oracle EBS on page 476](#).

User Access and User Responsibilities

In previous releases (that is, prior to the 7.9 release), permission and access were determined by user's memberships in various responsibilities.

In this release, new security groups have been introduced in the Analytics repository only, which are based on the position hierarchy level of the user. The administrator can check which group a user belongs to by opening the repository in online mode (while the user is logged in) and checking the value of the HIER_LEVEL variable in Manage\Sessions screen. These groups are called:

- Hierarchy Level (Base), Hierarchy Level 1 to Hierarchy Level 8
- Hierarchy Level (Top)

A user is assigned dynamically to one of these groups at Login time. For more information on these new groups, see [Filters for Oracle BI Security Groups on page 454](#).

For all other responsibilities, the term Responsibility is interchangeable with the terms Repository group and Presentation Services group; they have a one-to-one correspondence.

Oracle Business Intelligence come with certain pre-configured responsibilities. [Table 103 on page 445](#) shows these pre-configured responsibilities.

Checking Analytics 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.
- Go to the Oracle BI Web and click the My Account link.

The group membership for the user is shown near the bottom of the Web page.

Registering a New User Responsibility in CRM Analytics Applications

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 Server metadata through Oracle BI Web.
- After the metadata is reloaded, the user must log out from the Siebel or Oracle E-Business Suite operational application, then log in again.

Default Security Settings in CRM Analytics Applications

The User Administrator and the Group Administrators are a special user and group 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.

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.

Table 103. Oracle's Siebel Analytics Applications Pre-configured Responsibilities

Application	Responsibility
Sales	Sales Analytics Administrator
	VP of Sales Analytics
	Sales Manager Analytics
	Sales Representative Analytics
Marketing	Marketing Analytics User
	Marketing Analytics Administrator
Interactive Selling	ISS Analytics User
	ISS Analytics Administrator
Service	Service Analytics VP
	Service Analytics Manager
	Service Analytics Administrator
	Service Analytics Service User
	Service Analytics Call Center User
	Service Analytics eMail User

Table 103. Oracle's Siebel Analytics Applications Pre-configured Responsibilities

Application	Responsibility
Partner	Partner Service Rep Analytics User
	Partner Sales Rep Analytics User
	Partner Executive Analytics User
	Partner Operations Analytics User
	Partner Sales Manager Analytics User
	Partner Service Manager Analytics User
	Channel Accounts Manager Analytics User
	Channel Executive Analytics User
	Channel Operations Analytics User
	Channel Marketing Manager Analytics User
Employee Relationship Management	ERM Analytics VP
	ERM Analytics Manager
	ERM Analytics HelpDesk Agent
	ERM Analytics HelpDesk
	ERM Analytics User
	ERM Analytics HR
	Executive Analytics User
	Executive Analytics Administrator
Communications, Media, & Energy	CM Marketing Analytics User
	CM Marketing Analytics Administrator
	CM Sales Analytics User
	CM Sales Analytics Administrator
	CM Service Analytics User
	CM Service Analytics Administrator
Financial Services	Finance Analytics User
	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)
	Financial Services Analytics User

Table 103. Oracle's Siebel Analytics Applications Pre-configured Responsibilities

Application	Responsibility
Consumer Goods	CS Analytics VP Sales
	CS Analytics Key Account Manager
	CG Sales Analytics User
	CG Sales Analytics Administrator
Pharma Sales	LS Administrator
	PH US Sales Analytics User
	PH EMEA Sales Analytics User
	PH Sales Analytics Administrator
	PH US Marketing Analytics User
	PH EMEA Marketing Analytics User
	PH Marketing Analytics Administrator
	PH Disconnected Analytics User
	PH Disconnected Analytics Admin
	PH Medical Education Analytics User
	PH Medical Education Analytics Admin
	PH US Call Activity Analytics User
	PH EMEA Call Activity Analytics User
	PH Call Activity Analytics Admin
	PH US Executive Analytics User
	PH EMEA Executive Analytics User
	PH Executive Analytics Admin
Automotive	AT Service Analytics User
	AT Service Analytics Administrator
	AT Sales Analytics User
	AT Sales Analytics Administrator
	AT Partner Analytics User
	AT Partner Analytics Administrator

Table 103. Oracle's Siebel Analytics Applications Pre-configured Responsibilities

Application	Responsibility
Usage Accelerator NOTE: The Usage Accelerator Tracking responsibility is not visible in Oracle Business Intelligence. The Administrator must associate this responsibility with each user to be tracked, in the transactional database.	Usage Accelerator Administrator
	Usage Accelerator Sales Executive
	Usage Accelerator Sales Manager
	Usage Accelerator Sales Rep
	Usage Accelerator FINS Administrator
	Usage Accelerator FINS Sales Executive
	Usage Accelerator FINS Sales Manager
	Usage Accelerator FINS Sales Rep
	Usage Accelerator Tracking (transactional database)

Data-Level Security in CRM Analytics Applications

Data-level security defines what a user in one of Siebel (CRM) and Oracle E-Business Suite applications sees 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).

Where is Data-Level Security Configured?

Data-level security in Oracle Business Intelligence applications is based on the position and organization-based security model of the Siebel or Oracle E-Business Suite operational applications. Data-level security is configured exclusively through the Analytics repository using Repository groups. It is configured at the logical layer for sensitive Dimensions (for example, Account, Opportunity), and Facts (for example, Revenue, Order Item). [Table 104](#) shows the list of tables supported out-of-the-box. If required, the Administrator can implement further security restrictions on any table or column.

The security definition for the dimension decides the general area within which the user operates. The reports can further restrict access to the data within the report. For example, a report returns only rows that are owned by the user in the following scenarios:

- A report contains a filtered column (for example, "Dim - Opportunity."Closed Opportunity Revenue"), which is secured by a filter (for example, Core."Dim - Opportunity."Base Level Current Login" = VALUEOF(USER)).
- A report contains a metric or attribute from a fact or dimension that is secured by a filter (for example, Core."Dim - Security Dimension"."Current Base Level Login" = VALUEOF(USER)).

Other data security models such as custom models can be configured in the metadata using similar techniques. This security configuration requires an understanding of the business model setup and filtering mechanisms. See *Oracle Business Intelligence Server Administration Guide* on using variables in the repository.

The process of configuring data-level security is described in the topic [“Implementing Data-Level Security in the Oracle BI Repository” on page 452.](#)

Table 104. List of Security Supported Tables

Table Names
Dim - Asset
Dim - Case
Dim - Contact
Dim - Customer
Dim - Incident
Dim - Lead
Dim - Opportunity
Dim - Order
Dim - Quote
Fact - Account Action Plan
Fact - Account Summary
Fact - Account Usage
Fact - Activity Usage
Fact - Asset Action Plan
Fact - Asset Summary
Fact - Asset Usage
Fact - Contact Action Plan
Fact - Contact Summary
Fact - Contact Usage
Fact - CRM - Asset
Fact - CRM - Order Item
Fact - CRM - Order Item Quarter Ago
Fact - CRM - Order Item Year Ago
Fact - CRM - Revenue
Fact - Opportunity Action Plan
Fact - Opportunity Summary

Table 104. List of Security Supported Tables

Table Names
Fact - Opportunity Usage
Fact - Quote Usage
Fact - User Adoption
Facts - Call Account
Facts - Call Account [MAT]
Facts - Call Account [MAT] LY
Facts - Call Account [Top Total]
Facts - Call Attendee
Facts - Call Attendee [MAT]
Facts - Call Attendee [MAT] LY
Facts - Call Attendee [Top Total]
Facts - Call Contact
Facts - Call Contact [MAT]
Facts - Call Contact [MAT] LY
Facts - Call Contact [Top Total]
Facts - MedEd Event [Pre-post Ed Event Total Pop]
Facts - MedEd Event and IDS [Pre-post Ed Event]
Facts - MedEd Event and Rx [Pre-post Ed Event]
Facts - Medical Education
Facts - Objective
Facts - Profile Rank
Facts - Syndicated Data Direct Sales
Facts - Syndicated Data Direct Sales [Top Total]
Facts - Syndicated Data Direct Sales Market
Facts - Syndicated Data Direct Sales Market [Top Total]
Facts - Syndicated Data Indirect Sales
Facts - Syndicated Data Indirect Sales [Top Total]

Table 104. List of Security Supported Tables

Table Names
Facts - Syndicated Data Indirect Sales Market
Facts - Syndicated Data Indirect Sales Market [Top Total]
Facts - Syndicated Data NRx/TRx
Facts - Syndicated Data NRx/TRx [Top Total]
Facts - Syndicated Data NRx/TRx Market
Facts - Syndicated Data NRx/TRx Market [Top Total]
Facts - Syndicated Data Plantrak
Facts - Syndicated Data Plantrak [Top Total]
Facts - Syndicated Data Plantrak Market
Facts - Syndicated Data Plantrak Market [Top Total]
Facts - Syndicated Data Weekly
Facts - Syndicated Data Weekly [Top Total]
Facts - Syndicated Data Weekly Market
Facts - Syndicated Data Weekly Market [Top Total]
Facts - Syndicated Data XPlantrak
Facts - Syndicated Data XPlantrak [Top Total]
Facts - Syndicated Data XPlantrak Market
Facts - Syndicated Data XPlantrak Market [Top Total]
Facts Compound - Call Account
Facts Compound - Call Account [MAT] Time Series
Facts Compound - Call Activity Medical Education
Facts Compound - Call Activity Profile Rank
Facts Compound - Call Activity Syndicated data
Facts Compound - Call Attendee
Facts Compound - Call Attendee [MAT] Time Series

Table 104. List of Security Supported Tables

Table Names
Facts Compound - Call Contact
Facts Compound - Call Contact [MAT] Time Series
Facts Compound - MedEd/Pre-post Ed Event/Pre-post Total Pop
Facts Compound - Syndicated Data Indirect Sales
Facts Compound - Syndicated Data Indirect Sales Time Series
Facts Compound - Syndicated Data NRx/TRx
Facts Compound - Syndicated Data NRx/TRx Time Series
Facts Compound - Syndicated Data Plantrak
Facts Compound - Syndicated Data Plantrak Time Series
Facts Compound - Syndicated Data Weekly
Facts Compound - Syndicated Data XPlantrak

Implementing Data-Level Security in the Oracle BI Repository

This topic describes the process of configuring data from the Oracle Business Analytics Warehouse for Siebel (CRM) and Oracle E-Business Suite deployments, and refers only to the Oracle Business Analytics Warehouse tables. The setup for the transactional database data is similar, although the physical schema joins are different.

