Oracle® Business Intelligence Applications

Installation Guide for Informatica PowerCenter Users

Version 7.9.6.1

E14843-01

October 2009

Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users Version 7.9.6.1

E14843-01

Contents

Part III Appendices

A Supporting Tasks for DAC and Informatica PowerCenter

B Localizing Oracle Business Intelligence Deployments

C Integrating Interactive Dashboards and Operational Applications Data

D Configuring Metadata for Oracle Business Intelligence Applications

E Using Oracle Business Analytics Warehouse Exception Reports

F About the Versioned Informatica Repository

Index

Preface

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

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

Oracle BI Applications includes the following:

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

For more details on the applications included in this release of Oracle BI Applications, see the *Oracle Business Intelligence Applications Licensing and Packaging Guide*. This guide is included in the Oracle Business Intelligence Media Pack. Also, see the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*, available at available at `http://www.oracle.com/technology/documentation/bi_apps.html`.

Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users contains instructions for installing and setting up Oracle BI Applications release 7.9.6.1. For a high-level road map of the steps required to install and set up Oracle BI Applications, see Section 2.4, "Roadmap to Installing and Setting Up Oracle BI Applications."

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

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

Audience

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

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

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

Deaf/Hard of Hearing Access to Oracle Support Services

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

Related Documents

For more information, see the following documents in the Oracle BI Applications release 7.9.6.1 documentation set (available at `http://www.oracle.com/technology/documentation/bi_apps.html`):

- *Oracle Business Intelligence Applications Release Notes*

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

Also see the Oracle Business Intelligence Data Warehouse Administration Console documentation set (available at `http://www.oracle.com/technology/documentation/bi_dac.html`):

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

Conventions

The following text conventions are used in this document:

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

1 What's New in This Release

This section lists changes described in this version of the documentation to support release 7.9.6.1 of the software.

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

1.1 What's New in Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users

This guide includes the following changes:

- Section 3.5, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse" was updated.
- Section 3.8, "Partitioning Guidelines For Large Fact Tables" was added.
- Section 4.5, "Running the Oracle BI Applications Installer on Windows" was updated for the current release.
- Section 4.6, "Installing Informatica PowerCenter" was updated for Informatica version 8.6.1.
- Additional quality enhancements were made to the guide.

Part I

Getting Started

Part I helps you get started with Oracle BI Applications. It contains the following sections:

- Chapter 2, "Overview of Oracle BI Applications"

> **Note:** For a high-level road map for installation and set up steps for Oracle BI Applications, see Section 2.4, "Roadmap to Installing and Setting Up Oracle BI Applications."

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

2 Overview of Oracle BI Applications

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

- Section 2.1, "What Is Oracle BI Applications?"
- Section 2.2, "Oracle Business Analytics Warehouse Overview"
- Section 2.3, "Oracle Business Analytics Warehouse Architecture"
- Section 2.4, "Roadmap to Installing and Setting Up Oracle BI Applications"
- Section 2.5, "Using Oracle BI Repository Documentation"

2.1 What Is Oracle BI Applications?

Oracle BI Applications is a prebuilt business intelligence solution.

Oracle BI Applications supports Oracle sources, such as Oracle E-Business Suite Applications, Oracle’s Siebel Applications, Oracle’s PeopleSoft Applications, Oracle’s JD Edwards Applications, and non-Oracle sources, such as SAP Applications. If you already own one of the above applications, you can purchase Oracle Business Intelligence Enterprise Edition and Oracle BI Applications to work with the application.

Oracle BI Applications also provides complete support for enterprise data, including financial, supply chain, workforce, and procurement and spend sources. These enterprise applications typically source from both Oracle data sources, such as Oracle EBS and PeopleSoft and non-Oracle data sources, such as SAP.

Oracle BI Applications consists of the components shown in Table 2–1.

Table 2–1 Oracle BI Applications Components (with Informatica/DAC)

Component	Description
DAC Repository metadata 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 (ETL) operations for the data warehouse.
Prebuilt Informatica content	This content includes ETL repository objects, such as mappings, sessions, and workflows, and is contained in the Informatica Repository file (Oracle_BI_DW_Base.rep).

Table 2–1 (Cont.) Oracle BI Applications Components (with Informatica/DAC)

Component	Description
Prebuilt metadata content	This metadata content is contained in the Oracle BI 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, see Section 2.2, "Oracle Business Analytics Warehouse Overview."

2.2 Oracle Business Analytics Warehouse Overview

The Oracle Business Analytics Warehouse is a unified data repository for all customer-centric data, which supports the analytical requirements of the supported source systems.

The Oracle Business Analytics Warehouse includes the following:

- A complete relational enterprise data warehouse data model with numerous prebuilt star schemas encompassing many conformed dimensions and several hundred fact tables.

  For more information about the data warehouse data model, see the *Oracle Business Analytics Warehouse Data Model Reference*.

- An open architecture to allow organizations to use third-party analytical tools in conjunction with the Oracle Business Analytics Warehouse using the Oracle Business Intelligence Server.
- Prebuilt data extractors to incorporate data from external applications into the Oracle Business Analytics Warehouse.
- A set of ETL (extract-transform-load) processes that takes data from a wide range of source systems and creates the Oracle Business Analytics Warehouse tables.
- The Oracle Business Intelligence Data Warehouse Administration Console (DAC), a centralized console for the set up, configuration, administration, loading, and monitoring of the Oracle Business Analytics Warehouse.

2.3 Oracle Business Analytics Warehouse Architecture

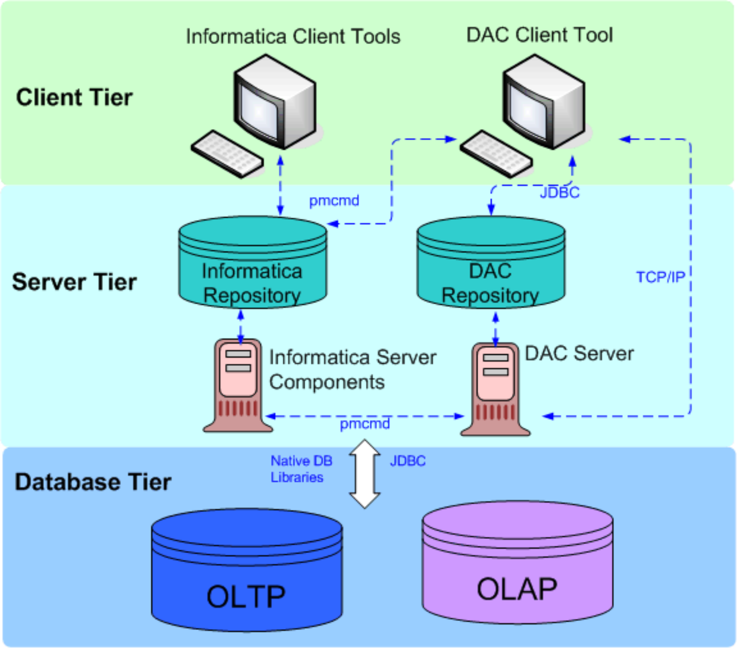
High-level analytical queries, like those commonly used in Oracle Business Intelligence, scan and analyze large volumes of data using complex formulas. This process can take a long time when querying a transactional database, which impacts overall system performance.

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

2.3.1 Oracle Business Analytics Warehouse Architecture Components

Figure 2–1 illustrates the Oracle Business Analytics Warehouse architecture when deployed with Informatica PowerCenter and DAC.

Figure 2–1 An example Oracle Business Analytics Warehouse architecture with Informatica PowerCenter and DAC



- The Client tier contains the Informatica PowerCenter client tools and the Oracle BI Data Warehouse Administration Console (DAC). DAC is a command and control interface for the data warehouse to allow for set up, configuration, administration, and monitoring of data warehouse processes.
- The Server tier contains the following:
  - DAC Server. Executes the instructions from the DAC Client. The DAC Server manages data warehouse processes, including scheduling, loading of the ETL, and configuring the subject areas to be loaded. It dynamically adjusts its actions based on information in the DAC Repository. Depending on your business needs, you might incrementally refresh the Oracle Business Analytics Warehouse once a day, once a week, once a month, or on another similar schedule.
  - DAC Repository. Stores the metadata (semantics of the Oracle Business Analytics Warehouse) that represents the data warehouse processes.
  - Informatica PowerCenter Services:
    - Integration Services - The Integration Service reads workflow information from the repository. The Integration Service connects to the repository through the Repository Service to fetch metadata from the repository.
    - Repository Services - The Repository Service manages connections to the PowerCenter Repository from client applications. The Repository Service is a separate, multi-threaded process that retrieves, inserts, and updates metadata in the repository database tables.
  - Informatica Repository. Stores the metadata related to Informatica workflows.

- The Database tier contains the OLTP and OLAP databases.

2.4 Roadmap to Installing and Setting Up Oracle BI Applications

To install and set up Oracle BI Applications, do the following:

- For the source systems that you are using, follow the appropriate preinstallation steps in Chapter 3, "Preinstallation and Predeployment Requirements for Oracle BI Applications."
- Install and set up the Oracle BI Applications components and Informatica PowerCenter components as described in Chapter 4, "Installing and Setting Up Oracle BI Applications."

  Note: For an example that shows a typical deployment topology for Oracle BI Applications, see Section 4.1, "About Oracle BI Applications Topologies."
- Perform the source-independent steps that are required before a full data load, described in the section entitled, "Configuring Common Areas and Dimensions," in *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*. Then, perform the source system-specific steps that are required before a full data load for the appropriate source system that you are using.
- Perform any required additional source-independent steps described in the section entitled, "Configuration Steps for Controlling Your Data Set for All Source System," in *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*. Then, perform any required additional steps for the appropriate source system that you are using.
- Perform the configuration steps that are required before a full data load for all applications that you are deploying. See *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users* for instructions on configuring the different application families.
- (Optional) If you want to customize the preconfigured Oracle BI Applications functionality, follow the steps described in the chapter entitled, "Customizing the Oracle Business Analytics Warehouse," in the *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*.
- (Optional) If you want to modify the preconfigured Oracle BI Applications security, see *Oracle Business Intelligence Applications Security Guide*.

Once you have installed and set up Oracle BI Applications components, configured the modules (optional), and customized Oracle BI Applications (optional), you are ready to start running ETL processes.

For a detailed example of how to run an ETL for Oracle Financials with an Oracle EBS OLTP data source, see Section 4.20, "About Running A Full Load ETL".

For information about running ETL processes, see *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

2.5 Using Oracle BI Repository Documentation

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

- Oracle BI Repository Documentation

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

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

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

- Presentation Services Catalog

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

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

Part II

Installation and Set Up Steps for Oracle BI Applications

Part II provides instructions for installing and setting up Oracle BI Applications. It contains the following sections:

- Chapter 3, "Preinstallation and Predeployment Requirements for Oracle BI Applications"
- Chapter 4, "Installing and Setting Up Oracle BI Applications"

> **Note:** For a high-level road map for installation, configuration, and customization steps for Oracle BI Applications, see Section 2.4, "Roadmap to Installing and Setting Up Oracle BI Applications."

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

3 Preinstallation and Predeployment Requirements for Oracle BI Applications

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

This section provides information about preparing to install and deploy Oracle BI Applications. You should review this information before you begin the installation and deployment process. You should also read the general guidelines for setting up the Oracle Business Analytics Warehouse and read the appropriate database-specific guidelines for the source OLTP databases that you are using.

You also need to satisfy the database and Informatica PowerCenter requirements that are specified in Section 4.3, "Mandatory Requirements."

Notes

- For information about database-specific settings, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.
- For information about code page settings, refer to the Informatica documentation, which is included on the Informatica PowerCenter DVD provided with Oracle BI Applications.

This section contains the following topics:

- Section 3.1, "General Guidelines for Setting Up Oracle Business Analytics Warehouse"
- Section 3.2, "IBM DB2 UDB-Specific Database Guidelines for Oracle Business Analytics Warehouse"
- Section 3.3, "IBM DB2 UDB zOS and OS/390 and z/OS-Specific Database Guidelines for Oracle Business Analytics Warehouse"

- Section 3.4, "SQL Server-Specific Database Guidelines for Oracle Business Analytics Warehouse"
- Section 3.5, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse"
- Section 3.6, "Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse"
- Section 3.7, "Additional Suggestions for Optimizing Oracle Performance in Oracle Business Analytics Warehouse"
- Section 3.8, "Partitioning Guidelines For Large Fact Tables"
- Section 3.9, "Miscellaneous Information About Oracle BI Applications Deployments"

3.1 General Guidelines for Setting Up Oracle Business Analytics Warehouse

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

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

  A complete listing of these tables is available in *Oracle Business Analytics Warehouse Data Model Reference.*
- To maximize ETL performance for Siebel CRM source systems running on DB2 databases, create three indexes on the Siebel OLTP database, using the following SQL commands:

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

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

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

The Informatica Repository stores all of the Informatica object definitions for the ETL mappings that populate the Oracle Business Analytics Warehouse. It is a series of repository tables that are stored in a database, which can be a transactional, analytical, or 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 indexes.
- 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.
- For Oracle databases, size the buffer pools based on content and size (number of tables and their sizes) of tablespaces.
- Allocate about 75 percent of the total available server memory to the database, assuming no other application is running on the same server.

During the Oracle Business Analytics Warehouse configuration process, when you create the data warehouse tables using the procedure Section 4.12.1, "Creating Data Warehouse Tables," you can create tables in one tablespace and indexes in another tablespace. However, for performance reasons, it is recommended that you create tablespaces as described in Table 3–1.

Table 3–1 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

Table 3–1 (Cont.) Recommended Tablespace Configuration

Tablespace Name	List of Tables
AGG	W_*A
OTHER	Remaining W* tables
DIM_INDX	Indexes of W_*D tables (for example, other tables would include W*G and W*GS tables)
FACT_INDX	Indexes of W_*F tables
OTHER_INDX	Remaining indexes of W* tables

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

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

Table 3–2 provides guidelines for parameter settings for DB2 relational database management system (RDBMS) usage. Use these guidelines as a starting point. You will need to make changes based on your specific database sizes, data shape, server size (CPU and memory), and type of storage. The database administrator should make changes to the settings based on performance monitoring and tuning considerations.

Table 3–2 Recommended DB2 Parameter Settings

Parameter	DB2 UDB V7	DB2 UDB V8 and V9	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	

Table 3–2 (Cont.) Recommended DB2 Parameter Settings

Parameter	DB2 UDB V7	DB2 UDB V8 and V9	Notes
NUM_ESTORE_SEGS	16	NIL	Restore is not needed in DB2 V8 64-bit because the V7 limit of 1.75 GB addressable memory has been lifted.
ESTORE_SEG_SZ	65536	NIL	
LOCKLIST	25000	25000	
APP_CTL_HEAP_SZ	5000	5000	
SORTHEAP	4000	4000	
STMTHEAP	40960	40960	
APPLHEAPSZ	2560	2560	
PCKCACHESZ	2560	2560	
STAT_HEAP_SZ	20000	20000	
DLCHKTIME	10000	10000	
MAXLOCKS	50	50	
LOCKTIMEOUT	1200	1200	
MAXAPPLS	500	500	
AVG_APPLS	10	10	
MAXFILOP	500	500	
GROUPHEAP_RATIO	N/A	70	New in V8
APPGROUP_MEM_SZ	N/A	30000	New in V8
DATABASE_MEMORY	N/A	AUTOMATIC	New in V8

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

3.3 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 Oracle BI Applications communicate with IBM DB2 UDB for z/OS and OS/390 (running on zSeries servers) through IBM DB2 Connect middleware.

  The following editions of DB2 Connect are supported:

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

  – **DB2 Connect Unlimited Edition (UE).** This edition provides the functionality of DB2 Connect Enterprise Edition but is priced differently.

- The ODBC driver for all connections must use the IBM DB2 ODBC Driver.
- Make the appropriate connections using the DB2 Client Configuration Assistant.
- Use the variable settings shown in Table 3–3.

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

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

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

This section provides guidelines for SQL Server database usage.

Note: The SQL Server database must be created with a collation sequence that supports binary sort order or case-sensitive dictionary sort order. Case-insensitive dictionary sort order is not supported. For example, for binary sort order with the U.S English character set, use the collation 'Latin1_General_BIN'. If you use the default collation setting of 'SQL_Latin1_General_CP1_CI_AS', the database is set to case-insensitive, which is not supported, and causes index creation failures.