In this release, the Position based data level security has been modified by introducing a new table (W_POSITION_DH) to replace the old bridge table W_PARTY_LOGIN, which was used to grant access to managers to the records owned by somebody in their organization. The new table W_POSITION_DH is a flattened hierarchy of W_POSITION_D, and it is treated as a slowly changing dimension of type 2.

A new record is created every time the Primary employee for that position or the parent division of that position are modified. Consequently, every Position in the transaction database table (S_POSTN) can be represented by more than one record in this table (identified by W_POSITION_DH.BASE_POSTN_ID=S_POSTN.ROW_ID), 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 BASE_POSTN_ID at any time. The columns that are not prefixed with CURRENT reflect the hierarchy structure for the same position during the period between EFFECTIVE_START_DT and EFFECTIVE_END_DT.

Installed out of the box, all data security filters in Oracle BI Applications are based on the Current hierarchy columns. The flattened hierarchy table supports 10 levels, but it can be easily extended to support more. Employee attributes are included at every level, to allow analysis based on the Employee in parallel with position. The employee considered here is obviously the Primary Employee for that position. Since managers of an employee at different hierarchy levels will be stored in different columns (for example, W_POSITION_DH.CURRENT_LVL1ANC_LOGIN, W_POSITION_DH.CURRENT_LVL2ANC_LOGIN...W_POSITION_DH.CURRENT_TOP_LVL_LOGIN), a different security group exists for every level of the hierarchy. In every group, a filter is added on every secured dimension/fact for that level, using the corresponding column added for that group level (for example, Core."Dim - Security Dimension"."Current Level 1 Login" = VALUEOF(NQ_SESSION."USER"). Users are associated automatically with the correct user group (Base Hierarchy Level, Hierarchy Level 1, and so on), based on their current hierarchy level (0-9). This new security design has been implemented in three business models (Core, Pharma and Usage Accelerator). In the Core business model, it only supports the Siebel or Oracle E-Business Suite application data warehouse. Primary Position Based Security group is still available in the repository, but has been disabled. For the other Siebel or Oracle E-Business Suite application based business models, the old data security is still enabled. Position based data security does not support Oracle BI Application data warehouse in this release.

Data-level security in Oracle BI Applications is implemented in three major steps:

- 1 Set up initialization blocks that obtain specific security-related information when a user logs in—for example, the user's primary position ID.
See [Initialization Blocks and Security-Related Information in Oracle BI on page 453](#).
- 2 Set up the filters for each security group on each logical table that needs to be secured.
See [Filters for Oracle BI Security Groups on page 454](#).
- 3 Set up the joins to the appropriate security tables in the metadata physical and logical layers.
See [Configuring Oracle BI Repository Table Joins for Security on page 455](#).

Initialization Blocks and Security-Related Information in Oracle BI

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. There are five initialization blocks, as follows:

■ Primary Position ID

This initialization block obtains the primary position of the given user, based on the user's login, and populates the variable PRIMARY_POSTN_ID. The SQL used by the block is run against the transactional database, and therefore reflects real-time position ID information.

■ **Primary Owner ID**

This initialization block obtains the owner ID for the given user. It obtains this information from the transactional database and populates the PR_OWNER_ID variable.

■ **Orgs for Org-based Security**

This initialization block obtains the list of organizations that a given user belongs to from the transactional database. The variable populated is ORGANIZATION.

■ **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, and therefore reflects the hierarchy level after the last ETL run that populated this table (W_POSITION_DH).

■ **Authorization**

This initialization block is used to associate users with all Security groups to which they belong. The association is based on the variable HIER_LEVEL, which is defined in the User Hierarchy Level initialization block, for the Hierarchy Level security groups, and on the table S_PER_RESP for the Primary Position based Security group. Therefore, User Hierarchy Level is a precedent to Authorization.

Filters for Oracle BI Security Groups

In addition to initialization blocks that implement data-level security in the Analytics repository, there are pre-configured groups (for Siebel or Oracle E-Business Suite application based data warehouses) that have been set up with filters on several logical tables. For Position based security, these groups are:

- Primary Position-Based Security
- Hierarchy Level (Base)
- Hierarchy Levels 1 to 8
- Hierarchy Level (Top)

The first group was used prior to the 7.9 release as the main group for position based security, and the Administrator was responsible for adding responsibilities to this group. In this release, the ten new groups replace Primary Position-Based Security.

For these new groups, users are assigned dynamically to one of these groups, based on their level in the Position hierarchy. The filter condition for these groups is defined using the following Security dimension for fact tables:

```
Core."Dim - Security Dimension"."Current Base Level Login" = VALUEOF(NQ_SESSION."USER")  
or the corresponding W_POSITION_DH alias for a dimension: Core."Dim - Opportunity"."Base  
Level Current Login" = VALUEOF(NQ_SESSION."USER")
```

Another group named Executive Visibility enables executives to be exempted from these data filters, using a filter on the same set of columns/tables as the Hierarchy level groups. The filter condition for this group is 1=1 for all tables, and given that the Oracle Business Intelligence server joins different filter conditions for the same user with an OR, being a member of the Executive group will nullify any data filter on the table imposed by the Hierarchy level security groups. The Administrator determines which OLTP responsibility should be added to the executive group.

Configuring Oracle BI Repository Table Joins for Security

You can add security to a new dimension or fact in Oracle Business Intelligence. You set up the joins for that table to the appropriate security tables in the metadata physical and logical layers. The following tasks uses a dimension W_AGREE_D (Agreements) 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_LOGIN,...to the logical table, and map them to the corresponding physical columns.
- 5** Open the security groups: Hierarchy Level (Base), Hierarchy Level 1,..., Hierarchy Level (Top), and add filters as follows:
 - a** Right-click the group and choose Properties.
 - b** In the Properties dialog box, click the Permissions box and select the Filter tab.
 - c** To add a new filter, click on the ellipsis box and find the business model layer table that needs to be secured.
 - d** Configure the WHERE clause on the table so that the data is filtered.

To add security support to a fact

- 1** Join the underlying physical table to Dim_W_POSITION_DH_Position_Hierarchy.
- 2** Join the logical table to the Dim - Security Dimension.
- 3** Open the security groups: Hierarchy Level (Base), Hierarchy Level 1,..., Hierarchy Level (Top), and add filters as follows:
 - a** Right-click the group and choose Properties.
 - b** In the Properties dialog box, click the Permissions box and select the Filter tab.
 - c** To add a new filter, click on the ellipsis box and find the business model layer table that needs to be secured.
 - d** Configure the WHERE clause on the table so that the data is filtered.

Responsibilities and Data-Level Visibility in Sales Analytics Applications

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 your Oracle's Siebel operational applications. It also describes the particular responsibilities associated with the application dashboards.

Oracle's Siebel Sales Analytics defaults to the following data-level visibility for certain responsibilities, as shown in [Table 105](#).

The visibility restrictions imposed vary by responsibility and by logical subject area, as follows:

- The Sales Manager has different visibility restrictions imposed in the Forecasting subject area than in the Pipeline, Customers, Orders, and Products subject areas.
- In general, users with the VP of Sales Analytics responsibility have no data-level security imposed, while those with the Sales Manager Analytics or Field Sales Representative Analytics responsibilities have primary-position security imposed.
- In primary position security, only those opportunities, orders, or forecasts owned by the user or the user's subordinates are visible to the user.

For a description of data-level security, see [Types of Security in Oracle BI on page 443](#) and *Security Guide for Siebel eBusiness Applications*.

Table 105. Oracle's Siebel Sales Analytics Default Data-Level Visibility

Responsibility	Subject Area	Default Visibility
VP of Sales Analytics	All	No data-level security is imposed.
Sales Manager Analytics Field Sales Representative Analytics	Forecasting	In addition to the Order, Opportunity, and Forecast dimensions, primary position data-level security is imposed on all facts. This means that these subject areas are visible to the user only for data owned by the user (or the user's subordinates).

Table 105. Oracle's Siebel Sales Analytics Default Data-Level Visibility

Responsibility	Subject Area	Default Visibility
Sales Manager Analytics Field Sales Representative Analytics	Pipeline Orders Customers	Primary position data-level security is imposed on: ■ Order dimension ■ Opportunity dimension ■ Account dimension ■ Opportunity Revenue, Expected Revenue, and Closed Revenue metrics ■ Number of Orders - Line Item Order Revenue (000) - Line Item Shipped Revenue (000) - Order Item Revenue - Rollup Total Order Revenue (000) - Rollup Average Order Size - Total Order Revenue - Total Order Revenue (000) - Total Shipped Revenue - Total Shipped Revenue (000) metrics
TeleMarketing Manager TeleMarketing Representative	All	No specific data-level security.
TeleSales Representative Direct Order Entry Agent	All	No specific data-level security. However, the primary-position security used for Sales Managers and Representatives can be used for the TeleSales Representative and Direct Order Entry Agent responsibilities as well. Note that the security model for the TeleMarketing Representative and Manager responsibilities must differ from that used for Sales Managers and Representatives. The security model is based on the CREATED_BY column in the Opportunity data model and not on the primary owner of the opportunity.

Changing the Security Group from Position-Based to Organization-Based in Oracle BI

By default in the Siebel ERM Analytics application, the responsibilities Partner Sales Rep Analytics User and Partner Service Rep Analytics User have been set up for position-based security. If your deployment requires a security group setup based on Organization, change the security groups using the following procedure.

To change the security group from Position-based to Organization-based

- 1 In the Oracle BI Administration Tool, go to Tools > Manage Security.
- 2 Click on Hierarchy.
- 3 Make the two security groups (Partner Sales Rep Analytics User and Partner Service Rep Analytics User) members of the Primary Org-Based group.
- 4 If you have been using the old position security mechanism (based on W_PARTY_LOGIN), or add the two groups above to the group 'Executive Analytics User' if you have been using the new security mechanism (based on the Position Hierarchy), remove the groups from the Primary Position-Based group.
- 5 Save changes to the RPD, then exit the Oracle BI Administration Tool.

Metadata Object-Level Security in Oracle BI Applications

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

Metadata Object-Level Security (Repository Groups)

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

Where is Repository Groups Security Configured?

Metadata object security is configured in the Analytics repository (OracleBIAnalyticsApps.rpd) using the Oracle BI Administration Tool. The User Group Everyone is denied access to each of the subject areas. Each subject area is configured to give explicit read access to selected related responsibilities. This access can be extended to table and column level.

NOTE: In the shipped product, only permissions at the subject area level have been configured.

The exceptions to the explicit configuration rule are the Communications and Financial Analytics industry applications, where there are tables and columns specific to these two industries scattered throughout the general Siebel operational application subject areas. These industry-specific metadata objects are hidden from other groups.

Oracle Business Intelligence supports hierarchies within the groups in the repository. In the Analytics repository there are certain groups that are parent groups, which define the behavior of all the child groups. Inheritance is used to let permissions ripple through to child groups. The parent groups and their purpose are shown in [Table 106](#).

Table 106. 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

Metadata Object-Level (Presentation Services) Security in Oracle BI Applications

Presentation Services objects, such as dashboards and pages, are controlled using Presentation Services groups, which have the same name as the Siebel responsibilities. Access to dashboards and pages is controlled using the Presentation Services groups. If you log on as a user who belongs to the Presentation Services group Field Sales Representative Analytics, then you see only the Overview, Forecasting, and Details pages within the Pipeline Dashboard. In a similar fashion, you see only dashboards that allow you access to at least one page within that dashboard. These groups are customized in the Oracle BI Web interface.

For Oracle Business Intelligence integrated with Oracle's Siebel operational applications, Presentation Services security makes use of the following principles:

- Security in Presentation Services has been pre-configured for the groups listed in [Table 106](#) 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.

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 Workforce, 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.

Partner Manager Role-Based Interactive Dashboards Mapping

[Table 107](#) provides the dashboard and page tab mapping for specific responsibilities in the Siebel PRM Partner Manager application.

Table 107. 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

Table 107. Siebel Responsibilities for PRM Analytics

Responsibility	Dashboard	Page Tab Name
Channel Executive Analytics User	Channel Customers	Customer Profile
	Channel Executive	Customer Satisfaction
	Channel Executive	Pipeline
	Channel Executive	Product
	Channel Executive	Program
	Channel Executive	Revenue
	Channel Executive	Service
	Channel Segmentation	Channel Mix
	Channel Segmentation	Partner Territory
	Channel 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

Table 107. Siebel Responsibilities for PRM Analytics

Responsibility	Dashboard	Page Tab Name
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
	Channel Training	Overview
	Channel Training	Performance

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 108](#).

Table 108. 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

Table 108. Responsibilities for PRM Partner Portal Analytics

Responsibility	Dashboard	Page Tab Name
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
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

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 Analytics 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 109](#), where a X indicates that subject area is visible for the user holding that responsibility.

Table 109. 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

Table 109. 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
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	

PRM Analytics Subject Area Visibility

The subject area visibility for roles in Partner Portal is shown in [Table 110](#), where a X indicates that subject area is visible for the user holding that responsibility.

Table 110. 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

Table 110. 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
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

PRM Analytics Data-Level Visibility

Oracle's Siebel 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 [“Implementing Data-Level Security in the Oracle BI Repository” on page 452](#). To change Partner Service Representative and Sales Representative to Organization-based security, follow the procedure shown in the topic [“Changing the Security Group from Position-Based to Organization-Based in Oracle BI” on page 458](#).

Table 111 shows the data-level security settings included for the responsibilities in Partner Manager and Partner Portal.

Table 111. Siebel PRM Data-Level Security Settings

Responsibility	Data-Level Security	Type	Comments
Channel Executive Analytics User	No	N/A	N/A
Channel Operations Analytics User	No	N/A	N/A
Channel Account Manager Analytics User	No	N/A	N/A

Table 111. Siebel PRM Data-Level Security Settings

Responsibility	Data-Level Security	Type	Comments
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.

Consumer Sector Analytics Security Settings

Table 112 describes the consumer sector responsibilities associated with each CS Dashboard.

Table 112. Consumer Sector Responsibilities Associated with Each CS Dashboard

Screen Name	Siebel Responsibility	Dashboard	Pages
Sales Analytics	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

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 ["Types of Security in Oracle BI" on page 443](#).

In addition to responsibilities provided by the operational applications, Oracle's Siebel Communications, Media, and Energy (CME) provides additional responsibilities, and responsibility-specific security, as indicated in [Table 113](#).

Table 113. CME Responsibilities Associated with Each CME Dashboard

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

Financial Services Analytics Security Settings

Oracle's Finance 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), and the Healthcare Analytics family of products (Oracle Healthcare Sales Analytics, Oracle Healthcare Service Analytics, Oracle Healthcare Marketing Analytics, Oracle Healthcare Partner Manager Analytics) apply Oracle's Siebel operational applications security model. This security model is described in the topic ["Types of Security in Oracle BI" on page 443](#). In addition to responsibilities provided by the Siebel operational applications, these applications provide additional responsibilities, and responsibility-specific security, as indicated in [Table 114 on page 471](#).

For the Financial Services products, the Oracle's Siebel operational applications security model has been extended in the following ways:

■ **Finance 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 Finance 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 Finance 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.

Parent and Child Group Behavior

Oracle Business Intelligence supports hierarchies in the repository groups, and certain groups within the Analytics 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. Finance 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 114](#).

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 Finance 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 114 shows the additional responsibilities, and responsibility-specific security in Oracle's Finance 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 116 on page 474.

Table 114. Financial Services Responsibility Required to View FS Dashboards

FS Responsibilities	Dashboards
Finance 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

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 115 shows Pharma Analytics Responsibilities and Functions.

Table 115. 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.

Table 115. Pharma Analytics Responsibilities and Functions

Responsibility	Use
PH EMEA Marketing Analytics User	Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas. Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.
PH EMEA Sales Analytics User	Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas. Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.
PH Executive Analytics Admin	Unrestricted access to all Pharma Analytics options with ZIP territories.
PH Marketing Analytics Administrator	Administrator privileges to Pharma ROI, Call Activity Profit & Loss Report, Pharma Promotional Effectiveness Subject Area, and Medical Education Effectiveness Subject Area.
PH Medical Education Analytics Admin	Administrator privileges to Medical Education Analytics option.
PH Medical Education Analytics User	Enables access to Medical Education Analytics option.
PH Disconnected Analytics Admin	Administrator privileges to the PH Disconnected Manager Analytics User and Sales Rep Analytics Dashboards.
PH Disconnected Analytics User	Enables the Pharma Disconnected Analytics Home Page. Allows access to Sales Rep Dashboard as part of the Sales Rep Analytics option.
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.

Table 115. Pharma Analytics Responsibilities and Functions

Responsibility	Use
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.

Usage Accelerator Analytics Security Settings

[Table 116](#) describes the additional security configurations that may be necessary and the particular responsibilities associated with the Siebel Usage Accelerator application dashboards.

Table 116. 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

Table 116. Usage Accelerator Responsibilities and Dashboards

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
Usage Accelerator–Sales Manager	No position-based Security	Score Card	Team Scorecard Individual Scorecard
		Action Plan	Account Coverage (Team) Contact Coverage (Team) Opportunity Coverage (Team) Financial Account Coverage (Team) —Financial Services only Account Completeness (Team) Contact Completeness (Team) Opportunity Updates (Team)
Usage Accelerator–Financial Services Sales Manager		Coverage	Account Coverage Account Coverage (Team) Contact Coverage Opportunity Coverage Financial Account Coverage—Financial Services only
		Completeness	Account Completeness Contact Completeness
		Opportunity Updates	Opportunity Updates
		User Adoption	Active Users Application Usage—excluded for Financial Services Application Usage—Financial Services only*

Table 116. Usage Accelerator Responsibilities and Dashboards

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
Usage Accelerator—Sales Executive	No position-based Security	Scorecard	Organization Scorecard Individual Scorecard
Usage Accelerator—Financial Services Sales Executive		Action Plan	Account Coverage (Org) Contact Coverage (Org) Opportunity Coverage (Org) Financial Account Coverage (Org)—Financial Services only Account Completeness (Org) Contact Completeness (Org) Opportunity Updates (Org)
Usage Accelerator—Administrator		Coverage	Account Coverage Contact Coverage Opportunity Coverage Financial Account Coverage—Financial Services only
Usage Accelerator—Financial Services Administrator		Completeness	Account Completeness Contact Completeness
		Opportunity Updates	Opportunity Updates
		User Adoption	Active Users Application Usage (excluded for Financial Services) Application Usage—Financial Services only Although the Application Usage dashboard names appears similar, the Application Usage—Financial Services-only version of this dashboard is different from the other dashboard.

Implementing Security With Oracle EBS

This section explains how security in Oracle BI Applications is deployed with Oracle EBS.

Background

Data Security is a must-have, when exposing critical data to the end-users for creating ad-hoc requests. The Object Security would ensure that the user is shown only those Subject Areas (or content) using which, a user can pull in certain columns and create an ad-hoc request. However, it is possible in a business scenario, that the user should still see only data pertaining to his visibility in the organization. For 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 of 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 (for example, HCM, SCM, FII, CRM).