This section includes the following topics:

- Section 3.4.1, "Setting the ANSI NULL Option"
- Section 3.4.2, "Modifying the DB Library Options Setting"
- Section 3.4.3, "Recommended SQL Server Database Parameters"

3.4.1 Setting the ANSI NULL Option

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

To set the ANSI NULL option

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

3.4.2 Modifying the DB Library Options Setting

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

To modify the DB Library Options setting

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

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

3.4.3 Recommended SQL Server Database Parameters

If necessary, SQL Server components can be configured to optimize performance, as shown in Table 3–4.

Table 3–4 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 multi threaded components (for example, several EIM threads).
Max server memory	2000 MB	Default is 2147483647.
Max text repl size	65536 B	
Max worker threads	100	Default is 255.
Media retention	0	
Min memory per query	1024 KB	
Min server memory	500 MB	Default is 0.

Table 3–4 (Cont.) Recommended Variable Settings for SQL Server Databases

Parameter	Recommended Setting	Notes
Nested triggers	1	
Network packet size	8192 B	Default is 4096.
Open objects	0	
Priority boost	0	
Query governor cost limit	0	Modify to 60 only if CPU is high.
Query wait	–1 sec	
Recovery interval	0 min	
Remote access	1	
Remote login timeout	20 sec	
Remote proc trans	0	
Remote query timeout	600 sec	
Scan for startup procs	0	
Set working set size	0	
Two-digit year cutoff	2049	
User connections	0	
User options	0	

- **SQL Server memory:** Make sure adequate memory is available.
- **Transaction logs and TempDB:** Reside on a separate disk from those used by database data.
- **Full load:** Full Recovery model for the database.
- **Incremental (Refresh) load:** Change from Full to Bulk-Logged Recovery model.

3.5 Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse

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

- Section 3.5.1, "Installation of Required JDBC Driver for Teradata Databases"
- Section 3.5.2, "General Guidelines for Teradata Deployments"
- Section 3.5.3, "Best Practices For Teradata Deployments"

3.5.1 Installation of Required JDBC Driver for Teradata Databases

The Data Warehouse Administration Console (DAC) requires JDBC drivers for database connectivity. You should only use JDBC drivers that are compatible with the supported databases. For information about supported databases, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*. Since JDBC drivers show variations with different database versions, only drivers that are shipped with the database or downloaded from the database vendor site and are known to be certified for the given database version should be used. Currently, third-party JDBC drivers for the databases are not supported.

If you have a Unicode environment on a Teradata database, you must install the Teradata JDBC Driver for Teradata 12.0 or Teradata 13.0. This driver is required for all supported versions of Teradata databases, including versions earlier than version 12. The Teradata JDBC Driver for Teradata 12.0 and Teradata 13.0 is available at http://www.teradata.com/DownloadCenter, under the title Teradata JDBC Driver.

3.5.2 General Guidelines for Teradata Deployments

The following requirements apply to Teradata database usage:

- Install the Teradata Parallel Data Pump (TPump) Teradata Load utility 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 prebuilt Informatica Repository file Oracle_BI_DW_Teradata.rep, which is located in OracleBI\dwrep\Informatica\Repository.
- In the ODBC configuration for Teradata, set the session Mode to ANSI and DateFormat to AAA, before you create the schema. Tables must be created as case specific. If you do the ODBC configuration after the schema is created, tables might not be created as case specific.
- To avoid overflows if the arithmetic calculation involves any decimal data type more than 18,3 (precision, scale), add the following static source system parameters:

  - $$Hint_Tera_Post_Cast = "as Decimal(18,3))"
  - $$Hint_Tera_Pre_Cast = "Cast("

  For more information about setting source system parameters in DAC, see Section 4.19.1, "Setting DAC Source System Parameters."

- Install the reswords.txt file in the server/bin directory for Informatica. When you are configuring the reswords.txt file, note the following points:

  If any table name or column name contains a database reserved word, such as MONTH or YEAR, the session fails with database errors when the Informatica Integration Service executes SQL against the database. You can create and maintain a reserved words file, reswords.txt, in the server/bin directory. When the Integration Service initializes a session, it searches for reswords.txt. If the file exists, the Integration Service places quotes around matching reserved words when it executes SQL against the database.

  Use the following rules and guidelines when working with reserved words:

  - The Integration Service searches the reserved words file when it generates SQL to connect to source, target, and lookup databases.
  - If you override the SQL for a source, target, or lookup, you must enclose any reserved word in quotes.
  - You may need to enable some databases, such as SQL Server, to use SQL-92 standards regarding quoted identifiers. Use connection environment SQL to issue the command.

For example, use the following command with SQL Server: SET QUOTED_ IDENTIFIER ON Sample reswords.txt File.

To use a reserved words file, create a file named reswords.txt and place it in the server/bin directory. Create a section for each database that you need to store reserved words for. Add reserved words used in any table or column name. You do not need to store all reserved words for a database in this file. Database names and reserved words in reswords.txt are not case sensitive.

A sample reswords.txt file is listed below:

```
[Teradata]
MONTH
DATE
INTERVAL
```

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

3.5.3 Best Practices For Teradata Deployments

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

- Section 3.5.3.1, "Collect Statistics as a Prerequisite"
- Section 3.5.3.2, "LEFT OUTER JOIN Issue"
- Section 3.5.3.3, "Group By Versus Distinct"
- Section 3.5.3.4, "Pruning of Mappings and Tables"
- Section 3.5.3.5, "Loader Configurations"

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

3.5.3.1 Collect Statistics as a Prerequisite

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).

DAC does the recollection of statistics as part of the ETL processes. However, DAC issues the collect statistics statement at the table level only (for example, collect statistics on w_org_d), and only for existing statistics.

3.5.3.2 LEFT OUTER JOIN Issue

Teradata code performance is highly dependent on the specific environment of each installation. A high number of occurrences of a single (or few) values in columns that are involved in joins, whether null or not, may cause skewing of data across Teradata AMPs. The impact of this is increased likelihood of exceeding the "per AMP" spool limit as well as increased CPU usage on one AMP while the other AMPs are much less utilized for the query experiencing the skew. This extends the processing time for this query, and negatively impacts other queries in the system that are competing for CPU resources on the AMP experiencing the skew.

Depending on the environment, the current code may redistribute the outer table on the join key, except when the inner table is very small; the Teradata Optimizer may choose to copy the inner table to all AMPs and not redistribute the outer table. Excessive nulls or other values in foreign keys will skew data in Teradata during join operations. If this happens, verify that statistics were defined and collected on the tables involved. If all the necessary statistics are defined and recently collected, it may be necessary to rewrite the SQL.

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

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

3.5.3.3 Group By Versus Distinct

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

3.5.3.4 Pruning of Mappings and Tables

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

3.5.3.5 Loader Configurations

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

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

- Teradata Parallel Data Pump (Tpump) - for more information, see Section 3.5.3.5.1, "Tpump".
- Fastload - for more information, see Section 3.5.3.5.2, "Fastload".
- Mload - for more information, refer to the Teradata documentation.

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

- Staged Mode: The Informatica process does the following in this order:
  - Reads from the source data.
  - Creates a data file.
  - Invokes the loader process to load the table using the data file created.

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

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

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

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

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

3.5.3.5.1 Tpump TPump is a data loading utility that helps you maintain (update, delete, insert, and atomic upsert) the data in your Teradata database. TPump allows you to achieve near real-time data in your data warehouse.

TPump uses standard Teradata SQL to achieve moderate to high data loading rates to the Teradata Database. Multiple sessions and multistatement requests are typically used to increase throughput.

Unlike most load utilities, TPump uses row hash locks rather than table level locks. This allows you to run queries while TPump is running. This also means that TPump can be stopped instantaneously.

Tpump can be used in the following modes:

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

- Tpump_Delete: Use to do deletes (this mode requires you to define the primary key in the Informatica target table definition).

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

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

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

3.5.3.5.2 Fastload The Fastload External Loader process is used on empty tables, such as loading staging tables and in initial loads where the tables are empty. When the Fastload process starts loading, it locks the target table, which means that processes (for example, lookups) cannot access that table. One solution to this problem is to specify dummy SQL for the look up overrides at the session level.

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

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

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

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

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

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

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

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

3.6 Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse

To configure the Business Analytics Data Warehouse on Oracle databases more easily, refer to the parameter template files init10gR2.ora and init11g.ora. These files are located in <DRIVE>:\<BI Applications install directory>\dwrep\Documentation\.

The parameter template files provide parameter guidelines based on the cost-based optimizer for Oracle 10g and 11g. Use these guidelines as a starting point. You will need to make changes based on your specific database sizes, data shape, server size (CPU and memory), and type of storage. The database administrator should make changes to the settings based on performance monitoring and tuning.

Copy the appropriate template file into your $ORACLE_HOME/dbs directory. Then, review the recommendations in the template file, and make the changes based on your specific database configuration. The database administrator should make changes to the settings based on performance monitoring and tuning considerations.

Note: The NLS_LENGTH_SEMANTICS parameter enables you to define byte- or character-length semantics. Oracle BI Applications supports BYTE and CHAR values for this parameter. You can add this parameter to the init10gR2.ora and init11g.ora files if you are using MLS characters.

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

This section contains additional suggestions for optimizing performance for Oracle databases.

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

  These settings are required for adequate performance from the dedicated Web client.
- Make sure that cost-based optimization is enabled in the Oracle development, test, and production databases and that statistics are kept up to date. Otherwise, the rule-based optimizer may be used.
- Create foreign keys in the Oracle database, but configure Oracle to not enforce the foreign key relationship. The existence of foreign keys will allow Oracle to better optimize certain queries. By turning off enforcement, the database load should not be negatively affected.
- Analyze application for occurrences of highly skewed data that is indexed. Create histogram statistics for these indexes 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)
   )
)
```

- 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.
- On the client side, edit the tnsnames.ora file. Modify the TNS alias by adding SDU= and TDU= as follows:

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

3.8 Partitioning Guidelines For Large Fact Tables

This section explains how to use partitioning to maximize performance in your Oracle BI Applications deployment. It contains the following topics:

- Section 3.8.1, "Introduction to Partitioning Large Fact Tables"
- Section 3.8.2, "Partitioning Large Fact Tables"
- Section 3.8.3, "Configuring DAC to Support Partitioned Tables"

3.8.1 Introduction to Partitioning Large Fact Tables

Taking advantage of range and composite range-range partitioning for fact tables reduces index and statistics maintenance time during ETL processes as well as improves Web query performance. Because the majority of inserts and updates impact the last partition(s), you only need to disable local indexes on a few impacted partitions, and then rebuild disabled indexes after the load and compute statistics on updated partitions only. Online reports and dashboards should also render results faster, since the optimizer builds more efficient execution plans using partitioning elimination logic.

Large fact tables, with more than 20 million rows, can be suitable for partitioning. To build an optimal partitioned table with reasonable data distribution, you can consider partitioning by month, quarter, year, and so on. You can either identify and partition target fact tables before the initial load or convert the populated tables into partitioned objects after the full load.

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

To deploy partitioned fact tables

1. Partition the large fact tables. For more information, see Section 3.8.2, "Partitioning Large Fact Tables."
2. Configure DAC to support ETL for partitioned tables. For more information, see Section 3.8.3, "Configuring DAC to Support Partitioned Tables."

3.8.2 Partitioning Large Fact Tables

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

The procedures in this section use an example that converts the fact table W_WRKFC_EVT_MONTH_F into a partitioned table and uses range partitioning by year.

To partition large fact tables

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

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

   - Identify eligible columns of type DATE for implementing range partitioning.
   - Connect to the Oracle BI Server repository and check the usage or dependencies on each column in the logical and presentation layers.
   - Analyze the summarized data distribution in the target table by each potential partitioning key candidate and data volumes per time range, month, quarter or year.
   - Based on the compiled data, decide on the appropriate partitioning key and partitioning range for your future partitioned table.
   - The recommended partitioning range for most implementations is a month; however, you may want to consider implementing quarterly or yearly partitioning ranges.
   - These guidelines assume the majority of incremental ETL volume data is made up of new records, which would be stored in one of the two latest partitions. Depending on the range granularity you chose, it is recommended that you rebuild local indexes for the most impacted of the latest partitions, as described below:

     – Monthly range. It is recommended that you maintain the two latest partitions, that is, the current and previous partition.

     – Quarterly range. It is only necessary to maintain the current partition.

     – Yearly range. It is recommended that you maintain the current partition.

2. Create a partitioned table.

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

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

   - Create the table using the "as select" SQL statement.

     This method is simpler and faster.

   - Create the table using exchange partition syntax and then split the partitions.

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

     The syntax in the following SQL commands uses the example of one tablespace named USERS:

     a. Rename the original table.

     ```
     SQL> rename W_WRKFC_EVT_MONTH_F to W_WRKFC_EVT_MONTH_F_ORIG;
     ```

     b. Create the partitioned table using the range partitioning by year:

     ```
     SQL> create table W_WRKFC_EVT_MONTH_F partition by range (EVENT_YEAR)(
     partition PART_MIN values less than (2006),
     ```

```
partition PART_2006 values less than (2007),
partition PART_2007 values less than (2008),
partition PART_2008 values less than (2009),
partition PART_2009 values less than (2010),
partition PART_2010 values less than (2011),
partition PART_MAX  values less than (maxvalue)
)
tablespace BIAPPS_DATA
nologging parallel enable row movement
as select * from W_WRKFC_EVT_MONTH_F_ORIG;
```

Note: You must use the format YYYY for partitioning by year, YYYYQQ for partitioning by quarter, and YYYYMMDD for partitioning by month. Make sure you check the partitioning column data type before partitioning the table.

The EVENT_YEAR column in the example above uses number (4) precision, so the table partition values are defined using the format YYYY. If you choose the WID column for a partitioning key, then you have to define your partition ranges using the format YYYYMMDD.

You can implement composite range-range partitioning using syntax similar to the following. This example uses the Quarter range for partitioning and the Year range for subpartitioning. Because the EXPENDITURE_DT_WID column has number(8) precision, the table partition values are defined using the format YYYYMMDD. You must use the correct date format for the range: YYYY for year, YYYYQQ for quarter, YYYYMMDD for month. Make sure you check the partitioning column data type before you partition a table.