This release of Oracle BI Applications supports the following types of data security:

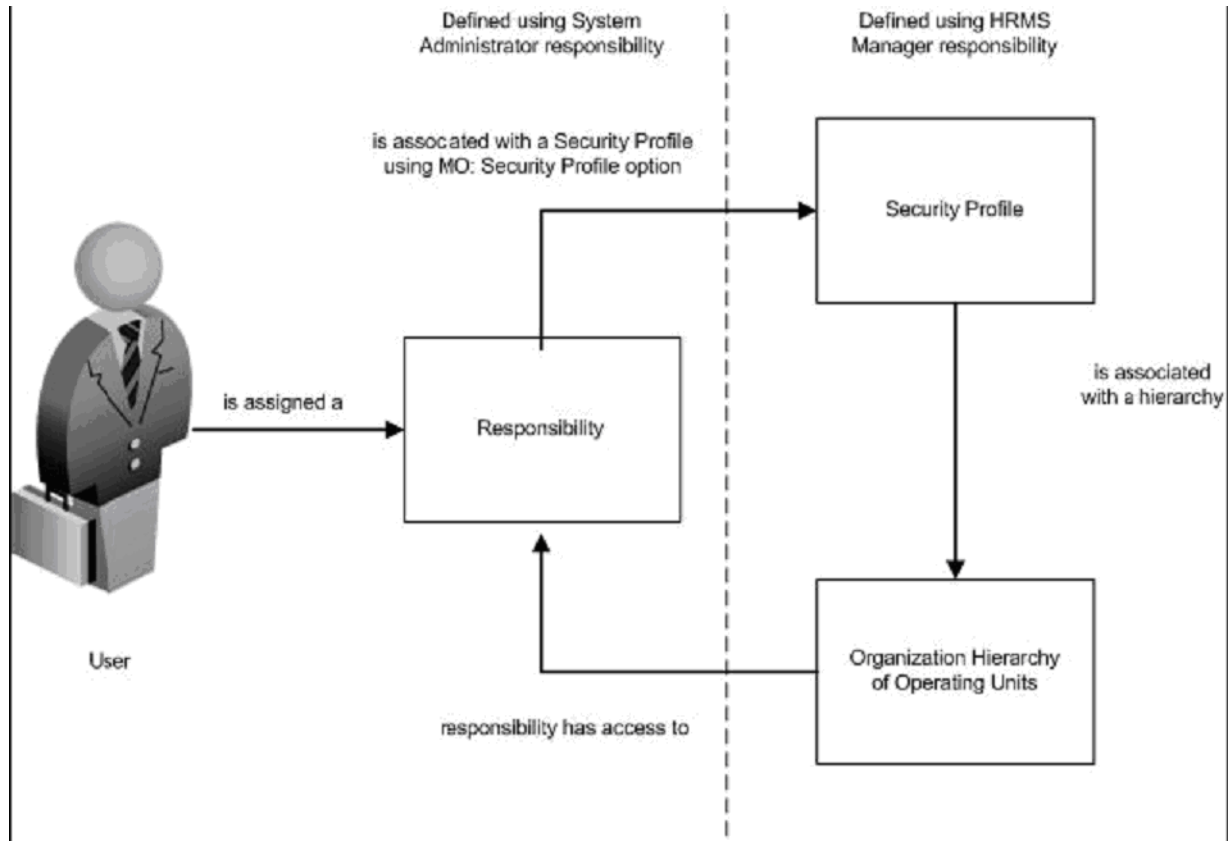
- Operating Unit-based security against Oracle EBS.
- Inventory Organization-based security against Oracle EBS.
- Position-based security against Siebel Applications.

This release of Oracle BI Applications does not support position-based security against Oracle EBS, or set of book, business group, GL flex field, or HR security profile-based security.

About 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 (see figure below).

Figure 35. 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 which 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), then we look at the profile, MO: Operating Unit. The only difference is that while MO: Operating Unit can be assigned only 1 OU, MO: Security Profile basically contains a Security Profile, which has access to 1 or more 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.

Implementation Steps For Unit-Based Security With Oracle EBS

The implementation steps are as follows:

- Oracle BI Applications uses LDAP authentication, and the following session variable is set automatically.

- System variable: USER -->

- Oracle BI Applications gets the operating unit set corresponding to the USER, from FND_USER_RESP_GROUPS. The following session variable is set automatically:

- System variable: OU_ORG -->

For more detailed information about this initialization block, see [Operating Unit Org Initialization Block on page 484](#).

- The security group hosts the collection of all related data access permissions. During implementation, customers have to decide which users or other groups to apply this security and add the users or other groups as members of the group.

- Security group: Operating Unit Org-based Security -->

- The permissions enforce the data security on the logical tables that have a relationship with operating units (for a list of logical tables, see [Table 117](#) below).

- When users creates ad-hoc reports, they will see the data that is assigned with these permissions. For reports involved with the logical tables (for a list of logical tables, see [Table 117](#) below), users are restricted to data pertaining to their visibility in the organization structure of the operating unit.

Table 117. Logical tables secured

Logical Table Names
"Core"."Fact - Sales - Backlog Lines"
"Core"."Fact - Sales - Booking Lines"
"Core"."Fact - Sales - Invoice Line"
"Core"."Fact - Sales - Backlog History"
"Core"."Fact - Sales - Pick Line"
"Core"."Fact - Sales - Order Line"
"Core"."Fact - Sales - Schedule Line"
"Core"."Fact - Sales - Cycle Line"
"Core"."Fact - Purchasing - Cost"
"Core"."Fact - Purchasing - Cycle Line"

Table 117. Logical tables secured

Logical Table Names
"Core"."Fact - Purchasing - Order Item"
"Core"."Fact - Purchasing - Receipt"
"Core"."Fact - Purchasing - Requisition"
"Core"."Fact - Purchasing - Request Status"
"Core"."Fact - Purchasing - Schedule"
"Core"."Fact - Fins - AP Employee Expense"
"Core"."Fact - Sales - Customer Status History"
"Core"."Dim - Operating Unit Org"
"Core"."Fact - Fins - AR Transaction"
"Core"."Fact - Fins - AP Transaction"
"Core"."Fact - Fins - (Internal) AP Current Closing Balance"
"Core"."Fact - Fins - (Internal) AP Balance Delta"
"Core"."Fact - Fins - (Internal) AR Current Closing Balance"
"Core"."Fact - Fins - (Internal) AR Balance Delta"
"Core"."Fact - Fins - AP Posted Transaction"
"Core"."Fact - Fins - AR Posted Transaction"
"Core"."Fact - Fins - AR Aging"
"Core"."Fact - Fins - AP Aging"

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.

Implementation Steps For Inventory Organization-Based Security With Oracle EBS

The implementation steps are as follows:

- Oracle BI Applications uses LDAP authentication, and the following session variable is set automatically.
 - System variable: USER -->
- Oracle BI Applications gets the operating unit set corresponding to the USER, from FND_USER_RESP_GROUPS. The following session variable is set automatically:
 - Row-wise variable: INV_ORG -->

For more detailed information about this initialization block, see [Inventory Organizations Initialization Block on page 484](#).
- The security group hosts the collection of all related data access permissions. During implementation, customers have to decide which users or other groups to apply this security and add the users or other groups as members of the group.
 - Security group: Inventory Org-based Security -->
- The permissions enforce the data security on the logical tables that have a relationship with operating units (for a list of logical tables, see [Table 118](#) below).

When users create ad-hoc reports, they will see the data that is assigned with these permissions. For reports involved with the logical tables (for a list of logical tables, see [Table 118](#) below), users are restricted to data pertaining to their visibility in the organization structure of the operating unit.

Table 118. Logical tables secured

Logical Table Names
"Core"."Fact - Sales - Backlog Lines"
"Core"."Fact - Sales - Booking Lines"
"Core"."Fact - Sales - Invoice Line"
"Core"."Fact - Sales - Backlog History"
"Core"."Fact - Sales - Pick Line"
"Core"."Fact - Sales - Order Line"

Table 118. Logical tables secured

Logical Table Names
"Core"."Fact - Sales - Schedule Line"
"Core"."Fact - Sales - Cycle Line"
"Core"."Fact - Purchasing - Cost"
"Core"."Fact - Purchasing - Cycle Line"
"Core"."Fact - Purchasing - Order Item"
"Core"."Fact - Purchasing - Receipt"
"Core"."Fact - Purchasing - Requisition"
"Core"."Fact - Purchasing - Request Status"
"Core"."Fact - Purchasing - Schedule"
"Core"."Fact - Supply Chain - Inventory Balance"
"Core"."Fact - Supply Chain - Inventory Transaction"
"Core"."Fact - Sales - Customer Status History"
"Core"."Fact - Supply Chain - Compound Inventory"
"Core"."Dim - Inventory Org"

About the Relationship Between Object Security and Data Security

Users can be defined as a member of one or more of Security Groups to enforce the corresponding data security or object security. However, a more complex security model can be implemented by adding one group as a child group of another. For example, a Data Security Group can be added under an Object Security Group. As a result, a user needs only to be added to the Security Group on the lowest level to inherit the union of all Securities Groups above.

Initialization Blocks

This section describes the initialization blocks available in Oracle BI Applications.

Authorization Initialization Block

This initialization block sets values for the session variable GROUP. Out of the box, the SQL is tuned for Oracle's Siebel Applications, and the connection pool is Siebel OLTP.

```

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

```

To make it applicable for Oracle EBS, the administrator has to change the connection pool to Oracle EBS OLTP and change the SQL to the following:

```

select DISTINCT 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')

```

Operating Unit Org Initialization Block

The initialization block sets the value for the 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'))))WHERE
PER_ORGANIZATION_LIST.SECURITY_PROFILE_ID = PROFILE_IDUNIONSELECT 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')
```

Inventory Organizations Initialization Block

The initialization block sets the value for the 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 TO_DATE(FND_USER_RESP_GROUPS.END_DATE) END) >= SYSDATE AND
FND_USER_RESP_GROUPS.USER_ID = (SELECT USER_ID FROM FND_USER WHERE USER_NAME
= ':USER')
```

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:

- [What Are Oracle's Siebel Operational Applications? on page 485](#)
- [Updating Finance Analytics Logical Table Sources on page 486](#)
- [Externalizing Financial Services Metadata Translation Strings on page 487](#)
- [Disabling the Finance Analytics Logical Table Sources on page 488](#)
- [Developing and Deploying Predictive Scores on page 489](#)
- [Analytics Metadata Requirements for Oracle's Siebel Industry Applications on page 489](#)
- [Oracle's Siebel Consumer Sector Dashboards and Pages on page 490](#)
- [Oracle's Siebel Consumer Sector Data Requirements on page 492](#)
- [Oracle's CME family of products \(Oracle Communications, Media and Energy Sales Analytics, on page 492](#)
- [Oracle's Siebel CME Dashboards and Pages on page 494](#)
- [Oracle's Siebel CME Data Requirements on page 496](#)
- [Oracle Pharma Sales Analytics Dimensions on page 496](#)
- [Dimensions Specific to Subject Areas in Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics on page 498](#)

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.

Updating Finance Analytics Logical Table Sources

In the Oracle BI repository file, logical table sources are set by default to settings in Oracle's Finance 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 Finance 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.

NOTE: Before performing the following procedure, shut down the Oracle BI Server.

To update Logical Table sources for Finance Analytics

- 1** Launch Oracle BI Administration Tool and open the Analytics Repository (OracleBIAnalyticsApps.rpd).

Go to the Business Model and Mapping window (the logical layer window) and open the Core folder.

Scroll down to the Fact - CRM - Asset logical table and open its Sources folder.

- 2** In the list of logical table sources, right-click Fact_ASSET_F_FINS.
Select Properties.

- 3 Click the General tab in the Properties window and make sure that the Active check box is checked.
If it is not, check it.
- 4 In the list of logical table sources, right-click W_ASSET_F.
Select Properties.
- 5 Click the General tab in the Properties window.
Make sure the Active check box is unchecked. If it is not, uncheck it.
Click OK and save the repository.
- 6 Restart Oracle BI Server.

Externalizing Financial Services Metadata Translation Strings

The Financial Services applications use a different set of translation strings from other Siebel operational applications.

You must externalize the metadata strings in the Analytics repository.

To externalize metadata strings in the Analytics repository

- 1 Open the OracleBIAnalyticsApps.rpd using the Oracle BI Administration Tool.
- 2 Select the entire presentation layer and right-click the mouse to display the menu.
- 3 Choose Display Names.
With this configuration, all metadata strings are read from an external Siebel operational application database, specifically from the table W_LOCALIZED_STRING_G.
- 4 Make sure that the connection pool Externalized Metadata Strings points to the Siebel operational application database and is working correctly.
- 5 Go to the Manage Variables menu and locate the initialization block External Metadata Strings.
Double-click on the initialization block to open the Edit window.
- 6 In the Initialization SQL area, change the SQL:

```
from
  select MSG_NUM, MSG_TEXT
  from VALUEOF(OLAP_TBO).W_LOCALIZED_STRING_G
  where MSG_TYPE = 'Metadata' and
  LANG_ID = decode('VALUEOF(NQ_SESSION.WEBLANGUAGE)'... [more]
to
```

```
select MSG_NUM, MSG_TEXT
from VALUEOF(OLAP_TBO).W_LOCALIZED_STRING_G
where MSG_TYPE = 'FINS_Metadata' and
LANG_ID = decode('VALUEOF(NQ_SESSION.WEBLANGUAGE)'... [more]
```

Click OK to commit the changes and save the repository.

- 7 Restart the Oracle BI Server.

Disabling the Finance Analytics Logical Table Sources

In the Analytics Repository file, logical table sources are set by default to settings in Oracle's Siebel Finance Analytics, the Healthcare Analytics family of products (Oracle Healthcare Sales Analytics, Oracle Healthcare Service Analytics, Oracle Healthcare Marketing Analytics, Oracle Healthcare Partner Manager Analytics), and the Oracle Insurance Analytics family of products (Oracle Insurance Partner Manager Analytics, Oracle Insurance Sales Analytics, Oracle Insurance Service Analytics, Oracle Insurance Marketing Analytics, Oracle Insurance Partner Manager Analytics). If you are using any Siebel Industry application that is *not* Oracle's Siebel Financial Services Analytics, you must first update the logical table sources in the Analytics Repository file.

These logical table sources must be deactivated in order for your Siebel Industry application Analytics reports to point to the correct logical model and retrieve the correct data. Do this by deactivating the FINS sources for the logical tables in the Core subject area and activating the other logical sources, as shown in the following procedure.

NOTE: Before performing the following procedure, shut down the Oracle BI Server.

To update Logical Table sources for Oracle's Siebel Industry Applications Analytics

- 1 Launch Oracle BI Administration Tool and open the Analytics Repository (OracleBIAnalyticsApps.rpd).
Go to Business Model and Mapping window (the logical layer window) and open the Core folder.
Scroll down to the Fact - CMR - Asset logical table and open its Sources folder.
- 2 In the list of logical table sources, right-click Fact_W_ASSET_F_FINS.
Select Properties.
- 3 Click the General tab in the Properties window and uncheck the Active check box.
- 4 In the list of logical table sources, right-click W_ASSET_F .
Select Properties.
- 5 Click the General tab in the Properties window.
Make sure that the Active check box is checked. If it is not, check it.
Click OK and save the repository.
- 6 Restart Oracle BI Server.

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 transactional database.

Once this is completed, scores may be viewed in the Siebel operational application by accessing the Accounts > Profiles > 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.

Analytics Metadata Requirements for Oracle's Siebel Industry Applications

Some metadata needs to be set up properly in the transactional database 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:

- [Oracle's Siebel Consumer Sector Dashboards and Pages on page 490](#)
- [Oracle's Siebel Consumer Sector Data Requirements on page 492](#)
- [Oracle's Siebel CME Dashboards and Pages on page 494](#)

■ [Oracle's Siebel CME Data Requirements on page 496](#)

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 119](#).

Table 119. 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.
	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.

Table 119. Consumer Sector Dashboards and Page Tabs

Dashboard	Page	Function
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?

Oracle's Siebel Consumer Sector Data Requirements

The data requirements for the Consumer Sector–specific portion of the data model are detailed in [Table 120](#). This includes the fund, promotion, performance and retail audit schema.

Table 120. 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 > SVP Category Details View > 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 transactional database 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 transactional database user interface and does not have to be the same as the activity date.

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 transactional database, 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.

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 121](#). These responsibilities also determine what subject areas the user may access.

Table 121. 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.

Table 121. Communications, Media and Energy Dashboards and Page Tabs

Dashboard	Page	Function
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.

Oracle's Siebel CME Data Requirements

The data requirements for each of the Communications, Media and Energy dashboards are detailed in [Table 122](#).

Table 122. 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 Developing and Deploying Predictive Scores on page 489 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.

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*.

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.

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 123](#).

Table 123. 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.

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 124](#).

Table 124. 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.

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 125](#).

Table 125. Product Costs For Customer Products

Product Type	Field to be Populated
Sample	Sample Cost
Detail	Avg. Promo Cost
Promotional Item Cost	Sample Cost

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*.

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.

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 126](#) lists the categories that need to have the Usage Type field populated in the Catalog Admin Category Detail list.

Table 126. 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

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 transactional database, 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 transactional database to the Oracle Business Analytics Warehouse.

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 transactional database 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.

Pharma Objectives Achievement

This subject is used to measure achievement and results for pharma call activity and Rx/sales targets. It is based on Pharma Objectives.

Objectives need to have a Unit populated in Retail Objective Form. Actual target numbers per contact and account need to be populated in the Pharma Campaign Target Account List or the Pharma Campaign Target Professional List Toggle.

Oracle Business Analytics Warehouse for Life Sciences Data Considerations

This chapter contains the following topics:

- [What's New for Pharma Analytics in Version 7.9 on page 501](#)
- [Oracle Business Analytics Warehouse for Life Sciences Overview on page 501](#)
- [Importing Syndicated Data into Oracle Business Analytics Warehouse on page 503](#)
- [Syndicated Data Flat File Formats on page 510](#)
- [Life Sciences Data Loading Issues with Oracle Business Analytics Warehouse on page 535](#)
- [Incremental Updates in the Oracle Business Analytics Warehouse LS Dimension Tables on page 536](#)

What's New for Pharma Analytics in Version 7.9

This section describes the new features for Pharma Analytics, Version 7.9.

- New support for loading Syndicated Data with multiple Alignment Types.
Syndicated data can now be loaded with different Alignment Types to support the requirements of each Syndicated data type being loaded.
- Location of Alignment Version flag in DAC has been changed.
 - The creation of historical Alignment Dimension hierarchy versions is controlled using a flag set within the DAC console. In Pharma Analytics 7.9 the location of this Alignment Version flag has moved to the Source System Parameters screen within DAC's Design area.

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 preaggregated 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 used only for contact primary address.

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:

- [Syndicated Loading Definitions on page 503](#)
- [Data Types Supported in the Oracle Business Analytics Warehouse on page 504](#)
- [Loading Syndicated Data into the Oracle Business Analytics Warehouse on page 505](#)

Syndicated Loading Definitions

[Table 127](#) provides terms and definitions related to syndicated loading.

Table 127. 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.

Table 127. Syndicated Loading Definitions

Term	Definition
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).

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 128](#).

Table 128. 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.
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.

Table 128. Data Types Supported in Oracle Business Analytics Warehouse

Data Type	Description
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.

Loading Syndicated Data into the Oracle Business Analytics Warehouse

The following options are available for loading syndicated data into the Oracle Business Analytics Warehouse:

- [Loading From a Flat File Source Using Syndicated Data Staging Tables and Cross-Referencing Tables on page 505](#)
- [Loading From a Flat File Source Using Syndicated Market Staging Tables on page 510](#)

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 130](#) to [Table 141](#) in section [Syndicated Data Flat File Formats on page 510](#).

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 129. 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
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.

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_SYNDX_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_SYNDX_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_SYNDX_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.

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 130](#) to [Table 141](#) in section [Syndicated Data Flat File Formats on page 510](#).

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 > 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.

Syndicated Data Flat File Formats

This section includes descriptions of the following flat files used in the syndicated data load process:

- [Syndicated Data Flat File for Direct Sales Data on page 511](#)
- [Syndicated Data Flat File for Indirect Sales Data on page 512](#)

- Syndicated Data Flat File for Prescription Data on page 514
- Syndicated Data Flat File for Plan Level Rx Data on page 516
- Syndicated Data Flat File for Physician Plan Level Rx Data on page 518
- Syndicated Data Flat File for Weekly Early View Data on page 520
- Syndicated Market Flat File for Indirect Sales Data on page 522
- Syndicated Market Flat File for Prescription Data on page 524
- Syndicated Market Flat File for Plan Level Rx Data on page 526
- Syndicated Market Flat File for Physician Plan Level Rx Data on page 527
- Syndicated Market Flat File for Weekly Early View Data on page 530
- Flat File Cross-Reference Data Population Rules on page 531

Syndicated Data Flat File for Direct Sales Data

Table 130 provides information about the syndicated data flat file for Direct Sales data.

Table 130. 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.

Table 130. Syndicated Data Flat File - Direct Sales Data

Field Name	Data Type	Field Required	Field Description
ACCNT_CITY	VARCHAR2(50)		Account city. Initial capital format; for example, Short Hills.
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)		

Syndicated Data Flat File for Indirect Sales Data

Table 131 provides information about the syndicated data flat file for Indirect Sales data.

Table 131. 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)
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.

Table 131. Syndicated Data Flat File - Indirect Sales Data

Field Name	Data Type	Field Required	Field Description
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.
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)		

Syndicated Data Flat File for Prescription Data

Table 132 provides information about the syndicated data flat file for Prescription data.

Table 132. 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.
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 132. Syndicated Data Flat File - Prescription Data

Field Name	Data Type	Field Required	Field Description
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)		

Syndicated Data Flat File for Plan Level Rx Data

Table 133 provides information about the syndicated data flat file for Plan Level Rx data.