```
SQL> create table W_PROJ_EXP_LINE_F
partition by range    (CHANGED_ON_DT)
subpartition by range (EXPENDITURE_DT_WID)
(partition PART_MIN values less then (TO_
DATE('01-JAN-2008','DD-MON-YYYY'))
( subpartition PART_MIN_MIN  values less than (19980000)
, subpartition PART_MIN_1998 values less than (19990000)
, subpartition PART_MIN_1999 values less than (20010000)
, subpartition PART_MIN_2001 values less than (20020000)
, subpartition PART_MIN_2002 values less than (20030000)
, subpartition PART_MIN_2003 values less than (20040000)
, subpartition PART_MIN_2004 values less than (20050000)
, subpartition PART_MIN_2005 values less than (20060000)
, subpartition PART_MIN_2006 values less than (20070000)
, subpartition PART_MIN_2007 values less than (20080000)
, subpartition PART_MIN_2008 values less than (20090000)
, subpartition PART_MIN_2009 values less than (20100000)
, subpartition PART_MIN_MAX values less than (MAXVALUE)
)
(partition PART_200801 values less then (TO_
DATE('01-APR-2008','DD-MON-YYYY'))
( subpartition PART_200801_MIN  values less than (19980000)
( subpartition PART_200801_1998  values less than (19990000)
( subpartition PART_200801_1999  values less than (20010000)
( subpartition PART_200801_2001  values less than (20020000)
( subpartition PART_200801_2002  values less than (20030000)
( subpartition PART_200801_2003  values less than (20040000)
( subpartition PART_200801_2004  values less than (20050000)
( subpartition PART_200801_2005  values less than (20060000)
( subpartition PART_200801_2006  values less than (20070000)
( subpartition PART_200801_2007  values less than (20080000)
( subpartition PART_200801_2008  values less than (20090000)
( subpartition PART_200801_2009  values less than (20100000)
( subpartition PART_200801_MAX  values less than (MAXVALUE)
)
...
...
, partition PART_MAX values less than (maxvalue)
( subpartition PART_MAX_MIN values less than (19980000)
, subpartition PART_MAX_1998 values less than (19990000)
, subpartition PART_MAX_1999 values less than (20010000)
, subpartition PART_MAX_2001 values less than (20020000)
, subpartition PART_MAX_2002 values less than (20030000)
, subpartition PART_MAX_2003 values less than (20040000)
, subpartition PART_MAX_2004 values less than (20050000)
, subpartition PART_MAX_2005 values less than (20060000)
, subpartition PART_MAX_2006 values less than (20070000)
, subpartition PART_MAX_2007 values less than (20080000)
, subpartition PART_MAX_2008 values less than (20090000)
, subpartition PART_MAX_2009 values less than (20100000)
, subpartition PART_MAX_MAX values less than (maxvalue)
)
) nologging parallel
enable row movement
as (select * from W_PROJ_EXP_LINE_F_ORIG);
```

3. Drop and rename indexes on the renamed table.

   ```
   SQL> spool drop_ind.sql
   SQL> SELECT ’DROP INDEX ’|| INDEX_NAME||’;’ FROM USER_INDEXES
        WHERE TABLE_NAME = ’W_WRKFC_EVT_MONTH_F_ORIG’;
   SQL> spool off
   SQL> @drop_ind.sql
   ```

   If you want to keep indexes on the original renamed table until you have successfully completed the partitioning conversion, use the following commands:

   ```
   SQL> spool rename_ind.xql
   SQL> SELECT 'ALTER INDEX '|| INDEX _NAME ||' rename to '|| INDEX_NAME || '_
   ORIG; ' FROM USER_INDEXES
   WHERE TABLE_NAME = 'W_WRKFC_EVT_MONTH_F_ORIG';
   SQL> spool off
   SQL> @rename_ind.sql
   ```

4. Create global and local indexes.

   a. Execute the following queries as the DAC Repository owner:

      ```
      SQL> spool indexes.sql
      SQL> SELECT ’CREATE’
           ||DECODE(ISUNIQUE,’Y’,’UNIQUE’)
           ||DECODE(ISBITMAP,’Y’,’BITMAP’)
           ||’INDEX’
           ||I.NAME
           ||’ON’
           ||T.NAME
           ||’(’
           ||MAX(DECODE(POSTN,1,C.NAME||’ASC’))
           ||MAX(DECODE(POSTN,2,’ ,’||C.NAME||’ASC’))
           ||MAX(DECODE(POSTN,3,’ ,’||C.NAME||’ASC’))
      ```

```
     ||MAX(DECODE(POSTN,4,' ,'||C.NAME||'ASC'))
     ||MAX(DECODE(POSTN,5,' ,'||C.NAME||'ASC'))
     ||MAX(DECODE(POSTN,6,' ,'||C.NAME||'ASC'))
     ||MAX(DECODE(POSTN,7,' ,'||C.NAME||'ASC'))
     ||') tablespace USERS_IDX '
     ||DECODE(ISUNIQUE, 'Y','GLOBAL','LOCAL')
     ||' NOLOGGING;'
FROM  W_ETL_TABLE T, W_ETL_INDEX I, W_ETL_INDEX_COL C
WHERE T.ROW_WID = I.TABLE_WID
      AND T.NAME = 'W_WRKFC_EVT_MONTH_F'
      AND I.ROW_WID = C.INDEX_WID
      AND I.INACTIVE_FLG = 'N'
GROUP BY T.NAME,I.NAME,ISBITMAP,ISUNIQUE;
SQL> spool off;
```

This script creates indexes with a maximum of seven positions. If you have indexes with more than seven column positions, then modify the "MAX(DECODE(POSTN...))" statement.

b. Run the spooled file indexes.sql in the data warehouse schema. For example:

```
SQL> @indexes.sql
```

5. Compute statistics on the partitioned table. For example:

```
SQL> BEGIN
     dbms_stats.Gather_table_stats(
          NULL
          tabname => 'W_WRKFC_EVT_MONTH_F',
          CASCADE => true,
          estimate_percent => dbms_stats.auto_sample_size,
          method_opt => 'FOR ALL INDEXED COLUMNS SIZE AUTO');
     END;
```

6. Configure Informatica to support partitioned tables by enabling row movement.

   a. In Informatica Workflow Manager, from the menu bar select Connections, and then Relational.

   b. In the Relational Connection Browser dialog box, select the DataWarehouse connection.

   c. Update the Connection Environment SQL with the following:

```
ALTER SESSION SET SKIP_UNUSABLE_INDEX=TRUE;
```

You now need to configure DAC to support the partitioned tables. For instructions, see Section 3.8.3, "Configuring DAC to Support Partitioned Tables."

3.8.3 Configuring DAC to Support Partitioned Tables

Once you have partitioned your fact tables as described in Section 3.8.2, "Partitioning Large Fact Tables," you need to configure DAC to support the partitioned tables. In this process, you first create new source system parameters. Then, you use the DAC Actions feature to create index actions for overriding the default behavior for dropping and creating indexes. You also create a table action for overriding the default action for analyzing tables. After you create the index and table actions, you then need to associate these actions with the appropriate indexes and tables in the DAC Repository.

For more information about the DAC Action feature, see *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

> **Note:** The example in this process shows how to set up rebuilding indexes and maintaining statistics for the last two partitions, that is, the previous and current partitions, for range partitioning by year. You should consider implementing previous and current partitions only for monthly or more granular ranges. If you implement partitions for quarterly or yearly ranges, you only need to maintain the current partition. Maintaining the previous partition for partitioning by a quarter or a year may introduce unnecessary overhead and extend your incremental ETL execution time.

To configure DAC to support partitioned tables, you must complete the following procedures:

- Section 3.8.3.1, "Creating Source System Parameters in DAC to Support Partitioned Tables"
- Section 3.8.3.2, "Creating Index Actions in DAC"
- Section 3.8.3.3, "Creating Table Actions in DAC"
- Section 3.8.3.4, "Assigning Index Actions to Indexes in DAC"
- Section 3.8.3.5, "Assigning Table Actions to Tables in DAC"

For additional information about the DAC Actions feature, see the *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*, which is available on the Oracle Technology Network at `http://www.oracle.com/technology/documentation/bi_dac.html`.

3.8.3.1 Creating Source System Parameters in DAC to Support Partitioned Tables

Follow this procedure to create source system parameters in DAC to support partitioned tables.

To create source system parameters in DAC to support partitioned tables

1. Log into DAC.

   For instructions on logging into DAC, see Section A.1, "How to Log into DAC."
2. Go to the Design view, and select the appropriate custom container from the drop-down list.
3. Click the Source System Parameters tab.
4. In the toolbar, click New to open a new record.
5. For yearly partitions:

   a. Create a parameter with the following values:

      -Name: $$CURRENT_YEAR_WID

      -Data Type: SQL

      -Value: `SELECT TO_CHAR(ROW_WID) FROM W_YEAR_D WHERE W_CURRENT_CAL_YEAR_CODE = 'Current'`

      -Logical Data Source: DBConnection_OLAP

   b. Create a second parameter with the following values:

      -Name: $$PREVIOUS_YEAR_WID

      -Data Type: SQL

-Value: `SELECT TO_CHAR(ROW_WID) FROM W_YEAR_D WHERE W_CURRENT_CAL_YEAR_CODE = 'Previous'`

-Logical Data Source: DBConnection_OLAP

6. For monthly partitions:

   a. Create a parameter with the following values:

      -Name: $$CURRENT_MONTH_WID

      -Data Type: SQL

      -Value: `SELECT TO_CHAR(ROW_WID) FROM W_MONTH_D WHERE W_CURRENT_CAL_MONTH_CODE = 'Current'`

      -Logical Data Source: DBConnection_OLAP

   b. Create a second parameter with the following values:

      -Name: $$PREVIOUS_MONTH_WID

      -Data Type: SQL

      -Value: `SELECT TO_CHAR(ROW_WID) FROM W_MONTH_D WHERE W_CURRENT_CAL_MONTH_CODE = 'Previous'`

      -Logical Data Source: DBConnection_OLAP

7. For quarterly partitions:

   a. Create a parameter with the following values:

      -Name: $$CURRENT_QTR_WID

      -Data Type: SQL

      -Value: `SELECT TO_CHAR(ROW_WID) FROM W_QTR_D WHERE W_CURRENT_CAL_QTR_CODE = 'Current'`

      -Logical Data Source: DBConnection_OLAP

   b. Create a second parameter with the following values:

      -Name: $$PREVIOUS_QTR_WID

      -Data Type: SQL

      -Value: `SELECT TO_CHAR(ROW_WID) FROM W_QTR_D WHERE W_CURRENT_CAL_QTR_CODE = 'Previous'`

      -Logical Data Source: DBConnection_OLAP

3.8.3.2 Creating Index Actions in DAC

This section provides instructions for creating index actions to disable and create indexes on partitioned tables. It includes the following topics:

- Section 3.8.3.2.1, "How to Create the Index Action to Disable the Local Index Parameter"
- Section 3.8.3.2.2, "How to Create the Index Action to Enable the Local Index Parameter"
- Section 3.8.3.2.3, "How to Create the Index Action to Enable the Local Sub-Partitioned Index Parameter"
- Section 3.8.3.2.4, "How to Create the Index Action to Create the Local Bitmap Index Parameter"

- Section 3.8.3.2.5, "How to Create the Index Action to Create the Local B-Tree Index Parameter"
- Section 3.8.3.2.6, "How to Create the Index Action to Create the Global Unique Index Parameter"

3.8.3.2.1 How to Create the Index Action to Disable the Local Index Parameter This index action disables local indexes. This example uses the year partition range. If you use quarterly or monthly partition ranges, substitute the correct name for the action and in the SQL, for example, PREVIOUS_MONTH_WID/CURRENT_MONTH_WID or PREVIOUS_QTR_WID/CURRENT_QTR_WID. You must define separate actions for each range you use.

1. Log into DAC.

   For instructions on logging into DAC, see Section A.1, "How to Log into DAC."

2. Go to the Design view, and select the appropriate custom container from the drop-down list.
3. From the Menu bar, select Tools, then Seed Data, then Actions, then Index Actions.
4. In the Index Actions dialog box, click New.

   A new record field appears at the top of the list of actions.

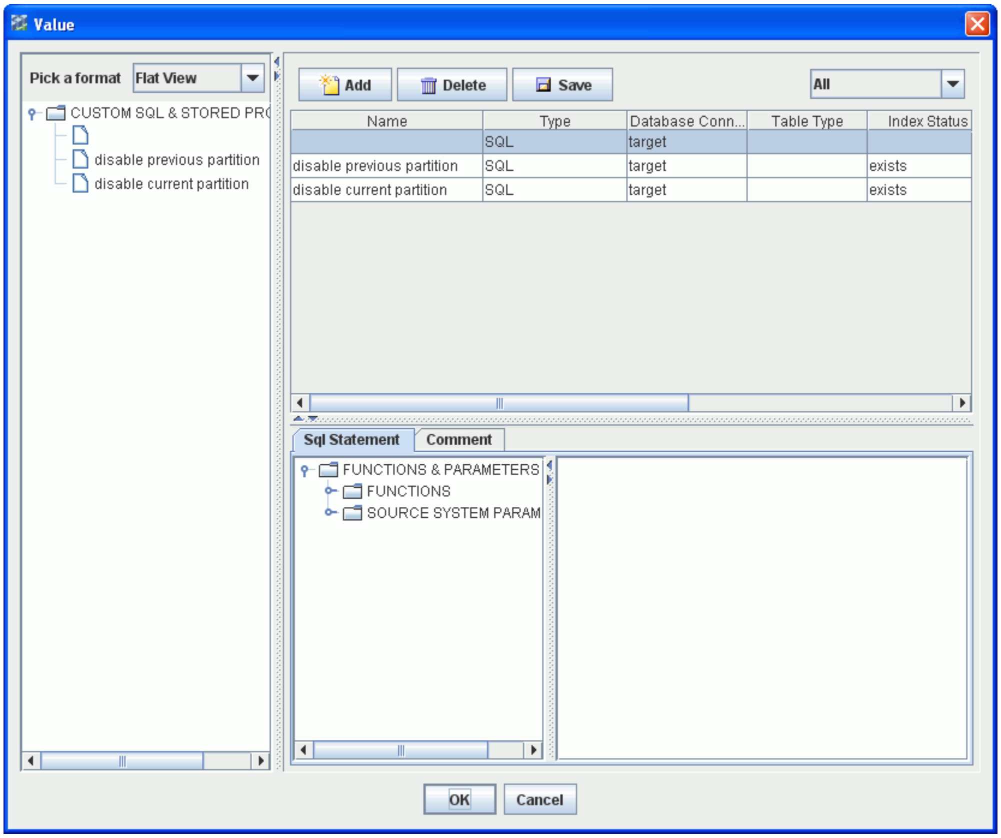
5. In the name field, enter "Year Partitioning: Disable Local Index."
6. Click Save.
7. Double-click in the Value field to open the Value dialog box.

   The Value dialog box appears.

8. Define the SQL script:

   a. Click Add.

      A new record field appears at the top of the list of SQL blocks.



Name	Type	Database Conn...	Table Type	Index Status
	SQL	target		
disable previous partition	SQL	target		exists
disable current partition	SQL	target		exists

b. Enter the following information in the new record field:

Field	**Description**
Name	Enter: Disable PREVIOUS_YEAR_WID Local Indexes
Type	Select: SQL
Database Connection	Select: Target
Valid Database Platforms	Double-click in the field to open the Supported Database Types dialog box, and select the appropriate database type.

c. In the text box on the lower-right side of the window, enter the following SQL:

```
alter index getIndexName() modify partition PART_@DAC_
$$PREVIOUS_YEAR_WID unusable
```

Note: Do not use a semicolon (;) at the end of the SQL in the text area.

d. Click Save.

9. Define the SQL:

a. Click Add.

A new record field appears at the top of the list of SQL blocks.

b. Enter the following information in the new record field:

Field	Description
Name	Enter: Disable CURRENT_YEAR_WID Local Indexes
Type	Select: SQL
Database Connection	Select: Target
Valid Database Platforms	Double-click in the field to open the Supported Database Types dialog box, and select the appropriate database type.

c. In the text box on the lower-right side of the window, enter the following SQL:

```
alter index getIndexName() modify partition PART_@DAC_
$$CURRECT_YEAR_WID unusable
```

d. Click Save.

3.8.3.2.2 How to Create the Index Action to Enable the Local Index Parameter This index action enables local indexes. This example uses the year partition range. If you use quarterly or monthly partition ranges, substitute the correct name for the action and in the SQL, for example, PREVIOUS_MONTH_WID/CURRENT_MONTH_WID or PREVIOUS_QTR_WID/CURRENT_QTR_WID. You must define separate actions for each range you use.

1. Log into DAC.

   For instructions on logging into DAC, see Section A.1, "How to Log into DAC."

2. Go to the Design view, and select the appropriate custom container from the drop-down list.

3. From the Menu bar, select Tools, then Seed Data, then Actions, then Index Actions.

4. In the Index Actions dialog box, click New.

   A new record field appears at the top of the list of actions.

5. In the name field, enter "Year Partitioning: Enable Local Index."

6. Click Save.

7. Double-click in the Value field to open the Value dialog box.

   The Value dialog box appears.

8. Define the SQL script:

   a. Click Add.

   A new record field appears at the top of the list of SQL blocks.

   b. Enter the following information in the new record field:

Field	Description
Name	Enter: Enable PREVIOUS_YEAR_WID Local Indexes
Type	Select: SQL
Database Connection	Select: Target

Field	Description
Valid Database Platforms	Double-click in the field to open the Supported Database Types dialog box, and select the appropriate database type.

c. In the text box on the lower-right side of the window, enter the following SQL:

```
alter index getIndexName() rebuild partition PART_@DAC_
$$PREVIOUS_YEAR_WID nologging
```

Note: Do not use a semicolon (;) at the end of the SQL in the text area.

d. Click Save.

9. Define the SQL script:

a. Click Add.

A new record field appears at the top of the list of SQL blocks.

b. Enter the following information in the new record field:

Field	Description
Name	Enter: Enable CURRENT_YEAR_WID Local Indexes
Type	Select: SQL
Database Connection	Select: Target
Valid Database Platforms	Double-click in the field to open the Supported Database Types dialog box, and select the appropriate database type.

c. In the text box on the lower-right side of the window, enter the following SQL:

```
alter index getIndexName() rebuild partition PART_@DAC_
$$CURRECT_YEAR_WID nologging
```

Note: Do not use a semicolon (;) at the end of the SQL in the text area.

d. Click Save.

3.8.3.2.3 How to Create the Index Action to Enable the Local Sub-Partitioned Index Parameter

This index action is for composite partitioning only. This example uses the year partition range. If you use quarterly or monthly partition ranges, substitute the correct name for the action and in the SQL.

1. Log into DAC.

   For instructions on logging into DAC, see Section A.1, "How to Log into DAC."

2. Go to the Design view, and select the appropriate custom container from the drop-down list.
3. From the Menu bar, select Tools, then Seed Data, then Actions, then Index Actions.
4. In the Index Actions dialog box, click New.

   A new record field appears at the top of the list of actions.

5. In the name field, enter "Year Partitioning: Enable Local Index."
6. Click Save.

7. Double-click in the Value field to open the Value dialog box.

   The Value dialog box appears.

8. Define the SQL script:

   a. Click Add.

      A new record field appears at the top of the list of SQL blocks.

   b. Enter the following information in the new record field:

Field	Description
Name	Enter: Enable Local Sub-partitioned Index
Type	Select: Stored Procedure
Database Connection	Select: Target
Valid Database Platforms	Double-click in the field to open the Supported Database Types dialog box, and select the appropriate database type.

   c. In the text box on the lower-right side of the window, enter the following SQL:

```
DECLARE
     CURSOR C1 IS
     SELECT DISTINCT SUBPARTITION_NAME
     FROM USER_IND_SUBPARTITIONS
     WHERE INDEX_NAME=’getIndexName()’ AND STATUS = ’UNUSABLE’;
BEGIN
   FOR REC IN C1 LOOP
     EXECUTE IMMEDIATE ’alter index getIndexName() rebuild subpartition
’||REC.SUBPARTITION_NAME||’’;
   END LOOP;
END
```

      Note: Do not use a semicolon (;) at the end of the SQL in the text area.

   d. Click Save.

3.8.3.2.4 How to Create the Index Action to Create the Local Bitmap Index Parameter

This example uses the year partition range. If you use quarterly or monthly partition ranges, substitute the correct name for the action and in the SQL.

1. Log into DAC.

   For instructions on logging into DAC, see Section A.1, "How to Log into DAC."

2. Go to the Design view, and select the appropriate custom container from the drop-down list.

3. From the Menu bar, select Tools, then Seed Data, then Actions, then Index Actions.

4. In the Index Actions dialog box, click New.

   A new record field appears at the top of the list of actions.

5. In the name field, enter "Year Partitioning: Create Local Bitmap Index."

6. Click Save.

7. Double-click in the Value field to open the Value dialog box.

   The Value dialog box appears.

8. Define the SQL script:

   a. Click Add.

      A new record field appears at the top of the list of SQL blocks.

   b. Enter the following information in the new record field:

Field	Description
Name	Enter: Create Local Bitmap Indexes
Type	Select: SQL
Database Connection	Select: Target
Valid Database Platforms	Double-click in the field to open the Supported Database Types dialog box, and select the appropriate database type.

   c. In the text box on the lower-right side of the window, enter the following SQL:

```
Create bitmap index getIndexName() on getTableName()
(getUniqueColumns()) tablespace getTableSpace() local
parallel nologging
```

      Note: Do not use a semicolon (;) at the end of the SQL in the text area.

   d. Click Save.

3.8.3.2.5 How to Create the Index Action to Create the Local B-Tree Index Parameter

This example uses the year partition range. If you use quarterly or monthly partition ranges, substitute the correct name for the action and in the SQL.

1. Log into DAC.

   For instructions on logging into DAC, see Section A.1, "How to Log into DAC."

2. Go to the Design view, and select the appropriate custom container from the drop-down list.

3. From the Menu bar, select Tools, then Seed Data, then Actions, then Index Actions.

4. In the Index Actions dialog box, click New.

   A new record field appears at the top of the list of actions.

5. In the name field, enter "Year Partitioning: Create Local B-Tree Index."

6. Click Save.

7. Double-click in the Value field to open the Value dialog box.

   The Value dialog box appears.

8. Define the SQL script:

   a. Click Add.

      A new record field appears at the top of the list of SQL blocks.

   b. Enter the following information in the new record field:

Field	Description
Name	Enter: Create Local B-Tree Index
Type	Select: SQL

Field	Description
Database Connection	Select: Target
Valid Database Platforms	Double-click in the field to open the Supported Database Types dialog box, and select the appropriate database type.

c. In the text box on the lower-right side of the window, enter the following SQL:

```
Create index getIndexName() on getTableName()
(getUniqueColumns()) tablespace getTableSpace() local
parallel nologging
```

Note: Do not use a semicolon (;) at the end of the SQL in the text area.

d. Click Save.

3.8.3.2.6 How to Create the Index Action to Create the Global Unique Index Parameter

This example uses the year partition range. If you use quarterly or monthly partition ranges, substitute the correct name for the action and in the SQL.

1. Log into DAC.

   For instructions on logging into DAC, see Section A.1, "How to Log into DAC."
2. Go to the Design view, and select the appropriate custom container from the drop-down list.
3. From the Menu bar, select Tools, then Seed Data, then Actions, then Index Actions.
4. In the Index Actions dialog box, click New.

   A new record field appears at the top of the list of actions.
5. In the name field, enter "Year Partitioning: Create Global Unique Index."
6. Click Save.
7. Double-click in the Value field to open the Value dialog box.

   The Value dialog box appears.
8. Define the SQL script:

   a. Click Add.

   A new record field appears at the top of the list of SQL blocks.

   b. Enter the following information in the new record field:

Field	Description
Name	Enter: Create Global Unique Index
Type	Select: SQL
Database Connection	Select: Target
Valid Database Platforms	Double-click in the field to open the Supported Database Types dialog box, and select the appropriate database type.

c. In the text box on the lower-right side of the window, enter the following SQL:

```
Create index getIndexName() on getTableName()
(getUniqueColumns()) tablespace getTableSpace() global
parallel nologging
```

Note: Do not use a semicolon (;) at the end of the SQL in the text area.

d. Click Save.

3.8.3.3 Creating Table Actions in DAC

This section provides instructions for creating table actions to gather statistics on partitioned tables. It includes the following topic:

- Section 3.8.3.3.1, "How to Create a Table Action to Gather Statistics on a Partitioned Table"

3.8.3.3.1 How to Create a Table Action to Gather Statistics on a Partitioned Table This example uses the year partition range. If you use quarterly or monthly partition ranges, substitute the correct name for the action and in the SQL.

1. Log into DAC.

   For instructions on logging into DAC, see Section A.1, "How to Log into DAC."

2. Go to the Design view, and select the appropriate custom container from the drop-down list.

3. From the Menu bar, select Tools, then Seed Data, then Actions, then Table Actions.

4. In the Table Actions dialog box, click New.

   A new record field appears at the top of the list of actions.

5. In the name field, enter "Year Partitioning: Gather Partition Stats."

6. Click Save.

7. Double-click in the Value field to open the Value dialog box.

   The Value dialog box appears.

8. Define the SQL script:

   a. Click Add.

      A new record field appears at the top of the list of SQL blocks.

   b. Enter the following information in the new record field:

Field	Description
Name	Enter: Gather Partition Stats
Type	Select: Stored Procedure
Database Connection	Select: Target
Valid Database Platforms	Double-click in the field to open the Supported Database Types dialog box, and select the appropriate database type.

   c. In the text box on the lower-right side of the window, enter the following SQL:

```
DECLARE
     CURSOSR C1 IS
     SELECT DISTINCT UIP.PARTITION_NAME
     FROM USER_IND_PARTITIONS UIP, USER_PART_INDEXES UPI
     WHERE UPI.TABLE_NAME = 'getTableName()'
          AND UIP.INDEX_NAME=UPI.INDEX_NAME
          AND UIP.STATUS = 'USABLE'
          AND UIP.PARTITION_NAME IN
('PART_@DAC_$$CURRENT_YEAR_WID','PART_@DAC_$$PREVIOUS_YEAR_WID');
BEGIN
   FOR REC IN C1 LOOP
DBMS_STATS.GATHER_TABLE_STATS(
               NULL,
               TABNAME => 'getTableName()',
               CASCADE => TRUE
               PARTNAME => REC.PARTITION_NAME
            ESTIMATE_PERCENT => DBMS_STATS.AUTO_SAMPLE_SIZE,
            GRANULARITY => 'PARTITION',
            METHOD_OPT => 'FOR ALL INDEXED COLUMNS SIZE AUTO',
            DEGREE => DBMS_STATS.DEFAULT_DEGREE);
   END LOOP;
END
```

Note: Do not use a semicolon (;) at the end of the SQL in the text area.

d. Click Save.

3.8.3.3.2 How to Create a Table Action to Gather Statistics on a Partitioned Table for Composite Partitioning If you use quarterly or monthly partition ranges, substitute the correct name for the action and in the SQL.

Note: Do not change the Drop and Create Always or Drop and Create Always Bitmap properties for the modified indexes. If you deselect these check boxes, DAC will skip the index actions that are defined.

1. Log into DAC.

   For instructions on logging into DAC, see Section A.1, "How to Log into DAC."

2. Go to the Design view, and select the appropriate custom container from the drop-down list.

3. From the Menu bar, select Tools, then Seed Data, then Actions, then Table Actions.

4. In the Table Actions dialog box, click New.

   A new record field appears at the top of the list of actions.

5. In the name field, enter "Quarter Composite Partitioning: Gather Partition Stats."

6. Click Save.

7. Double-click in the Value field to open the Value dialog box.

   The Value dialog box appears.

8. Define the SQL script:

   a. Click Add.

      A new record field appears at the top of the list of SQL blocks.

   b. Enter the following information in the new record field:

Field	Description
Name	Enter: Gather Partition Stats
Type	Select: Stored Procedure

Field	Description
Database Connection	Select: Target
Valid Database Platforms	Double-click in the field to open the Supported Database Types dialog box, and select the appropriate database type.

c. In the text box on the lower-right side of the window, enter the following SQL:

```
DECLARE
     CURSOSR C1 IS
     SELECT DISTINCT UIP.PARTITION_NAME
     FROM USER_IND_PARTITIONS UIP, USER_PART_INDEXES UPI
     WHERE UPI.TABLE_NAME = 'getTableName()'
          AND UIP.INDEX_NAME=UPI.INDEX_NAME
          AND UIP.STATUS = 'USABLE'
          AND UIP.PARTITION_NAME IN
('PART_@DAC_$$CURRENT_QTR_WID','PART_@DAC_$$PREVIOUS_QTR_WID');
BEGIN
   FOR REC IN C1 LOOP
DBMS_STATS.GATHER_TABLE_STATS(
               NULL,
               TABNAME => 'getTableName()',
               CASCADE => TRUE
               PARTNAME => REC.PARTITION_NAME
            ESTIMATE_PERCENT => DBMS_STATS.AUTO_SAMPLE_SIZE,
            GRANULARITY => 'PARTITION',
            METHOD_OPT => 'FOR ALL INDEXED COLUMNS SIZE AUTO',
            DEGREE => DBMS_STATS.DEFAULT_DEGREE);
   END LOOP;
END
```

Note: Do not use a semicolon (;) at the end of the SQL in the text area.

d. Click Save.

3.8.3.4 Assigning Index Actions to Indexes in DAC

After you create index actions in DAC, you need to assign the index actions to specific indexes for each of the following index actions you created in the previous procedure:

- Disable Local Indexes
- Enable Local Indexes
- Create Local Bitmap Index
- Create Local B-Tree Index
- Create Global Unique Index

To assign index actions to indexes

1. In the DAC Design view, click the Index tab.
2. Query for the appropriate indexes on the partitioned tables, based on the index action you want to assign.

   Note: Do not include global indexes in the query. Global indexes must not have any assigned index action tasks.
3. Right-click on the list of query results, and select Add Actions.

   The Add Actions dialog box opens.

4. In the Action Type field, select the appropriate Action Type.
5. In the Load Type field:

   Select Incremental for Disable and Enable Local Indexes actions

   Select Initial for Create Local Bitmap Index, Create Local B-Tree Index, and Create Global Unique Index.
6. In the Action field, double-click to open the Choose Action dialog.
7. Select the appropriate action name.
8. Click OK to close the Add Actions dialog box.

3.8.3.5 Assigning Table Actions to Tables in DAC

After you create table actions in DAC, you need to assign the table actions to specific tables for each of the table actions you created in the previous procedure.

To assign index actions to indexes

1. In the DAC Design view, click the Index tab.
2. Query for the appropriate table, based on the table action you want to assign.
3. Right-click on the list of query results, and select Add Actions.

   The Add Actions dialog box opens.
4. In the Action Type field, select Analyze Table.
5. In the Load Type field, select Incremental.
6. In the Action field, double-click to open the Choose Action dialog.
7. Select the appropriate action name, for example, <Range> Partitioning: Gather Partition Stats.
8. Click OK to close the Add Actions dialog box.
9. Click OK to close the Add Actions dialog box.

Note: Make sure you use the appropriate composite partitioning table action for composite range-range tables.

3.9 Miscellaneous Information About Oracle BI Applications Deployments

This section contains the following topics:

- Section 3.9.1, "Preconfigured Mapping for the SA System Subject Area"
- Section 3.9.2, "Using Initialization Blocks"
- Section 3.9.3, "Creating Custom Indexes in Siebel Source Databases for Incremental Load Performance"
- Section 3.9.4, "Creating Custom Indexes in Oracle EBS Source Databases for Incremental Load Performance"
- Section 3.9.5, "Running an ETL with a Non-English OLTP Data Source"

3.9.1 Preconfigured Mapping for the SA System Subject Area

For Oracle's Siebel Applications customers, Table 3–5 describes the preconfigured mappings for the SA system subject area. Fields that are not available in Oracle's Siebel transactional database will default to values shown in the table.

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

  ```
  SA User.(User)
  ```

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

- **Setting Provider Information.** Typically, the cell phone and the fax numbers in the Oracle Business Analytics Warehouse do not contain a provider name. Therefore, the Pager will typically be a numeric value such as 555-483-3843. To append a provider to this address, use the following guidelines:

  - If the entire company has the same provider, then you can append the provider in the column mappings.
  - If users can have different providers, you need to create an extension table. For instructions, refer to the documentation about configuring tables and columns for Oracle's Siebel business applications.

Table 3–5 Preconfigured Mappings for the User Table in the SA System Subject Area

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

3.9.2 Using Initialization Blocks

Table 3–6 lists some of the initialization blocks common to all Oracle BI Applications and their purposes. Initialization blocks that are specific to each Oracle BI Applications area are not listed here.

To view the initialization blocks provided with Oracle BI Applications, open the Variable Manager in the Oracle Business Intelligence Enterprise Edition Administration Tool. For instructions, see *Oracle Business Intelligence Server Administration Guide*.

Table 3–6 Initialization Blocks and Their Purposes

Initialization Block	Purpose
Authorization	Calculates user's responsibilities from the database.
Authentication	Authenticates against the database and verifies the user exists as a operational application user.
External Metadata Strings	Gets the translated value of metadata strings for the user's locale. This initialization block is critical to Intelligence Dashboards in international deployment.
LOCALE	Sets the user's locale specification in the Oracle BI Server.
Login Properties	Gets user's login properties, such as full name, from the database. This initialization block also sets the logging level of the users. By default, the log level is 0 for all users. If you wish to generate Oracle BI query logs for all users, this initialization block should be updated by changing the default value as well as the value in the initialization SQL.
Default System Language ID	Sets the variable OLTP_LANG_ID by querying the operational application database.
Organizations for Org-based Security	Queries the operational application database to get the organization membership for each user. It sets the variable ORGANIZATION.
Primary Owner ID	Sets the primary owner ID based on the user login ID.
Primary Position ID	Queries the operational application database to set the variable PRIMARY_POSTN_ID.
Warehouse Refresh Date	Sets several time-based variables such as CURRENT_YEAR.
ETL Run Date	Retrieves the ETL run date.
ETL Default Currency	Retrieves the default currency.

3.9.3 Creating Custom Indexes in Siebel Source Databases for Incremental Load Performance

To maximize performance for Siebel CRM, you can implement indexes using SQL files that are available in the \OracleBI\dwrep directory. Table 3–7 describes the SQL files that are appropriate for the specified applications.

Table 3–7 SQL files for Siebel transactional databases

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

The SQL files generate indexes on all S_.* tables that are being used by the preconfigured applications.

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

3.9.3.1 An Example of Change Capture SQL and Required Indexes

Change capture SQL generates the following SQL:

```
Insert into S_ETL_I_IMG_XX (ROW_ID, LAST_UPD)
AS
SELECT ROW_ID, LAST_UPD, MODIFICATION_NUM
From
S_XXX
WHERE LAST_UPD > 'LAST REFRESH_DATE - PRUNE DAYS'
AND NOT EXISTS
(
SELECT 'X' FROM S_ETL_R_IMAGE
WHERE S_ETL_R_IMAGE.ROW_ID = S_XXX.ROW_ID AND
S_ETL_R_IMG_XX.MODIFICATION_NUM = S_XXX.MODIFICATION_NUM AND
S_ETL_R_IMG_XX.LAST_UPD = S_XXX.LAST_UPD
)
```

Table 3–8 shows the indexes that, based on the SQL above, are created on the S_CONTACT table by the SQL generation scripts.

Table 3–8 Indexes Created on the S_CONTACT Table in Siebel transactional database

Index	Index Column
S_CONTACT_W1	LAST_UPD, ROW_ID_MODIFICATION_NUM
S_CONTACT_W11	LAST_UPD

3.9.4 Creating Custom Indexes in Oracle EBS Source Databases for Incremental Load Performance

Oracle EBS source database tables contain mandatory LAST_UPDATE_DATE columns, which are used by Oracle BI Applications for capturing incremental data changes. Some Oracle EBS source tables used by Oracle BI Applications do not have an index on the LAST_UPDATE_DATE column because the presence of the index can impede performance of the source application.

There are three categories of Oracle EBS tables as they relate to indexes on the LAST_UPDATE_DATE column:

- Category 1: Tables that do not have indexes on the LAST_UPDATE_DATE column but on which indexes can be created without impeding performance.
- Category 2: Tables that have indexes on LAST_UPDATE_DATE columns. These indexes were introduced in Oracle EBS release 12.
- Category 3: Tables that cannot have indexes on the LAST_UPDATE_DATE column because performance will be impeded in the Oracle EBS environment.

3.9.4.1 Creating Indexes for Category 1 Tables

The following DDL script creates custom indexes on the LAST_UPDATE_DATE column for Category 1 tables, that is, tables in all Oracle EBS releases that do not already have this index created and for which there are no known performance implications for creating such indexes.

You should run this DDL script if your source system is Oracle EBS release 11i or release 12 and you have experienced slow incremental extraction mapping performance while implementing specific subject areas.

Note: If your source system is Oracle EBS release 12, Oracle EBS release 11.5.10, Oracle EBS release 11.5.9 or lower and has been migrated to Oracle Applications Tablespace Model (OATM), then replace `<IDX_TABLESPACE>` with `APPS_TS_TX_IDX`.

The DDL script is as follows:

```
CREATE index AP.OBIEE_AP_INVOICE_PAYMENTS_ALL ON AP.AP_INVOICE_
PAYMENTS_ALL(LAST_UPDATE_DATE) tablespace <IDX_TABLESPACE>;

CREATE index AP.OBIEE_AP_PAYMENT_SCHEDULES_ALL ON AP.AP_PAYMENT_
SCHEDULES_ALL(LAST_UPDATE_DATE) tablespace <IDX_TABLESPACE>;

CREATE index AP.OBIEE_AP_INVOICES_ALL ON AP.AP_INVOICES_
ALL(LAST_UPDATE_DATE) tablespace <IDX_TABLESPACE>;

CREATE index GL.OBIEE_GL_JE_HEADERS ON GL.GL_JE_HEADERS (LAST_
UPDATE_DATE) tablespace <IDX_TABLESPACE>;

CREATE index ONT.OBIEE_OE_ORDER_HEADERS_ALL ON ONT.OE_ORDER_
HEADERS_ALL(LAST_UPDATE_DATE) tablespace <IDX_TABLESPACE>;

CREATE index PER.OBIEE_PAY_INPUT_VALUES_F ON PER.PAY_INPUT_
VALUES_F (LAST_UPDATE_DATE) tablespace <IDX_TABLESPACE>;

CREATE index PER.OBIEE_PAY_ELEMENT_TYPES_F ON PER.PAY_ELEMENT_
TYPES_F (LAST_UPDATE_DATE) tablespace <IDX_TABLESPACE>;

CREATE index PO.OBIEE_RCV_SHIPMENT_LINES ON PO.RCV_SHIPMENT_
LINES (LAST_UPDATE_DATE) tablespace <IDX_TABLESPACE>;

CREATE index PO.OBIEE_RCV_SHIPMENT_HEADERS ON PO.RCV_SHIPMENT_
HEADERS (LAST_UPDATE_DATE) tablespace <IDX_TABLESPACE>;

CREATE index AR.OBIEE_AR_CASH_RECEIPTS_ALL ON AR.AR_CASH_
RECEIPTS_ALL (LAST_UPDATE_DATE)

tablespace <IDX_TABLESPACE>;

CREATE index WSH.OBIEE_WSH_DELIVERY_DETAILS ON WSH.WSH_DELIVERY_
DETAILS (LAST_UPDATE_DATE) tablespace <IDX_TABLESPACE>;

CREATE index WSH.OBIEE_WSH_NEW_DELIVERIES ON WSH.WSH_NEW_
DELIVERIES (LAST_UPDATE_DATE) tablespace <IDX_TABLESPACE>;
```

Note:

- Make sure you use FND_STATS to compute statistics on the newly created indexes and update statistics on newly indexed table columns in the Oracle EBS database.
- All indexes created with the DDL in this section have the prefix OBIEE_. This prefix does not follow standard Oracle EBS index naming conventions. Therefore, Autopatch may fail during future upgrades. In such cases, the indexes with the OBIEE_ prefix should be dropped and Autopatch restarted.

3.9.4.2 Creating Indexes for Category 2 Tables

The following DDL creates custom indexes on the LAST_UPDATE_DATE column for Category 2 tables, that is, tables for which indexes were introduced for the LAST_UPDATE_DATE column in Oracle release 12.

You should run this DDL script if your source system is Oracle EBS release 11i.

Note: If your source system is Oracle EBS release 11.5.10, Oracle EBS release 11.5.9 or lower and has been migrated to Oracle Applications Tablespace Model (OATM), then replace `<IDX_TABLESPACE>` with `APPS_TS_TX_IDX`.

The DDL script is as follows:

```
CREATE index PO.RCV_TRANSACTIONS_N23 ON PO.RCV_TRANSACTIONS
(LAST_UPDATE_DATE) INITIAL 4K NEXT 2M MINEXTENTS 1 MAXEXTENTS 50
PCTINCREASE 0 INITRANS 2 MAXTRANS 255 PCTFREE 10 tablespace
<IDX_TABLESPACE>;

CREATE index PO.PO_DISTRIBUTIONS_N13 ON PO.PO_DISTRIBUTIONS_ALL
(LAST_UPDATE_DATE) INITIAL 4K NEXT 2M MINEXTENTS 1 MAXEXTENTS 50
PCTINCREASE 0 INITRANS 2 MAXTRANS 255 PCTFREE 10 tablespace
<IDX_TABLESPACE>;

CREATE index PO.PO_LINE_LOCATIONS_N11 ON PO.PO_LINE_LOCATIONS_
ALL (LAST_UPDATE_DATE) INITIAL 4K NEXT 2M MINEXTENTS 1
MAXEXTENTS 50 PCTINCREASE 0 INITRANS 2 MAXTRANS 255 PCTFREE 10
tablespace <IDX_TABLESPACE>;

CREATE index PO.PO_LINES_N10 ON PO.PO_LINES_ALL (LAST_UPDATE_
DATE) INITIAL 4K NEXT 4K MINEXTENTS 1 MAXEXTENTS 50 PCTINCREASE
0 INITRANS 2 MAXTRANS 255 PCTFREE 10 tablespace <IDX_
TABLESPACE>;

CREATE index PO.PO_REQ_DISTRIBUTIONS_N6 ON PO.PO_REQ_
DISTRIBUTIONS_ALL (LAST_UPDATE_DATE) INITIAL 4K NEXT 250K
MINEXTENTS 1 MAXEXTENTS 50 PCTINCREASE 0 INITRANS 4 MAXTRANS 255
PCTFREE 10 tablespace <IDX_TABLESPACE>;

CREATE index PO.PO_REQUISITION_LINES_N17 ON PO.PO_REQUISITION_
LINES_ALL (LAST_UPDATE_DATE) INITIAL 4K NEXT 250K MINEXTENTS 1
MAXEXTENTS 50 PCTINCREASE 0 INITRANS 4 MAXTRANS 255 PCTFREE 10
tablespace <IDX_TABLESPACE>;

CREATE index PO.PO_HEADERS_N9 ON PO.PO_HEADERS_ALL (LAST_UPDATE_
DATE) INITIAL 4K NEXT 1M MINEXTENTS 1 MAXEXTENTS 50 PCTINCREASE
0 INITRANS 2 MAXTRANS 255 PCTFREE 10 tablespace <IDX_
TABLESPACE>;

CREATE index PO.PO_REQUISITION_HEADERS_N6 ON PO.PO_REQUISITION_
HEADERS_ALL (LAST_UPDATE_DATE) INITIAL 4K NEXT 250K MINEXTENTS 1
MAXEXTENTS 50 PCTINCREASE 0 INITRANS 4 MAXTRANS 255 PCTFREE 10
tablespace <IDX_TABLESPACE>;