Table 133. 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.

Table 133. Syndicated Data Flat File - Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
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)		

Syndicated Data Flat File for Physician Plan Level Rx Data

Table 134 provides information about the syndicated data flat file for Physician Plan Level Rx data.

Table 134. 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.

Table 134. Syndicated Data Flat File - Physician Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
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.
PERIOD_CD	VARCHAR2(30)		Enter the following Siebel period code: Month

Table 134. Syndicated Data Flat File - Physician Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
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)		

Syndicated Data Flat File for Weekly Early View Data

Table 135 provides information about the syndicated data flat file for Weekly Early View data.

Table 135. 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.

Table 135. Syndicated Data Flat File - Weekly Early View Data

Field Name	Data Type	Field Required	Field Description
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 135. 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)		

Syndicated Market Flat File for Indirect Sales Data

Table 136 provides information about the syndicated market flat file for Indirect Sales data.

Table 136. 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.

Table 136. Syndicated Market Flat File - Indirect Sales Data

Field Name	Data Type	Field Required	Field Description
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.
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.

Table 136. Syndicated Market Flat File - Indirect Sales Data

Field Name	Data Type	Field Required	Field Description
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)		

Syndicated Market Flat File for Prescription Data

Table 137 provides information about the syndicated market flat file for Prescription data.

Table 137. 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.

Table 137. Syndicated Market Flat File - Prescription Data

Field Name	Data Type	Field Required	Field Description
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.
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.

Table 137. Syndicated Market Flat File - Prescription Data

Field Name	Data Type	Field Required	Field Description
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)		

Syndicated Market Flat File for Plan Level Rx Data

Table 138 provides information about the syndicated market flat file for Plan Level Rx data.

Table 138. 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 138. 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 Rx of 20050101 data.
MKT_NRX_M04 - MKT_NRX_M26	NUMBER(22,7)		
MKT_TRX_M01	NUMBER(22,7)		Stores Total Rx of current month data (for example, 20050301 data).
MKT_TRX_M02	NUMBER(22,7)		Stores Total Rx of previous month data (for example, 20050301 data).
MKT_TRX_M03	NUMBER(22,7)		Stores Total Rx of 20050101 data.
MKT_TRX_M04 - MKT_TRX_M26	NUMBER(22,7)		

Syndicated Market Flat File for Physician Plan Level Rx Data

Table 139 provides information about the syndicated market flat file for Physician Plan Level Rx data.

Table 139. 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.
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 139. Syndicated Market 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.
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.

Table 139. Syndicated Market Flat File - Physician Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
MKT_TRX_M03	NUMBER(22,7)		Stores Total Rx of 20050101 data.
MKT_TRX_M04 - MKT_TRX_M26	NUMBER(22,7)		

Syndicated Market Flat File for Weekly Early View Data

Table 140 provides information about the syndicated market flat file for Weekly Early View data.

Table 140. 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.
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.

Table 140. Syndicated Market Flat File - Weekly Early View Data

Field Name	Data Type	Field Required	Field Description
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)		

Flat File Cross-Reference Data Population Rules

Table 141 provides information about flat file cross-reference data population rules for the following data types: Area (Brick), Account and Plan, Payer Type, and Contact.

Table 141. Flat File Cross-Reference Data Population Rules

Field Name	Data Type	Field Required	Field Description
Area (Brick) Data			
EXT_DATASRC_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.

Table 141. Flat File Cross-Reference Data Population Rules

Field Name	Data Type	Field Required	Field Description
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.
Account and Plan Data			
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 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.

Table 141. Flat File Cross-Reference Data Population Rules

Field Name	Data Type	Field Required	Field Description
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.
Payer Type Data			
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.
Contact Data			
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 141. Flat File Cross-Reference Data Population Rules

Field Name	Data Type	Field Required	Field Description
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.
Product and Market Data			
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.

Life Sciences Data Loading Issues with Oracle Business Analytics Warehouse

This issue is specific to Analytics for Life Sciences and does not affect other products.

The ETL process updates the Oracle Business Analytics Warehouse for Life Sciences with a full refresh or using incremental updates. In the DAC, you can run a full load by selecting Tools > ETL Management > 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".

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.

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.

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.

W_ALIGNVER_DH

Slowly Changing Dimension Type.

Attribute. Historical Zip code, Brick, or Account/Contact Hierarchy depends on what alignment item type is selected.

NOTE: In release 7.9, all Zip Code, Brick and Account/Contact hierarchy alignment can be loaded.

Description. Tracks historical alignments. Note the following:

- Depending on selected alignment type in AlignmentType.csv, only the selected alignment rules are extracted and loaded into the Oracle Business Analytics Warehouse. You can change the alignment type only when you run a full refresh ETL.
- NOTE:** In release 7.9, all Zip Code, Brick and Account/Contact alignment types can be extracted and loaded.
- When the alignment rule is loaded into the Oracle Business Analytics Warehouse, the current alignment rule is stored in the W_ALIGNMT_DH base table and the existing historical alignment rules are stored in W_ALIGNVER_DH table.

- The creation of Alignment Versions is controlled through the setting of the Alignment Version flag in DAC console. In Pharma Analytics 7.9 this flag is now set in the Source System Parameters screen within DAC's Design area. The parameter name is "\$\$ALIGNMENT_VER" and takes a value of either Y or N.
 - If the \$\$ALIGNMENT_VER parameter is set to N, then the ETL makes changes to existing alignments if there is a change on an assignment rule.
 - If the \$\$ALIGNMENT_VER parameter is set to Y, it creates a new alignment version. Any existing alignment moves to the W_ALIGNVER_DH table as history. The W_ALIGNVER_DH table is used only when the \$\$ALIGNMENT_VER parameter is set to Y.

The dimension has these characteristics:

- The first alignment after Full load is 1.
- Incremental Updates in the Oracle Business Analytics Warehouse LS Dimension Tables
- A new version is created when a new assignment rule (new position or position relationship) or an assignment criteria is modified (change in postal code, brick, contact, or account) if the Alignment Version flag is set to Y.
- Assignment criteria:
 - Contact ZIP Code or Account ZIP Code cannot use ZIP Code ranges. Each ZIP Code assigned to a territory needs to be on a separate row, so the same value needs to be entered for both ZIP Code Low and ZIP Code High. For example, if ZIP Code 09654 is assigned to a territory, the value for both ZIP Code High and ZIP Code Low should be 09654. Also, the same ZIP Code should not be assigned twice to the same territory and a ZIP Code should be unique to a territory when using assignment criteria for Contact ZIP Code or Account ZIP Code.
 - Contact Brick or Account Brick require unique bricks assigned to a territory.
 - Every new version increases the counter by one and is a sequential number. Any pre-existing history version alignment data moves to W_ALIGNVER_DH historical alignment table whenever a new version is created. That is, the W_ALIGNMT_DH table always maintains the latest version while the W_ALIGNVER_DH table maintains the rest of the historical version if the \$\$ALIGNMENT_VER parameter is set to Y. So, when the historical alignment dimension table gets too large, your administrator can create a SQL query to delete all attributes for version numbers from the W_ALIGNVER_DH table.
 - Effective date of the alignment is assignment activation date.



Pharma Disconnected Analytics Administration - Supplemental Information

This appendix describes additional configuration steps for Oracle's Pharma Disconnected Analytics, and contains the following sections:

- [About Pharma Disconnected Analytics](#)
- [Sourcing Reports for Oracle's Pharma Disconnected Analytics](#)

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*.

Sourcing Reports for Oracle's Pharma Disconnected Analytics

Table 142 on page 541 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 142. 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
Pharma Promotional Effectiveness	Contact Call Activity Aggregate	W_CON_CALL_N_A

Table 142. Sourcing Reports for Pharma Disconnected Analytics

Subject Area	Sourcing Report Name	Target Table
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.

Index

A

action links

- about 432
- testing 432

alignments dimension, configuring 497

ANSI NULL option, setting 45

Applications

- Oracle BI Applications Consumer 337
- Oracle Contact Center Telephony Analytics 263
- Oracle Financial Analytics 315
- Supply Chain Analytics 283

attributes

- historically significant attributes, identifying 223

B

browsers

- settings, configuring 434

bulkload mode, error while creating a custom session 405

C

circular hierarchy exceptions, about and example 395

CME Analytics applications

- data requirements 496
- security settings 468

CME analytics applications

- dashboards and pages 494
- data requirements 496
- order management analytics, about 493
- predictive scores, developing and deploying 489

code lookup

- about 234
- code mappings 235
- codes mapplets 235
- IA_CODES table 235
- sessions for lookups, configuring 236

complex products defined 493

Configuring

- fiscal calendar 127

configuring

- DAC client 71, 89
- DAC config.bat 89, 137

- DAC metadata repository 95
- email recipients in DAC client 110
- Oracle Business Analytics Warehouse database 104
- SSE role 105

conformed dimensions

- universal source, configuring for 245

consumer sector

- See Oracle's Siebel Consumer Sector Analytics applications

Cost List exceptions

- See also exception reports
- about and example 394
- Cost lists, using to display 398
- products without a cost list 395

cross-module configuration, performing

- codes lookup 234
- conformed dimensions, configuring 245
- dimension keys, resolving 237
- document, local, and group currencies, working with 227
- domain values, working with 238
- extracts, configuring 215
- loads, configuring 219
- Oracle 11i, configuring for 247
- records, filtering and deleting 219
- slowly changing dimensions, changing 222
- stored lookups, configuring 234

currencies

- configuring, about 227
- document, local, and group currencies, working with 227
- European Monetary Union currencies, handling 228
- exchange rate type, configuring 228
- exchange rates, extracting from a custom table 228

currency, diagnosing translation issues 398

D

DAC

- Source System Parameters 131

DAC config.bat

- configuring 89, 137

DAC ETL Preferences 131

DAC metadata repository

- configuring 95
- dashboards**
 - CME dashboards and pages 494
 - consumer sector dashboards and pages 490
 - customizing, about 435
 - exposing dashboards and requests 442
 - information, about viewing information 441
 - locating dashboard and page names 442
 - views, adding 438
- data flow**
 - Online Analytical Processing (OLAP) database, about and diagram 25
- data import**
 - syndicated data loading, overview 503
- data loading**
 - Oracle's Siebel Analytics for Life Sciences, issues when loading 535
 - troubleshooting, Informatica and data loading issues 403
- data warehouse**
 - architecture 24
 - database connections 100
 - overview 23
- Data Warehouse Application Console (DAC)**
 - about 26
 - configuring on UNIX 165
- Data Warehouse Application Console (DAC) client**
 - configuring 71, 89
- Data Warehouse Application Console (DAC) system properties**
 - setting up 96
- database connections**
 - transactional, creating 100
- database connections, creating**
 - DB2 installations, creating database connections for 85
 - Oracle installation, creating database connections for 85
- data-level security**
 - about 448
 - implementing 452
- DB Library Options setting, modifying 45**
- DB2**
 - database connections, creating for 85
 - Oracle Business Analytics Warehouse requirements 32
 - Oracle Business Analytics Warehouse, cannot connect to 402
- DB2 databases**
 - DB2CODEPAGE 64
- DB2CODEPAGE environment variable 64**

dimension keys

- dimension key resolution, using a lookup 237
- resolving 237

dimensions

- alignments dimension, configuring 497
- pharmaceuticals, configuring dimensions specific to 498
- positions dimension, configuring 496
- product costs, configuring 498
- products dimension, configuring 498

domain values

- domain value conversion process, understanding 238
- domain value set, configuring 244
- domain value set, extending 242
- importance, understanding 241
- working with 238