CREATE index AR.RA_CUSTOMER_TRX_N14 ON AR.RA_CUSTOMER_TRX_ALL
(LAST_UPDATE_DATE) INITIAL 4K NEXT 4M MINEXTENTS 1 MAXEXTENTS 50
PCTINCREASE 0 INITRANS 4 MAXTRANS 255 PCTFREE 10 tablespace
<IDX_TABLESPACE>;
```

Note: Make sure you use FND_STATS to compute statistics on the newly created indexes and update statistics on newly indexed table columns in the Oracle EBS database.

3.9.5 Running an ETL with a Non-English OLTP Data Source

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

To run an ETL with a non-English OLTP data source

1. In DAC, select File, then New Source System to display the New Source System Container dialog box.
2. Select the Create as a Copy of Existing Container radio button.
3. From the Existing Containers drop-down list, select the container that you want to copy, then click OK.
4. Go to the Design view.
5. Ensure that you have selected the correct container from the Containers drop-down list.
6. Select the Source System Parameters tab.
7. Use the Edit tab below the list of Source System Parameters to change the value of the following parameters in the list:

   - $$DFLT_LANG (Default Language) – For example, for a Japanese data source, change this value to JPN.
   - (optional) $$DFLT_COUNTRY (Default Country)
   - (optional) $$DFLT_CONTINENT (Default Continent)

   Tip: To find out the value to specify for the $$DFLT_LANG parameter, issue the query 'select VAL from S_SYS_PREF where SYS_PREF_CD=<*ETL value*>' against the OLTP database. For example, to find out the default ETL language, issue the following command:

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

8. Create a new ETL plan for the new source system container, and edit its parameters, as follows:

   a. Click the Execute tab.

   b. Click the Execution Plans subtab.

   c. Click New to create a new blank execution tab and use the subtabs below (for example, Subject Areas, Parameters, Ordered Tasks) to specify the execution plan details.

   d. Click Save.

9. Click Run Now to run the new ETL plan (or use the Schedule tab to specify when you want to run the new execution plan).

4

Installing and Setting Up Oracle BI Applications

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

This section explains how to install and set up the Oracle BI Applications components. It contains the following main topics:

- Section 4.15, "Configuring Relational and Application Connections in Informatica Workflow Manager"
- Section 4.16, "Configuring the Oracle BI Repository Connections"
- Section 4.17, "Configuring the SiebelUnicodeDB Custom Property"
- Section 4.18, "Setting Up DAC to Receive Email Notification"
- Section 4.19, "Additional Configuration Tasks"
- Section 4.20, "About Running A Full Load ETL"

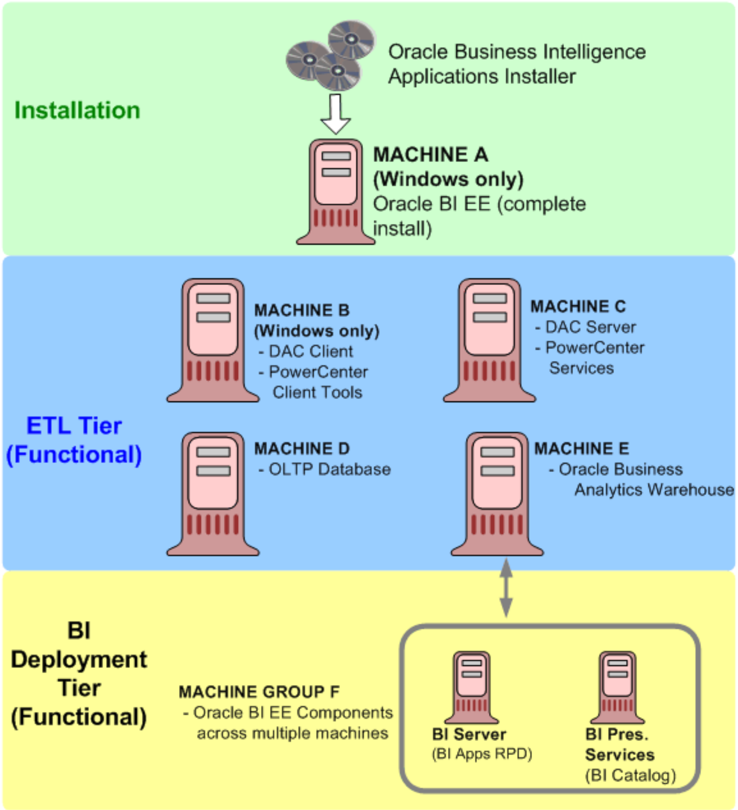
For information about supported Windows versions, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

4.1 About Oracle BI Applications Topologies

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

This section describes a typical topology for an Oracle BI Applications deployment, as illustrated in Figure 4–1.

Figure 4–1 Typical Topology for an Oracle BI Applications Deployment


Installation
Oracle Business Intelligence Applications Installer
MACHINE A (Windows only) Oracle BI EE (complete install)
ETL Tier (Functional)
MACHINE B (Windows only) - DAC Client - PowerCenter Client Tools
MACHINE C - DAC Server - PowerCenter Services
MACHINE D - OLTP Database
MACHINE E - Oracle Business Analytics Warehouse
BI Deployment Tier (Functional)
MACHINE GROUP F - Oracle BI EE Components across multiple machines
BI Server (BI Apps RPD)
BI Pres. Services (BI Catalog)

In Figure 4–1, note the following:

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

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

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

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

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

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

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

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

    Runs the DAC Client and Informatica PowerCenter Client Tools.
  - Machine C (Windows, UNIX, Linux)

    Runs the DAC Server and Informatica PowerCenter Services.
  - Machine D (Windows, UNIX, Linux)

    Hosts the transactional (OLTP) database.
  - Machine E (Windows, UNIX, Linux)

    Hosts the Oracle Business Analytics Warehouse database.
- **BI Deployment Tier (Functional)**

  The BI Deployment tier is used to deploy the business intelligence dashboards.
  - Machine Group F (Windows, UNIX, Linux)

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

4.2 High-Level Installation and Configuration Task List

This section contains a high-level task list of the tasks you must complete to install Oracle BI Applications. You must complete these tasks in the order listed below.

1. Before you start to install Oracle BI Applications and Informatica PowerCenter Services, do the following:

   a. Make sure that you satisfy the Informatica PowerCenter requirements that are specified in Section 4.3.2, "Informatica PowerCenter Requirements."

   b. Make sure that you satisfy the Code Page requirements that are specified in Section 4.3.3, "Code Page and Data Movement Requirements."

   c. Make sure that you perform the mandatory preinstallation tasks that are specified in Section 4.4, "Preinstallation Tasks." The preinstallation tasks are the following:

      - Section 4.4.1, "Create Databases for Oracle BI Applications and Informatica PowerCenter Components"

      - Section 4.4.2, "Install and Configure Database Connectivity Software"

      - Section 4.4.3, "Perform Prerequisites for Informatica PowerCenter Installation"

2. Run the Oracle BI Applications 7.9.6.1 installer on a Windows machine. For more information, see Section 4.5, "Running the Oracle BI Applications Installer on Windows."

   a. Make sure that a complete installation of Oracle Business Intelligence Enterprise Edition is installed on the machine on which you want to install Oracle BI Applications. For more information, see Section 4.3.1, "Oracle Business Intelligence Infrastructure Requirements."

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

3. Install Informatica PowerCenter Services and Client Tools version 8.6.1 and Hotfix 6 software. For more information, see Section 4.6, "Installing Informatica PowerCenter."

   > **Note:** See the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* for information about additional hotfixes or emergency bug fixes that may be required to support the current version of Informatica.

4. Restore the prebuilt Informatica Repository file provided with Oracle BI Applications. This file includes ETL repository objects, such as mappings, sessions, and workflows. For more information, see Section 4.7, "Restoring the Prebuilt Informatica Repository."

5. Configure the Informatica PowerCenter components for use with Oracle BI Applications. For more information, see Section 4.8, "Configuring Informatica PowerCenter Components."

6. Install the DAC Client and Server Platform. The DAC Platform is installed by the Oracle Data Warehouse Administration Console (DAC) installer. For more information, see Section 4.9, "Installing and Setting Up the DAC Platform."

7. Install the DAC metadata files. The DAC metadata files are installed by the Oracle BI Applications installer and must be copied to the machines hosting the DAC Client and Server. For more information, see Section 4.10, "Installing DAC Metadata Files."

8. Log into DAC and Import Metadata into the DAC Repository. For more information, see Section 4.11, "Logging into DAC for the First Time and Importing Metadata into the DAC Repository."

9. Create the Oracle Business Analytics Warehouse tables. For more information, see Section 4.12, "Creating the Oracle Business Analytics Warehouse Tables."
10. Configure the DAC Server. For more information, see Section 4.13, "Configuring the DAC Server."
11. Configure DAC Integration Settings. For more information, see Section 4.14, "Configuring DAC Integration Settings."
12. Create Relational Connections in Informatica Workflow Manager, as specified in Section 4.15, "Configuring Relational and Application Connections in Informatica Workflow Manager."
13. Configure the Oracle BI Repository connections, as specified in Section 4.16, "Configuring the Oracle BI Repository Connections."
14. Configure the SiebelUniCodeDB Custom Property, as specified in Section 4.17, "Configuring the SiebelUnicodeDB Custom Property."
15. Set up DAC to receive email notification. For more information, see Section 4.18, "Setting Up DAC to Receive Email Notification."
16. Perform required post-installation tasks. For more information, see Section 4.19, "Additional Configuration Tasks."

Note: After you have completed the steps listed above and completed any customizations that you need to make, you are ready to perform a full load of your OLTP data. For an example of performing a full load of OLTP data, see Section 4.20, "About Running A Full Load ETL." For detailed information about using DAC to perform ETL processes, see *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

4.3 Mandatory Requirements

This section includes mandatory requirements that you must satisfy before you can deploy Oracle BI Applications.

This section contains the following topics:

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

4.3.1 Oracle Business Intelligence Infrastructure Requirements

The Oracle BI Applications installer requires that you have already installed Oracle Business Intelligence Enterprise Edition on a Windows machine. The instance of Oracle Business Intelligence Enterprise Edition must be a complete installation; that is, you must install it using the Complete setup type option.

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

To determine the minimum version of Oracle Business Intelligence Enterprise Edition that is supported for this release of Oracle BI Applications, see the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

4.3.2 Informatica PowerCenter Requirements

Oracle BI Applications release 7.9.6.1 requires Informatica PowerCenter 8.6.1 with Hotfix 6. If you license Informatica PowerCenter with Oracle BI Applications, you install Informatica PowerCenter 8.6.1 and Hotfix 6 from the Informatica PowerCenter DVD provided with Oracle Business Intelligence Applications.

If you license Informatica PowerCenter separately and you do not have Informatica PowerCenter 8.6.1 and Hotfix 6, you must upgrade your Informatica PowerCenter license to 8.6.1 and Hotfix 6 before deploying Oracle BI Applications.

Note: See the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* for information about additional hotfixes or emergency bug fixes that may be required to support the current version of Informatica.

Informatica PowerCenter 8.6.1 has significant architecture changes from previous 7.x versions. Before installing Oracle BI Applications, Oracle recommends that you do the following:

- Read the Informatica PowerCenter documentation to familiarize yourself with the new architecture, components, and features. Informatica PowerCenter 8.6.1 documentation is included on the Informatica PowerCenter DVD provided with Oracle Business Intelligence Applications.
- Plan your topology carefully before installing Informatica PowerCenter Services.

Also, note the following about Informatica and DAC requirements:

- Informatica PowerCenter Client Tools and the DAC Client must be co-located on the same machine.
- PowerCenter Services and the DAC Server must be co-located on the same machine.
- Informatica PowerCenter Services runs on 32-bit or 64-bit platforms. For more information about these two platforms, see the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

About Changes in Informatica Terminology

Review this section if you are upgrading from an Informatica 7.x version to PowerCenter 8.6.1.

Table 4–1 Changes in Informatica Terminology

Term used in 7.1.x versions of Informatica PowerCenter	Term used in 8.x version of Informatica PowerCenter
Informatica Repository	Informatica Repository The Informatica Repository is managed by the Repository Service.
Informatica Repository Server	Not applicable. This component does not exist in Informatica PowerCenter 8.6.1. The Informatica Repository is managed by the Repository Service.
Informatica Server	Integration Services
Not applicable	Repository Service

4.3.3 Code Page and Data Movement Requirements

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

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

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

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

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

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

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

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

- **Determining the Data Movement Mode**. Before you deploy Oracle BI Applications, you must determine what data movement mode to use (ASCII or Unicode) for the PowerCenter Integration Service. The Character Data Movement Mode is an Informatica PowerCenter Integration Service option that you choose based on whether you want to move single-byte or multi-byte data. Choose Unicode if non-ASCII characters have to be moved. Otherwise, choose ASCII. The Data Movement Mode option is configurable and can be reset after installation.

  To set the Data Movement Mode, log into Informatica PowerCenter Administration Console, select the Integration Service, then click the Properties tab, then display the General Properties tab, and set the DataMovementMode value.

- **Determining code pages for Informatica PowerCenter components.** In addition to source and target code pages, Informatica PowerCenter uses code pages for PowerCenter Client, the Integration Service, the Informatica Repository, and PowerCenter command line programs (pmcmd and pmrep, which are used by

DAC to communicate with PowerCenter Services). Carefully review "Chapter 22: Understanding Globalization," in the *Informatica PowerCenter Administrator Guide*, particularly if your environment requires the Data Mode Movement mode to be set to UNICODE. The section discusses code page compatibility and code page requirements for the Informatica components.

- **Setting environment variables.** You must manually set the appropriate environment variables for UNIX environments. In addition, the Informatica installer requires the appropriate locale to be set on UNIX machines. Use LANG, LC_CTYPE or LC_ALL to set the UNIX code page. For more information, see the topic titled "Configuring Environment Variables," in "Chapter 2: Before You Install," in the *Informatica PowerCenter Installation Guide*.

  If your environment uses Oracle or DB2 database, you need to set the environment variables NLS_LANG or DB2CODEPAGE. For information on how to set these environment variables see Section 4.4.2.1, "How to Set the NLS_LANG Environment Variable for Oracle Databases" and Section 4.4.2.2, "How to Set the DB2CODEPAGE Environment Variable for DB2 Databases."

- Configuring Relational Connections. When you configure relational connections in the Workflow Manager, choose a code page that is compatible with the code page of the database client. If you set a database environment variable to specify the language for the database, ensure the code page for the connection is compatible with the language set for the variable. For example, if you set the NLS_LANG environment variable for an Oracle database, ensure that the code page of the Oracle connection is identical to the value set in the NLS_LANG variable.

For more information about data movement modes, refer to the Informatica PowerCenter documentation. Informatica PowerCenter 8.6.1 documentation is included on the Informatica PowerCenter DVD provided with Oracle BI Applications.

4.4 Preinstallation Tasks

This section explains the mandatory preinstallation tasks that you must perform for an Oracle BI Applications deployment. It contains the following topics:

- Section 4.4.1, "Create Databases for Oracle BI Applications and Informatica PowerCenter Components"