E**eBusiness Applications**

- See Siebel Operational Applications

email notification, Outlook error when

- sending post session email 405**

Enterprise Manager, error displayed after

- installing ODBC driver 402**

environment variables

- DB2CODEPAGE 64
- NLS_LANG 63
- setting on UNIX 162
- setting up on Windows 86

errors

- See troubleshooting

ETL History view, using to view the history of the ETL processes and

- statuses 398**

European Monetary Union currencies

- EXP_CURR_CONVERSION_TRANSFORM
 - Expression Transformation, using 230
- flat file, using to supply exchange rates 229
- handling 228

exception reports

- See also Cost List exceptions; system preferences
- additional exceptions, list of 398
- circular hierarchy exceptions, about and example 395
- cleansing the data, process and diagram 396
- ETL History view, using to view the history of the ETL processes and

- statuses 398
- exception report components, list of 393
- exception, fixing 397
- exchange rate exceptions, about 395
- Exchange Rate view, using to diagnose
 - currency translation issues 398
- executing the report, cleansing steps 396
- invalid hierarchy exceptions, about 395
- List of Values exceptions 393
- List of Values view, using to compare values
 - with Oracle Business Analytics Warehouse 397
- exchange rate exceptions, about** 395
- Exchange Rate view, using to diagnose currency issues** 398
- extracts, configuring**
 - about 215
 - extracting additional data, about 216
 - extracting data from new source table 216
 - extracting new data from existing source table 216

F

- filters for security groups, about** 454
- Finance Analytics application**
 - logical table sources, disabling 488
 - logical table sources, updating 486
 - metadata translation strings, externalizing 487
- Financial Analytics applications**
 - process of configuring Oracle 11i for 316
- Fiscal Calendar**
 - configuring 127
 - setting up 126
- fiscal month**
 - configuring 127
- fiscal week**
 - configuring 127
- Full_Extract_Siebel_DW, error returned by Informatica** 403
- Full_Load_Siebel_DW**
 - error when running 404

H

- Healthcare Analytics**
 - security settings 469
- hierarchical relationships, invalid hierarchy exceptions, about** 395
- historically significant attributes**
 - identifying 223
- home page**
 - content, customizing 435
 - customizing, about 435

- path argument, determining 437

Horizontal applications

- indexing for performance 67

I

Incentive Compensation Analytics, turning on 434

indexes

- to improve performance 67

Indexing

- S_.* tables 67

Informatica

- copying source files 79
 - installing patches on UNIX 161
 - license files, updating on UNIX 160
 - lookup files 79

Informatica client software

- Enable Data Code Pages Validation, checking if conditions met 88

Informatica patches installation 88

Informatica PowerCenter

- installation wizard 75

Informatica repository

- configuring 113
 - configuring in Workflow Manager 118
 - creating 113
 - dropping 121
 - restoring 113
 - verifying configuration 125

Informatica Repository Server

- installing on UNIX 151
 - registering 114

Informatica Server

- installing on other than default location 119
 - installing on UNIX 153
 - registering 110
 - setting up 79
 - starting 138
 - starting on UNIX 167
 - stopping on UNIX 168

Informatica Server and repository, configuring

- Unicode mode, configuring under Windows 87
 - Windows, setting up Informatica service under 138

Initialization Blocks 66

installation

- data warehouse installation and configuration process 28
 - database connectivity 90
 - Informatica patches 88

- Informatica PowerCenter 75
- Informatica Server, installing on other than default directory 119
- Java SDK 71, 89
- Java SDK and drivers 89
- JDBC drivers 90
- Oracle BI Applications installation process task list 69
- wizard, Oracle BI Applications 72
- installing**
 - Oracle BI repository, setting up 200
- Insurance Analytics family of products**
 - security settings 469
- IP address**
 - virtual IP address, creating 431
 - virtual IP address, testing 432
- J**
- Java SDK**
 - installation 71, 89
- JDBC drivers**
 - installation 90
- L**
- language packs**
 - installing 425
- licensing**
 - product licensing 19
- List of Values**
 - exceptions, about and example 393
 - view, using to compare values with the Oracle Business Analytics Warehouse 397
- loads, configuring**
 - about 219
 - Source Adapter mapplet. creating 218
- localization packs**
 - installing 425
- M**
- Mappings**
 - SA System Subject Area 65
- metadata object level security**
 - repository groups 458
 - Web Catalog groups 459
 - Web Catalog security 459
- metadata strings**
 - externalizing 423
- metadata, Siebel Industry applications**
 - Analytics applications, about metadata requirements 489
 - CME dashboards and pages 494
 - CME data requirements 496
 - consumer sector dashboards and pages 490
 - consumer sector data requirements 492
 - dimensions, configuring specific to pharmaceuticals 498
 - Finance Analytics applications, disabling logical table sources 488
 - Finance Analytics applications, externalizing metadata translation strings 487
 - Finance Analytics applications, updating logical table sources 486
 - Pharma medical education effectiveness, configuring subject area 499
 - Pharma objectives achievement, configuring subject area 500
 - Pharma product categories, configuring subject area 499
 - Pharma promotional effectiveness, configuring subject area 499
 - Pharma Sales Analytics, about metadata requirements 496
 - Pharma sales effectiveness, configuring subject area 498
 - positions dimension, configuring 496
 - predictive scores, developing and deploying 489
 - product costs dimension, configuring 498
 - products dimension, configuring 498
- metadata, Siebel Industry applications**
 - alignments dimension, configuring 497
- Microsoft Outlook**
 - closes when finishing ETL process, probable cause 406
 - error when sending post session email 405
- N**
- NLS_LANG environment variable 63**
- O**
- OLAP**
 - See Online Analytical Processing database
- Online Analytical Processing database**
 - about 30
 - Data Warehouse, data flow into 25
- Online Transaction Processing database**
 - Online Analytical Processing database, differences from 30
- OPB_OBJECT_LOCKS Informatica session deadlocked, probable cause 405**
- Oracle**
 - database connections, creating for 85
 - mappings hang, probable cause 403
 - Oracle Business Analytics Warehouse requirements 34

Oracle 11i

- Customers Dimension table, mapping source customer hierarchies to 248
- group currency code, configuring 247
- last extract date, modifying 252
- Oracle Flex fields, incorporating 247
- Order Management and Fulfillment, configuring for 338
- process of configuring Financial Analytics applications for 316
- process of configuring Supply Chain Analytics for 286
- product categories, configuring 250
- Source ID, modifying 251

Oracle BI

- repository, setting up 200

Oracle BI Applications

- initialization, completing 428
- installation and configuration process task list 69
- installation process 146
- installation wizard 72
- installing, UNIX 150
- UNIX, deploying 145

Oracle BI Applications, configuring

- action links, about 432
- action links, testing 432
- browser settings, configuring 434
- dashboards, adding views 438
- home page and dashboards, about customizing 435
- home page content, customizing 435
- home page, determining path argument 437
- Incentive Compensation Analytics, turning on 434
- optional Siebel Analytics applications, accessing 433
- Oracle BI Applications, completing initialization 428
- process 427
- Siebel operational applications, about integrating within 433
- Siebel Web Extension, configuring to work with 430
- virtual IP address, creating 431
- virtual IP address, testing 432

Oracle BI Charts

- metadata strings, externalizing 423
- Oracle BI seed data, updating for non-English locales 422

Oracle BI Presentation layer

- process of maintaining translation tables 421

Oracle BI repository

- externalizing metadata strings 423
- repository groups, security 458
- table joins, configuring 455

Oracle BI seed data

- updating for non-English locales 422

Oracle BI Supply Chain Analytics

- domain values and CSV worksheet files 313
- process of aggregating tables for 299
- process of configuring universal source for 306
- universal source, configuring expenses 305

Oracle Business Analytics Warehouse

- architecture 24
- architecture components 25
- database parameters and requirements 30
- database, configuring 104
- DB2 requirements 32
- deployment configuration 28
- operating system, driver, connectivity requirements 29
- Oracle Business Analytics Warehouse, differences from 30
- Oracle requirements 34
- overview 23
- SQL server-specific requirements 45
- tables, creating 105
- Teradata requirements 47

Oracle Contact Center Telephony Analytics

- Benchmarks and Targets table, setting up 272
- Contact Representative Snapshot table, setting up 271
- Contact Status Type columns, about configuring for 268
- dates and times, configuring 275
- dimension key resolution 264
- dimension table, loading 264
- Event Type columns, about configuring for 265
- fact table, loading 264
- overview 263
- Representative Activities table, setting up 269

Oracle databases

- NLS_LANG 63

Oracle Human Resources Analytics

- domain values and CSV worksheet files 379
- overview 363
- Payroll table, about aggregating 377
- Payroll table, aggregating 378