- Section 4.4.2, "Install and Configure Database Connectivity Software"
- Section 4.4.3, "Perform Prerequisites for Informatica PowerCenter Installation"

4.4.1 Create Databases for Oracle BI Applications and Informatica PowerCenter Components

Before you install Oracle BI Applications, the Data Warehouse Administration Console (DAC), and Informatica PowerCenter, use your target database tool to create database instances to hold the following:

- DAC Repository
- Domain Configuration Database
- Informatica Repository
- Oracle Business Analytics Warehouse

Note the following points:

- The transactional (OLTP) database user that is registered in DAC should be the database table owner. Alternatively, at a minimum, the user registered in DAC must have read privileges on the transactional database for all tables and aliases as well as rights to create triggers and views on all tables and aliases.
- For efficient ETL, DAC and Informatica components utilize multiple connections to the Oracle Business Analytics Warehouse. The Oracle Business Analytics Warehouse database must allow for a minimum of 100 connections to be made by DAC and Informatica. In addition, ensure that these connections are not allowed to time out. Consult with your network administrator and DBA for information on how to ensure these requirements.
- Make sure that the Oracle Business Analytics Warehouse instance is granted the SSE_ROLE. For more information, see Section 4.4.1.1, "How to Create the SSE Role."
- You can store the DAC Repository and the Oracle Business Analytics Warehouse in separate databases or in the same database.
- The DAC and Informatica repositories are not supported on all database platforms. For information about which database platforms are supported, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.
- For database requirements for the Informatica PowerCenter components, see the topic titled, "Verifying Prerequisites," in "Chapter 2: Before You Install," in the *Informatica PowerCenter Installation Guide*.
- You must create the Informatica PowerCenter domain configuration database before you run the PowerCenter installer. For more information see the topic titled, "Creating the Domain Configuration Database," in "Chapter 2: Before You Install," in the *Informatica PowerCenter Installation Guide*.

  The Informatica documentation is included on the Informatica PowerCenter DVD provided with Oracle BI Applications.

4.4.1.1 How to Create the SSE Role

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

Note: The transactional (OLTP) database user that is registered in DAC should be the database table owner. Alternatively, at a minimum, the user registered in DAC must have read privileges on the transactional database for all tables and aliases as well as rights to create triggers and views on all tables and aliases.

To create the SSE role

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

   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 from the data warehouse database, the SSE_ROLE must have the following additional privileges:
  - SELECT
  - INSERT
  - 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 the alternative SSE role name SSEROLE if required. To specify a different role name, modify the 'set GRANTEE="SSEROLE"' line in the OracleBI\dwrep\createwtables.bat file.

4.4.2 Install and Configure Database Connectivity Software

> **Note:** The Informatica documentation referred to in this section is available on the Informatica PowerCenter DVD that is provided with Oracle BI Applications.

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

The machine that hosts the PowerCenter Integration Services requires connectivity to the Oracle Business Analytics Warehouse (target) database and transactional (source) database(s). For information about installing and configuring native connectivity software for Integration Services, see the topic titled, "Integratioin Service Connectivity," in "Chapter 10: Integration Service Architecture," in the *Informatica PowerCenter Administrator Guide*. Also see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* for information on certified database connectivity software used by Integration Services to connect to the Oracle Business Analytics Warehouse.

The machine that hosts the PowerCenter Repository Service requires native connectivity to communicate with the Informatica Repository database. For information about installing and configuring the required software, see the topic titled, "Database Connect Strings," in the section, "Creating the Repository Service," in "Chapter 7: Creating and Configuring the Repository Service," in the *Informatica PowerCenter Administrator Guide*.

For additional information about database connectivity for Integration Services and Repository Services, see the topic titled, "Install and Configure Database Client Software," in "Chapter 2: Before You Install," in the *Informatica PowerCenter Installation Guide*.

The machine or machines that host the DAC Client and DAC Server require connectivity to the Oracle Business Analytics Warehouse (target) database, transactional (source) database(s), and the DAC Repository database. For instructions on configuring connectivity for DAC, see Section 4.9.3, "Installing JDBC Drivers for DAC Database Connectivity" and Section 4.9.4, "Creating ODBC Database Connections for the DAC Client"

Notes

- For Oracle databases, set the NLS_LANG environment variable on the machines where the Oracle client is installed, as described in Section 4.4.2.1, "How to Set the NLS_LANG Environment Variable for Oracle Databases".
- For DB2 databases, set the DB2CODEPAGE environment variable on the machines where the DB2 client is installed, as described in Section 4.4.2.2, "How to Set the DB2CODEPAGE Environment Variable for DB2 Databases".

4.4.2.1 How to Set the NLS_LANG Environment Variable for Oracle Databases

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

> **Note:** You need to set the NLS_LANG environment variable on each machine that has the Oracle client installed.

To set the NLS_LANG environment variable for Oracle databases

1. Determine the NLS_LANG value.

   a. In the data warehouse database, run the following command:

   ```
   SELECT * FROM V$NLS_PARAMETERS
   ```

   b. Make a note of the NLS_LANG value, which is in the format [NLS_LANGUAGE]_[NLS_TERRITORY].[NLS_CHARACTERSET].

   For example: `American_America.UTF8`

2. For Windows:

   a. Navigate to Control Panel > System and click the Advanced tab. Click Environment Variables.

   b. In System variables section, click New.

   c. In the Variable Name field, enter `NLS_LANG`.

   d. In the Variable Value field, enter the NLS_LANG value that was returned in Step 1.

   The format for the NLS_LANG value should be [NLS_LANGUAGE]_[NLS_TERRITORY].[NLS_CHARACTERSET].

   For example: `American_America.UTF8`.

> **Note:** The NLS_LANG character set should reflect the setting of the operating system character set of the client. For example, if the database character set is AL32UTF8 and the client is running on a Windows operating system, then you should not set AL32UTF8 as the client character set in the NLS_LANG parameter because there are no UTF-8 WIN32 clients. Instead, the NLS_LANG setting should reflect the code page of the client. For example, on an English Windows client, the code page is 1252. An appropriate setting for NLS_LANG is AMERICAN_AMERICA.WE8MSWIN1252.
>
> Setting NLS_LANG correctly allows proper conversion from the client operating system character set to the database character set. When these settings are the same, Oracle assumes that the data being sent or received is encoded in the same character set as the database character set, so character set validation or conversion may not be performed. This can lead to corrupt data if the client code page and the database character set are different and conversions are necessary

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

   ```
   setenv NLS_LANG <NLS_LANG>
   ```

   For example: `setenv NLS_LANG` `American_America.UTF8`.

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

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

4.4.2.2 How to Set the DB2CODEPAGE Environment Variable for DB2 Databases

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

To set the DB2CODEPAGE environment variable for DB2 databases

1. Determine the DB2CODEPAGE value.

   a. Connect to the Source database, using the following command:

      ```
      SELECT CODEPAGE FROM SYSCAT.DATATYPES WHERE TYPENAME = 'VARCHAR'
      ```

   b. Make a note of the result.

      For example: 1208

2. For Windows:

   a. Navigate to Control Panel > System and click the Advanced tab. Click Environment Variables.

   b. In System variables section, click New.

   c. In the Variable Name field, `DB2CODEPAGE`.

   d. In the Variable Value field, enter the value that was returned in Step 1.

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

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

For example: `setenv 1208`.

4.4.3 Perform Prerequisites for Informatica PowerCenter Installation

Before you run the Informatica PowerCenter installer, you must perform the prerequisites that are described in the chapter, "Before You Install," in the *Informatica PowerCenter Installation Guide*.

4.5 Running the Oracle BI Applications Installer on Windows

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

This section explains how to install the Oracle BI Applications files using the Oracle BI Applications Installation Wizard. When you run the wizard, the Oracle BI Applications files are installed into the existing Oracle Business Intelligence infrastructure directory (for example, <DRIVE>:\OracleBI\).

> **Note:** If you have a previous release of Oracle BI Applications installed, you must uninstall this release before you run the installer for Oracle BI Applications release 7.9.6.1. If you do not uninstall the previous version, some release 7.9.6.1 directories will not be correctly installed. (Make a back-up of your Oracle BI Repository, Presentation Catalog, and DAC and dwrep directories before you uninstall the previous release.)
>
> You can use the InstallShield Wizard to uninstall Oracle BI Applications. To access the InstallShield Wizard, from the Control Panel, select Add or Remove Programs.

To install Oracle BI Applications on Windows

1. Access the installation files on the installation DVD, 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 the following screens in console installation mode. Instead, you enter input as plain text in the terminal window when prompted.

2. In the Welcome page, click Next.
3. Enter or browse for the location for the Oracle Business Intelligence infrastructure (for example, <drive>:\OracleBI\) and for the Oracle Business Intelligence data (for example, <drive>:\OracleBIData\). Click Next.
4. Select the Oracle BI Applications that you want to install, and then click Next.

   To determine the applications you should install for the Oracle BI Applications licenses you have purchased, see the *Oracle Business Intelligence Applications*

Licensing and Packaging Guide. This guide is part of the Oracle Business Intelligence Media Pack.

Once you have installed applications, you can deploy them for one or more source systems selectively when you create the DAC Repository in a later step. For more information, see Section 4.11.2, "Importing Metadata into the DAC Repository."

5. 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 Business Intelligence infrastructure installation.

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

6. Click Finish.

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

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

- The dwrep folder.
- The dwrep\DAC_metadata folder.
- The Oracle BI Applications repository file named OracleBIAnalyticsApps.rpd in the OracleBI\server\Repository folder.
- The Presentation Catalog in the OracleBIData\web\catalog folder. The Presentation Catalog is provided as a zipped file named EnterpriseBusinessAnalytics.zip and will need to be unzipped before it can be used.
- The Upgrade folder containing repository files from previous versions of Oracle BI Application and Siebel Analytics Application releases.

Notes

- The credentials required to log into the OracleBIAnalyticsApps.rpd file using the Oracle Business Intelligence Administration Tool are the following:
  - Username: Administrator
  - Password: SADMIN

  > **Note:** You should change the default password. Use the Oracle Business Intelligence Administration Tool to do so. For instructions, see the *Oracle Business Intelligence Server Administration Guide*.

- For information on configuring the Oracle BI Repository, see the *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*.

- Refer to the Oracle Business Intelligence Enterprise Edition documentation for more information on working with the Oracle BI Repository and Presentation Catalog.

4.6 Installing Informatica PowerCenter

This section contains the following topics:

- Section 4.6.1, "Installing Informatica PowerCenter Client Tools (Windows)"
- Section 4.6.2, "Installing Informatica PowerCenter Services"
- Section 4.6.3, "Creating the Informatica Repository Service"
- Section 4.6.4, "Creating the Informatica Integration Service"

Notes

- You need to determine the machines that will host Informatica PowerCenter Client Tools and Informatica PowerCenter Services. The PowerCenter Services can be installed on UNIX or on Windows. The PowerCenter Client Tools must be installed on Windows.
- You need to co-locate the DAC Client with the PowerCenter Client Tools.
- You need to co-locate the DAC Server with the PowerCenter Services.
- DAC produces parameter files that are used by Informatica. If an execution plan fails in DAC and you want to debug the workflow by running it directly from Informatica, then the parameter file produced by DAC should be visible to Informatica. This is one reason for the requirement to co-locate the DAC and Informatica components as stated above.
- You must meet all requirements and complete preinstallation tasks as described in Section 4.3, "Mandatory Requirements" and in the chapter 'Before You Install' of the *Informatica PowerCenter Installation Guide*. Informatica PowerCenter 8.6.1 documentation is included on the Informatica PowerCenter DVD provided with Oracle BI Applications.
- When you install PowerCenter Services and Client Tools version 8.6.1 and Hotfix 6, use the Informatica PowerCenter installation documentation in conjunction with the steps provided in the following sections. The *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users* summarizes the steps and assumes a single-machine installation for the PowerCenter Services components. See the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* for information about additional hotfixes or emergency bug fixes that may be required to support the current version of Informatica.

> **Note:** If you have licensed Informatica PowerCenter separately from Oracle BI Applications, skip this section and go to section Section 4.7, "Restoring the Prebuilt Informatica Repository."
>
> Make sure that you have installed the required version of Informatica PowerCenter (for more information, see Section 4.3.2, "Informatica PowerCenter Requirements").

- If you are using a Windows Vista operating system, note the following:

  The DAC Client uses the Informatica pmrep and pmcmd command line programs when communicating with Informatica PowerCenter. The installation of

PowerCenter Client Tools provides the pmrep executable file. The installation of PowerCenter Services provides the pmcmd executable file. The DAC installer performs post-install configuration tasks related to the pmrep and pmcmd executable files. Therefore, you must run the run the PowerCenter Services installer on the Windows Vista machine where you will install the DAC Client. This installation of Informatica PowerCenter Services will be a non-functional installation because Informatica does not support PowerCenter Services on Windows Vista.

When installing PowerCenter Services on Windows Vista, you might receive an error message similar to the following. You can ignore any such error messages.