- Workforce Operations, about configuring for
 - Oracle 11i 364
- Workforce Operations, process of configuring
 - for Oracle 11i 364
- Workforce Payroll, process of configuring for
 - Oracle 11i 374
- Oracle Order Management and Fulfillment Analytics Option** 337
- Oracle Partner Analytics**
 - role-based analytics dashboard
 - mapping 460
- Oracle Pharma Sales Analytics applications**
 - security settings 472
- Oracle's Siebel Industry Applications**
 - Oracle Pharma Sales Analytics security
 - settings 472
 - Oracle's Siebel Financial Services security
 - settings 469
 - Siebel Usage Accelerator Analytics security
 - settings 474
- Oracle's Siebel Life Sciences**
 - data loading issues 535
- Oracle's Siebel Operational Applications**
 - Oracle Partner Analytics security
 - settings 460
- Oracle's Siebel Operational Applications, integration with**
 - Oracle's Siebel Industry 427
- order management analytics, about** 493
- Order Management and Fulfillment Analytics**
 - Oracle 11i, process of configuring for 338
- Outlook**
 - closes when finishing ETL process, probable
 - cause 406
 - error when sending post session email 405
- P**
- page names, locating and dashboards** 442
- parent-child relationship, circular hierarchy exceptions** 395
- PeopleSoft Trees** 336
- PerfIndex_Horizontal.sql** 67
- PerfIndex_Industry.sql** 67
- performance**
 - components configured to optimize
 - performance 46
 - implementing indexes to improve 67
- Pharma**
 - medical education effectiveness, configuring
 - subject area 499
 - objectives achievement, configuring
 - subject area 500
 - product categories, configuring
 - subject area 499
 - promotional effectiveness, configuring
 - subject area 499
 - sales effectiveness, configuring
 - subject area 498
- Pharma Sales Analytics**
 - alignments dimension, configuring 497
 - dimensions, configuring specific to
 - pharmaceuticals 498
 - metadata requirements, about 496
 - Pharma medical education effectiveness,
 - configuring subject area 499
 - Pharma objectives achievement, configuring
 - subject area 500
 - Pharma product categories, configuring
 - subject area 499
 - Pharma promotional effectiveness,
 - configuring subject area 499
 - Pharma Sales effectiveness, configuring
 - subject area 498
 - positions dimension, configuring 496
 - product costs dimension, configuring 498
 - products dimension, configuring 498
 - security settings 472
- positions dimension, configuring** 496
- predictive analytics metadata**
 - predictive scores, developing and
 - deploying 489
- Presentation Catalog, administering**
 - dashboards, about viewing
 - information 441
 - dashboards, exposing and requests 442
 - dashboards, locating and page names 442
- primary extract and delete mapping**
 - delete and primary extract sessions,
 - enabling 222
 - source-archived records, deletion
 - configuration 221
 - working with 221
- product costs dimension, configuring** 498
- product licensing** 19
- products dimension, configuring** 498
- Q**
- Query Analyzer, can not open after installing ODBC driver** 402
- R**
- RDBMS error message, probable cause** 404
- records**
 - source-archived records, deletion
 - configuration 221

records, filtering and deleting

- about 219
- primary extract and delete mappings, understanding 219
- primary extract and delete mappings, working with 221

Refresh_Extract_Siebel_DW

- error returned by Informatica 403

repository

- Oracle BI repository, setting up 200

requests

- exposing dashboards and requests 442

responsibilities

- access in Oracle BI 444
- integrated security, about 443
- user responsibilities, checking 444
- user responsibilities, registering new responsibility 445

S**S_.* tables**

- indexing 67

S_MSG table, adding custom translations

- 424

SA System Subject Area

- mapping 65

schema definitions

- transactional database, updating 138

SDEINC_RecordExtractStart session fails, probable cause

- 404

Security

- initialization blocks 66

security

- CME security settings 468
- consumer sector security settings 468
- data-level security, about 448
- data-level security, implementing 452
- default security settings 445
- filters for security groups, about 454
- Finance Analytics security settings 469
- metadata object level security (repository groups), about 458
- operational applications security 456, 460
- Oracle Partner Analytics security settings 460
- Oracle Pharma Sales Analytics security settings 472
- Oracle's Siebel Usage Accelerator Analytics security settings 474
- repository table joins, configuring 455
- responsibilities and access in Oracle Business Intelligence 444

- responsibilities, integrated security for 443

- security levels, about 443

- Siebel Sales Analytics security

- settings 456

- user responsibilities, checking 444

- user responsibilities, registering new

- responsibility 445

- Web Catalog security 459

seed data

- importing 422

server components, installation requirements

- See also server component software
- Oracle Business Analytics Warehouse, about 30

setting up

- DAC system properties 96

Siebel Analytics Charts

- local seed data, importing 422
- localization packs, installing 425
- S_MSG table, adding custom translations 424
- string translations, adding for metadata 424
- translation tables, process of maintaining 421
- Web catalog strings, about translating 425

Siebel Analytics repository

- repository groups, security 458

Siebel CME Analytics applications

- predictive scores, developing and deploying 489

Siebel Communications, Media, and Energy Analytics

- See Siebel CME Analytics applications

Siebel Consumer Sector Analytics applications

- consumer sector dashboards and pages 490
- consumer sector data requirements 492
- security settings 468

Siebel Enterprise Contact Center Analytics

- configuring flags for 276
- excluding representative data 277

Siebel Enterprise Sales Analytics

- currency and quantity grain 359
- multiple products, about tracking 353
- overview 337
- Sales Order Dates, about 360

Siebel Enterprise Workforce Analytics

- domain values and CSV worksheet files, configuring 384

Siebel Finance Analytics applications

- logical table sources, disabling 488

- logical table sources, updating 486
- metadata translation strings, externalizing 487
- security settings 469
- Siebel Financial Analytics**
 - overview, 315
 - PeopleSoft Trees 336
- Siebel Industry Applications**
 - alignments dimension, configuring 497
 - Analytics applications, about metadata requirements 489
 - CME dashboards and 494
 - CME data requirements 496
 - CME security settings 468
 - consumer sector dashboards and pages 490
 - consumer sector data requirements 492
 - consumer sector security settings 468
 - dimensions, configuring specific to pharmaceuticals 498
 - Finance Analytics applications, disabling logical table sources 488
 - Finance Analytics applications, externalizing metadata translation strings 487
 - Finance Analytics applications, updating logical table sources 486
 - Pharma medical education effectiveness, configuring subject area 499
 - Pharma objectives achievement, configuring subject area 500
 - Pharma product categories, configuring subject area 499
 - Pharma promotional effectiveness, configuring subject area 499
 - Pharma Sales Analytics, about metadata requirements 496
 - Pharma sales effectiveness, configuring subject area 498
 - positions dimension, configuring 496
 - predictive scores, developing and deploying 489
 - product costs dimension, configuring 498
 - products dimension, configuring 498
 - seed data, importing 427
- Siebel Operational Applications**
 - dashboards, adding views 438
 - home page content, customizing 435
 - home page path argument, determining 437
 - security 456, 460
 - Siebel Analytics, about integrating within 433
 - Siebel Sales Analytics security settings 456
- Siebel Operational Applications, integration with**
 - data types 427
- Siebel Pharma Sales Analytics applications**
 - dimensions, configuring specific to pharmaceuticals 498
 - metadata requirements, about 496
 - Pharma medical education effectiveness, configuring subject area 499
 - Pharma objectives achievement, configuring subject area 500
 - Pharma product categories, configuring subject area 499
 - Pharma promotional effectiveness, configuring subject area 499
 - Pharma Sales effectiveness, configuring subject area 498
 - positions dimension, configuring 496
 - product costs dimension, configuring 498
 - products dimension, configuring 498
- Siebel PRM Analytics**
 - data-level visibility 466
 - portal-based analytics dashboard mapping 463
 - subject area mapping 464
 - subject area visibility 465
- Siebel Sales Analytics**
 - responsibilities and data-level visibility 456
- Siebel Transaction Database**
 - connection error 402
 - Cost lists, displaying 398
- Siebel Transactional database**
 - importing locale seed data 422
- Siebel Web Extension**
 - configuring to work with 430
 - virtual IP address, creating 431
 - virtual IP address, testing 432
- SIF files**
 - verifying SIF patch 141
- simple products defined 493**
- slowly changing dimensions, configuring**
 - about 222
 - currencies, configuring 227
 - effective dates 225
 - historically significant attributes, identifying 223
 - Type I changing dimension 223
 - Type II slowly changing dimension 224
- Source Adapter mapplet**
 - creating 218
- Source System Parameters**
 - about 131
 - setting 131

source-archived records

deletion configuration 221

SQL Server

DB Library Options setting, modifying 45
 Oracle Business Analytics Warehouse requirements 45

SSE role

creating 105

stored lookups

configuring 234

stored procedures

creating 142

string translations

adding for metadata 424

Supply Chain Analytics

process of configuring Oracle 11i for 286

SWE

See Siebel Web Extension

syndicated data

importing, overview 503

system preferencesSee *also* exception reports**T****Teradata**

activating join indexes 104
 avoiding arithmetic overflows 47
 code page file property, setting 124
 configuration checklist, Windows 143
 creating and dropping tables 109
 data warehouse requirements 47
 database hints 47
 general guidelines 47
 Hosts file, setting up 124
 installation checklist for UNIX 169
 installing patches on UNIX 161
 setting up Hosts file on UNIX 163

translation tables

process of maintaining 421

troubleshooting

Informatica and data loading issues 403
 Oracle Business Analytics Warehouse installation 402
 Unicode 408, 415
 UNIX installation 409

U**Unicode**

deploying Informatica and Oracle Business Analytics Warehouse in 53
 local seed data, importing 422
 localization pack, installing 425
 metadata strings, externalizing 423

Oracle BI seed data, updating for non-English locales 422

S_MSG table, adding custom translations 424

string translations, adding for metadata 424

translation tables, process of maintaining 421

Web catalog strings, about translating 425

Unicode mode

Windows, configuring under 87

universal source

conformed dimensions, configuring 245

Oracle BI Supply Chain Analytics, configuring expenses 305

process of configuring Oracle BI Supply Chain Analytics for 306

UNIX

deploying Oracle BI Applications 145

updating license files 160

URL

dashboards, configuring for 438

home page, configuring for 435

Usage Accelerator Analytics security

settings 474

user responsibilities

checking 444

new responsibility, registering 445

V**Vertical applications**

indexing for performance 67

views, adding to dashboards 438**virtual IP address**

creating 431

testing 432

W**Web Catalog groups**

Siebel Analytics applications, security in 459

Web Catalog strings

about translating 425

Windows

Informatica Server, setting up under 138

Unicode mode, configuring under Windows 87

workflows

error message when executing on the Informatica Server Manager, probable cause 403

Server Manager error message, probable cause 403