```
Use the error below and catalina.out and node.log in the
server/tomcat/logs directory on the current machine to get
more information. EXITCODE: S" Select Retry to continue the
installation.
```

4.6.1 Installing Informatica PowerCenter Client Tools (Windows)

This section explains how to install Informatica PowerCenter 8.6.1 Client Tools and Hotfix 6 for an Oracle BI Applications deployment. For detailed generic information about installing Informatica 8.6.1, refer to the *Informatica PowerCenter Installation Guide*, and related documentation, which is included on the Informatica PowerCenter DVD provided with Oracle BI Applications. See the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* for information about additional hotfixes or emergency bug fixes that may be required to support the current version of Informatica.

To install Informatica PowerCenter Client Tools on Windows

1. Run the program install.exe from the PowerCenter_8.6.1_SE_for_Windows_32bit on the Informatica PowerCenter DVD provided with Oracle BI Applications.
2. Follow the instructions on the Informatica PowerCenter Client 8.6.1 installation wizard.
3. Install Informatica PowerCenter Hotfix 6 from the PowerCenter_8.6.1_HotFix6_for_Windows_32bit folder on the Informatica PowerCenter DVD provided with Oracle BI Applications.
4. Follow the instructions on the Informatica PowerCenter Client 8.6.1 installation wizard.

4.6.2 Installing Informatica PowerCenter Services

This section explains how to install Informatica PowerCenter Services 8.6.1 and Hotfix 6 for an Oracle BI Applications deployment. For detailed generic information about installing Informatica 8.6.1, refer to the *Informatica PowerCenter Installation Guide* and related documentation, which is included on the Informatica PowerCenter DVD provided with Oracle BI Applications. See the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* for information about additional hotfixes or emergency bug fixes that may be required to support the current version of Informatica.

Note: If you have purchased the 'Informatica OEM PowerCenter ETL Server and PowerConnect Adapters' license from Oracle, use the Oracle_All_OS_Prod.key license key file. This file is located in the DVD root directory of the Informatica PowerCenter DVD that is provided with Oracle BI Applications.

Note: On Windows, do not install Informatica in a directory that has a space in the directory name. For example, do not install in D:\Program Files\INFA. If you install into a directory with a space in the name, you will cause errors in DAC.

To install Informatica PowerCenter Services 8.6.1 and Hotfix 6

1. Start the Informatica PowerCenter Services installer.

   To start the installer on Windows, run the install.bat file from the PowerCenter_8.6.1_SE_for_Windows_<bit mode> folder on the Informatica PowerCenter DVD that is provided with Oracle BI Applications.

   To start the installer on UNIX or Linux, run install.sh at a shell command line. The PowerCenter 8.6.1 installation files are located in the folder PowerCenter_8.6.1_SE_<OS>_<bit mode> on the Informatica PowerCenter DVD provided with Oracle Business Intelligence Applications.

   On UNIX or Linux, select the option to use the GUI mode installation.

2. Follow the instructions on the Informatica PowerCenter Services installation wizard.
3. Stop any Informatica services that are running.
4. Install the Informatica PowerCenter Hotfix 6 from the PowerCenter_8.6.1_HotFix6_for_<OS>_<bit mode> folder.
5. Follow the instructions on the Informatica PowerCenter Services installation wizard.

4.6.3 Creating the Informatica Repository Service

Follow this procedure to create the Informatica Repository Service.

For detailed information about creating the Repository Service, see the topic titled, "Creating the Repository Service," in "Chapter 7: Creating and Configuring the Repository Service," in the *Informatica PowerCenter Administrator Guide*, which is included on the Informatica DVD provided with Oracle BI Applications.

To create the Informatica Repository Service

1. Open the Informatica PowerCenter Administration Console.
2. On the left side of the Administration Console, click Create, and then select Create Repository Service.
3. In the Create New Repository Service dialog box, specify the appropriate values for the required properties. For a description of the required properties, see "Creating the Repository Service," in "Chapter 7: Creating and Configuring the Repository Service," in the *Informatica PowerCenter Administrator Guide.*

   In particular, note the following properties:

a. For the Code Page, select a code page that is compatible with the code page of the machine on which you installed Informatica PowerCenter. For more information, refer to the Informatica PowerCenter documentation and to Section 4.3.3, "Code Page and Data Movement Requirements."

   Note: The code page of the Repository Service cannot be changed once the Repository Service has been created.

b. For the Creation Mode, make sure you select Create New Repository Content.

4.6.4 Creating the Informatica Integration Service

Follow this procedure to create the Informatica Integration Service.

For detailed information about creating the Integration Service, see the topic titled, "Creating an Integration Service," in "Chapter 9: Creating and Configuring the Integration Service," in the *Informatica PowerCenter Administrator Guide*, which is included on the Informatica DVD provided with Oracle BI Applications.

To create the Informatica Integration Service

1. Open the Informatica PowerCenter Administration Console.
2. On the left side of the Administration Console, click Create, and then select Create Integration Service.
3. In the Create New Integration Service dialog box, specify the appropriate values for the required properties.

   For a description of the required properties, see the topic titled, "Creating an Integration Service," in "Chapter 9: Creating and Configuring the Integration Service," in the *Informatica PowerCenter Administrator Guide*.

4.7 Restoring the Prebuilt Informatica Repository

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

This section includes the following topics:

- Section 4.7.1, "Restoring the Prebuilt Informatica Repository for Environments in English"
- Section 4.7.2, "Restoring the Prebuilt Informatica Repository on a Non-English Operating System"

4.7.1 Restoring the Prebuilt Informatica Repository for Environments in English

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

To load the prebuilt Oracle_BI_DW_Base.rep or Oracle_BI_DW_Teradata.rep repository into Informatica

1. Copy the file Oracle_BI_DW_Base.rep (or Oracle_BI_DW_Teradata.rep if your Oracle Business Analytics Warehouse is on Teradata) from the OracleBI\dwrep\Informatica\Repository directory on the machine where the

Oracle BI Applications installer was run to the following folder on the machine where Informatica PowerCenter Services has been installed:

- On Windows copy the file to the \Informatica PowerCenter8.6.1\server\infa_shared\Backup directory.
- On UNIX or Linux copy the file to $Informatica/PowerCenter8.6.1/server/infa_shared/Backup directory.

2. In Informatica PowerCenter Administration Console, select the Repository Service that was created in the procedure in Section 4.6.3, "Creating the Informatica Repository Service."
3. In the General Properties area of the Properties tab, make sure the OperatingMode value is Exclusive.

   To change the OperatingMode value, click Edit, and then select a new value from the drop-down list. Click OK to exit edit mode.
4. Choose Actions, then Delete Contents.
5. At the Delete Contents for <repository name> dialog box, enter the repository username and password (for example, Administrator\Administrator), then click OK.
6. Choose Actions > Restore Contents.
7. At the Restore Contents dialog box, select Oracle_BI_DW_Base.rep (or Oracle_BI_DW_Teradata.rep for Teradata installations) from the Select Backup File drop-down list.
8. Select the Restore as New check box.
9. Click OK to start the restore process.
10. When the restore process is complete, click Close (or click Save first to save the logging information).

    When the restore is complete (the process typically takes approximately ten to twenty minutes), you will see a 'Success' message.
11. When a repository is restored, the repository becomes a standalone repository. After restoring the repository, you need to promote it to a global repository.

    For instructions, see the topic titled, "Promoting a Local Repository to a Global Repository," in "Chapter 8: Managing the Repository," in *Informatica PowerCenter Administrator Guide.*
12. Change the OperatingMode value to Normal.
    a. Go to the Properties tab.
    b. In the General Properties area, click Edit.
    c. Click the OperatingMode drop-down list, and select Normal.
13. If prompted, enter the repository username and password.

Note: The prebuilt Oracle_BI_DW_Base and Oracle_BI_DW_Teradata repository files are versioned from Oracle BI Applications release 7.9.3 and higher. For more information about the versioned repositories, see Appendix F, "About the Versioned Informatica Repository."

4.7.2 Restoring the Prebuilt Informatica Repository on a Non-English Operating System

If Informatica PowerCenter Services is installed on a non-English version of the operating system, you must use the command line to restore the prebuilt Informatica Repository provided with Oracle BI Applications.

To restore the prebuilt Informatica Repository on a non-English operating system

1. Open a Command window.
2. Enter the following command to connect to the repository:

   ```
   Pmrep connect -r <RepositoryName> -d <Domain>
   ```

3. Enter the following command to restore the repository:

   ```
   PmRep restore -u <domain_user_name> -p <domain_user_password> -i
   <input_file_name> -n
   ```

where the `input_file_name` is the name of the prebuilt repository file.

4.8 Configuring Informatica PowerCenter Components

This section explains how to configure Informatica PowerCenter Services for use with Oracle BI Applications. It contains the following topics:

- Section 4.8.1, "Copying Source Files and Lookup Files"
- Section 4.8.2, "Setting PowerCenter Integration Services Relaxed Code Page Validation"
- Section 4.8.3, "Setting PowerCenter Integration Services Custom Properties"
- Section 4.8.4, "Creating the Repository Administrator User in the Native Security Domain"

4.8.1 Copying Source Files and Lookup Files

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

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

- The default Informatica directory for source files is <drive>:\Informatica\PowerCenter8.6.1\server\infa_shared\SrcFiles. You can confirm the source file directory for your environment by launching the PowerCenter Administration Console and going to the General Properties area of the Processes tab. The parameter name for the source file directory is $PMSourceFileDir.

  Note: The value of the $PMSourceFileDir parameter must match the DAC system property called InformaticaParameterFileLocation. You will set this property later in the set up process, using the instructions in Section 4.14.1, "Setting DAC System Properties."

- The default Informatica directory for lookup files is <drive>:\Informatica\PowerCenter8.6.1\server\infa_shared\LkpFiles. You can confirm the lookup file directory for your environment by launching the PowerCenter Administration Console and going to the General Properties area of the Processes tab. The parameter name for the source file directory is $PMLookupFileDir

To copy source files and lookup files to Informatica PowerCenter Services on Windows

1. Copy the source files in \OracleBI\dwrep\Informatica\SrcFiles on the machine where the Oracle BI Applications installer was run to the source files directory on the PowerCenter Services machine, for example \Informatica PowerCenter8.6.1\server\infa_shared\SrcFiles.
2. Copy the lookup files in \OracleBI\dwrep\Informatica\LkpFiles on the machine where the Oracle BI Applications installer was run to the lookup files directory on the PowerCenter Services machine, for example Informatica PowerCenter8.6.1\server\infa_shared\LkpFiles.

To copy source files and lookup files to Informatica PowerCenter Services on UNIX or Linux

1. Copy the source files in \OracleBI\dwrep\Informatica\SrcFiles on the Windows machine where the Oracle BI Applications installer was run to the source files directory on the PowerCenter Services machine, for example $Informatica/PowerCenter8.6.1/server/infa_shared/SrcFiles.
2. Copy the lookup files in \OracleBI\dwrep\Informatica\LkpFiles on the Windows machine where the Oracle BI Applications installer was run to the lookup files directory on the PowerCenter Services machine, for example $Informatica/PowerCenter8.6.1/server/infa_shared/LkpFiles.

4.8.2 Setting PowerCenter Integration Services Relaxed Code Page Validation

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

To configure Informatica PowerCenter Integration Services for relaxed code page validation

1. Log in to Informatica PowerCenter Administration Console.
2. Select the Integration Service.
3. Select the Properties tab.
4. In the Configuration Properties area, click Edit.
5. Deselect the 'ValidateDataCodePages' check box.

4.8.3 Setting PowerCenter Integration Services Custom Properties

Follow this procedure to set PowerCenter Integration Services custom properties.

To set Informatica PowerCenter Integration Services Custom Properties

1. In Informatica PowerCenter Administration Console, select the Integration Service.
2. Click the Properties tab.
3. In the Custom Properties area, click Edit.
4. Create a custom property by clicking Add to display new Name and Value fields.
5. For each of the custom properties in the table below, click Add and add a new custom property with an appropriate value:

Table 4–2 Custom Properties for Informatica PowerCenter Integration Services

Custom Properties Name	Custom Properties Value	Notes
ServerPort	<Server Port Number>	For <Server port number>, enter the number of an available port. For example, 4006.This custom property configures Informatica PowerCenter Services to listen on <Server port number>. DAC communicates with the PowerCenter Integration Services service using this port.
overrideMpltVarWithMapVar	Yes	Enables Informatica to evaluate parameters within mapplets.
DisableDB2BulkMode	Yes	Add this custom property and set value to Yes if your Oracle Business Analytics Warehouse is on a DB2/390 or a DB2 UDB database.

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

4.8.4 Creating the Repository Administrator User in the Native Security Domain

For DAC to be able to access Informatica and perform tasks in pmcmd and pmrep command line programs, DAC must log in to Informatica as an Informatica Repository Administrator user. This user must be configured in the native security domain.

You need to create such a Repository Administrator user, or , if your security policies allow, you can use the default Administrator user (whose privileges include Repository administration) for DAC connectivity to Informatica.

For more information on creating users and Informatica security domains, see "Chapter 4: Managing Users and Groups," in the *Informatica PowerCenter Administrator Guide*.

To create a Repository Administrator defined in the native security domain

1. Log in to the PowerCenter Administration Console as Administrator.
2. Navigate to the Security page by clicking the Configure Security icon in the top, right corner of the page.
3. On the Security page, click Create User, or select an existing user defined in the native domain.

   Note: Do not create this user in an LDAP domain or select an existing user from an LDAP domain.
4. On the Privileges tab of the new or existing user, click the appropriate Repository Service.
5. In the Edit Roles and Privileges page, expand the Repository Service, and under System-Defined Roles, select the Administrator role check box.
6. Click OK.

4.9 Installing and Setting Up the DAC Platform

This section provides instructions for installing the DAC Client and Server as well as installing and creating the required database connections. For information about the version of the DAC platform supported with Oracle BI Applications, see the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

The DAC installer installs the DAC Client and Server on Windows and the DAC Server on Linux. Oracle does not provides an installer for DAC on UNIX.

To install a DAC Server on UNIX, you need to copy the \dac folder from the Windows machine on which the DAC Client is installed to a UNIX machine.

After you install the DAC platform, you then need to install JDBC drivers for DAC database connectivity and create ODBC database connections for the DAC Client.

This section includes the following topics:

- Section 4.9.1, "Installing DAC Using the DAC Installer"
- Section 4.9.2, "Installing the DAC Server on UNIX"
- Section 4.9.3, "Installing JDBC Drivers for DAC Database Connectivity"
- Section 4.9.4, "Creating ODBC Database Connections for the DAC Client"
- Section 4.9.5, "Installing pmcmd and pmrep to Enable Communication Between Informatica PowerCenter and DAC"

4.9.1 Installing DAC Using the DAC Installer

You can run the DAC installer to install DAC in the following configurations:

- DAC Client and Server on Windows
- DAC Server on Linux

Note:

- The DAC Client can only be installed on Windows.
- The DAC Client must be installed on the machine where Informatica PowerCenter Client Tools was installed.
- The DAC Server must be installed on the machine where Informatica PowerCenter Services was installed.
- You must install Informatica before you install DAC.
- The correct version of the JDK is installed by the DAC installer.
- The DAC installer installs DAC in the \orahome\10g\bifoundation\dac directory.

To install DAC using the DAC installer

1. Access the installation files on the Oracle Business Intelligence Data Warehouse Administration Console DVD, and start the DAC installer by double-clicking the setup.exe file.
2. Follow the instructions on the installation wizard, as described in the following table:

Page	Your Action	Notes
Welcome	Click Next to proceed.	None.

Page	Your Action	Notes
Specify Installation Location	Enter the absolute path for the location where you want to install DAC or use the Browse button to select an existing location. Click Next.	The installation directory path can contain alphanumeric, underscore (_), hyphen (-) or dot (.) characters and must begin with an alphanumeric character. The directory in which you install DAC is referred to as the Oracle Home location.
Install Component	Click Next.	This screen is read only. If you are running the installer on Windows, the DAC Client and Server will be installed. If you are running the installer on Linux, the DAC Server will be installed.
Select Informatica Version	Select Informatica version 8.x, and then click Next.	Oracle BI Applications 7.9.6.1 requires Informatica version 8.6.1. Therefore, you must select the option Informatica version 8.x.
Select Informatica Location	Enter or browse for the location of the following: ■ Informatica PowerCenter Services ■ Informatica PowerCenter domain file	None.
Pre-requisite Checks	If the checks pass, click Next to proceed with the installation	The DAC installer automatically performs pre-requisite checks on your system. The Pre-requisite Checks screen indicates the progress of the checks. If a check fails, you can click Retry to run the check again, or click Continue to proceed with the installation.
Installation Summary	Click Install to begin the installation.	None.
Installation Progress	Click Next.	None.
Installation Completed	Click Finish.	None.

DAC Configuration Handled by the DAC Installer

The DAC installer handles the following configuration:

- In the config.bat file, the DAC installer configures the JAVA_HOME and DAC_HOME variables.
- In the dac_env.bat file, the DAC installer creates an environment variable named INFA_DOMAINS_FILE and sets the value to the directory path of the domans.infa file.
- The DAC installer adds the directory path to Informatica PowerCenter binaries to the PATH environment variable.

Note: If you reinstall or upgrade Informatica PowerCenter without reinstalling DAC, you need to manually set these environment variables.

4.9.2 Installing the DAC Server on UNIX

Oracle does not provide an installer for DAC on UNIX.

To install a DAC Server on UNIX, you need to copy the \dac folder from the Windows machine on which the DAC Client is installed to a UNIX machine, as described in the procedure below.

Note: You can also use this procedure to install the DAC Server on Linux.

When installing a DAC Server, note the following prerequisites:

- On the UNIX machine, make sure the following are installed:
  - JDK version 1.6.0 or higher
  - Zip and unzip utility
- Make sure that Informatica PowerCenter Services is co-located on the machine that will host the DAC Server. For more information, see Section 4.6, "Installing Informatica PowerCenter."
- Make sure that Informatica has been configured according to the instructions in Section 4.8, "Configuring Informatica PowerCenter Components."

To install the DAC Server on a UNIX machine

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

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

2. On the machine where the DAC Client is installed, copy the \dac directory (that is, the directory installed by the DAC installer in \orahome\10g\bifoundation) to the temporary directory (for example, \OracleBI_UNIX\).

3. From the \dac directory in the temporary directory, remove the \export and \icons subfolders.

   These folders do not need to be copied to the machine hosting the DAC Server.

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

5. Copy the zip file to the target UNIX machine.

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

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

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

   Shell scripts are provided in the /dac directory. After copying these files to a UNIX machine and before using them, you might need to use a MS-DOS to UNIX conversion tool to convert the script files to UNIX format (that is, remove the carriage return and line feed characters). Alternatively, you can manually remove the carriage return and line feed characters from the script files. For more information, see Section A.7, "About the DAC Server Shell Scripts."

8. Copy the contents of the /dac/unix_script_bkp directory into the /dac directory.
9. Edit the config.sh file located in the /dac directory to point to the correct version of the JDK by setting the JAVA_HOME environment variable.

4.9.3 Installing JDBC Drivers for DAC Database Connectivity

DAC requires JDBC drivers for database connectivity. The JDBC drivers that are used should be for the databases supported. Since JDBC drivers show variations with different database versions, only drivers that are shipped with the database or downloaded from database vendor site and known to be certified for the given database version should be used. Currently, third-party JDBC drivers for the databases are not supported.

To enable DAC database connectivity, you must install the appropriate JDBC driver in the dac\lib directory on the machines where the DAC Client and Server are installed.

> **Note:** This section applies to Windows, UNIX and Linux. The instructions in this section use the Windows folder path format.

To install JDBC drivers in the \dac\lib directory

- **Oracle**
  - If you are using Oracle database 11g, find the directory where Oracle is installed. Copy the file named ojdbc6.jar in the jdbc\lib directory and paste it in the \dac\lib directory.
  - If you are using an Oracle database 9.x or 10.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 database 8.x, copy the file named classes12.zip and paste it in the 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 \dac\lib directory

  > **Note:** If your source or target database is DB2-UDB, you also need to create DB2-UDB stored procedures to be used during the ETL process. For instructions, see Section 4.19.2, "Creating Stored Procedures for DB2-UDB".

- **MSSQL**

  DAC is configured for Microsoft SQL Server 2005 JDBC drivers. If you are using a Microsoft SQL Server database, then download the Microsoft SQL Server 2005 JDBC Driver file sqljdbc.jar and copy it to the \dac\lib directory.

You can use the Microsoft SQL Server 2000 JDBC Driver files if you edit the \conf\connection_templates.xml file and un-comment the section that starts <!-- THIS SECTION IS FOR SQL SERVER 2000. Comment this section while using SQL Server 2005.-->.

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

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

- **Teradata**

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

4.9.4 Creating ODBC Database Connections for the DAC Client

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

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

For the ODBC connections to Oracle databases, you must use the Oracle Merant ODBC Driver that is installed with the DAC platform installation. For all other databases, you should use ODBC drivers supplied by your database vendor.

> **Note:** On Windows Vista, the Oracle Merant Driver is not successfully installed by the DAC installer. Use Microsoft ODBC Administrator to configure an ODBC connection with the native ODBC driver instead. Use this ODBC when creating tables in the Oracle Business Analytics Warehouse using the DAC Client.

Refer to the appropriate instructions for your database type:

- Section 4.9.4.1, "How to Create ODBC Connections for DB2 Databases"
- Section 4.9.4.2, "How to Create ODBC Connections for Oracle Databases"
- Section 4.9.4.3, "How to Create ODBC Connections for SQL Server Databases"
- Section 4.9.4.4, "How to Create ODBC Connections for Teradata Databases"

4.9.4.1 How to Create ODBC Connections for DB2 Databases

Follow these instructions for creating ODBC connections for DB2 databases on Windows. For instructions on creating ODBC connections for DB2 databases on UNIX or Linux, see the documentation provided with your database.

To create ODBC connections for DB2 databases

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

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

2. If necessary, on Windows, in the System DSN tab of the ODBC Data Source Administrator, create an ODBC connection to the Oracle Business Analytics Warehouse (and, if your source system is Siebel CRM, to the transactional database) using an ODBC driver.

3. Test the connections to make sure they work.

4.9.4.2 How to Create ODBC Connections for Oracle Databases

Follow these instructions for creating ODBC connections for Oracle databases on Windows. For instructions on creating ODBC connections for Oracle databases on UNIX or Linux, see the documentation provided with your database.

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

To create ODBC connections for Oracle databases

1. On the Windows machine that will host the DAC Client, navigate to the ODBC Data Source Administrator.

   Use the System DSN tab of the ODBC Data Source Administrator to create an ODBC connection to the Oracle Business Analytics Warehouse database using the Oracle Merant ODBC driver that is supplied with Oracle BI Applications.

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

2. Click the System DSN tab.

3. Click Add.

4. In the list of drivers, select the Oracle Merant ODBC driver that is installed with DAC, for example, Oracle Merant ODBC Driver in DAC 10g_Oracle - OH1.

5. In the ODBC Oracle Driver Setup dialog box, enter or select the following:

Field	Description
Data Source Name	Enter any meaningful name.
Server Name	Enter the tnsname for the database.
Client Version	Select 10gR1 for 10g and 11g databases.

6. Click Test Connect to make sure the connection works.

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

4.9.4.3 How to Create ODBC Connections for SQL Server Databases

Follow these instructions for creating ODBC connections for SQL Server databases on Windows.

To create ODBC connections for SQL Server databases

1. In Windows, in the System DSN tab of the ODBC Data Source Administrator, create the following:
   - An ODBC connection to the Oracle Business Analytics Warehouse database on the machine that will host the DAC Client.
   - (If your source system is Siebel CRM) An ODBC connection to the transactional database on the machine where you will run the Oracle BI Applications installer.

   Note: Select SQL Server as the ODBC driver.
2. Test the connections to make sure they work.

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

4.9.4.4 How to Create ODBC Connections for Teradata Databases

Follow these instructions for creating ODBC connections for Teradata databases.

To create ODBC connections for Teradata databases on Windows

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

   Field=Enter

   DateFormat=AAA

   SessionMode=ANSI

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

To create ODBC connections for Teradata databases on UNIX

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
3. For UNICODE environments, in the Teradata ODBC Driver Advanced Options dialog box, add the following:
   - CharacterSet=UTF8

4.9.5 Installing pmcmd and pmrep to Enable Communication Between Informatica PowerCenter and DAC

DAC uses the Informatica *pmrep* and *pmcmd* command line programs to communicate with Informatica PowerCenter in the following ways:

- DAC Server uses
  - pmrep to communicate with PowerCenter Repository Services.
  - pmcmd to communicate with PowerCenter Integration Services to run the Informatica workflows.
- DAC Client uses
  - pmrep to synchronize tasks with Informatica workflows and to keep the DAC task source and target tables information up to date.

The *pmrep* program is installed in the PowerCenter Client and PowerCenter Services bin directories (for example, \PowerCenter8.6.1\client\bin and \PowerCenter8.6.1\server\bin). Because of the requirement to co-locate the DAC Client with the PowerCenter Client, the pmrep program is available on the machine for the DAC Client to use.

The *pmcmd* program is installed in the PowerCenter Services bin directory. In order for DAC to access pmcmd, it must also reside in the PowerCenter Client bin folder on the same machine where the DAC Client is installed. Copying the pmcmd executable file from the PowerCenter Services bin folder to the PowerCenter Client bin folder is a manual process.

To install pmcmd in the PowerCenter Client 8.6.1 bin directory

Do one of the following:

- If PowerCenter Services 8.6.1 has been installed on the same machine as the DAC Client and PowerCenter Client 8.6.1, copy the pmcmd.exe and pmrep.exe from the PowerCenter 8.6.1\server\bin directory to PowerCenter 8.6.1\client\bin directory.
- If Informatica PowerCenter Services 8.6.1 is installed on a Windows machine other than the one that hosts the DAC Client, copy the pmcmd.exe file from the \PowerCenter 8.6.1\server\bin directory on the machine where PowerCenter Services is installed to the \PowerCenter 8.6.1\client\bin directory on the DAC Client machine.
- If Informatica PowerCenter Services is installed on a UNIX or Linux machine, and an installation on Windows is not available, run the Informatica PowerCenter Services installer on the DAC Client machine or (any other Windows machine) to install only Integration Services as follows:

  a. Run the Informatica PowerCenter Services installer as described in Section 4.6.2, "Installing Informatica PowerCenter Services."

  b. Copy the pmcmd.exe file located in PowerCenter 8.6.1\server\bin to PowerCenter 8.6.1\client\bin.

4.10 Installing DAC Metadata Files

DAC metadata files are installed by the Oracle BI Applications installer (as described in Section 4.5, "Running the Oracle BI Applications Installer on Windows"). You need to copy these files to the machines hosting the DAC Client and Server.

Table 4–3 describes the location of the file or folder names in the Oracle BI Applications installation that you need to copy to the machine hosting the DAC Client.

Table 4–4 describes the location of the file names in the Oracle BI Applications installation that you need to copy to the machine hosting the DAC Server.

Table 4–3 DAC Metadata Files That Need to Be Copied to DAC Client Machine

File/Folder Name	Location of File/Folder in BI Applications Installation	File/Folder to Be Copied to Following Location
Export folder	OraclelBI\dwrep\DAC_metadata\DAC_Client\export	Copy contents of the Export folder to the dac\export folder on machine hosting the DAC Client. For example: C:\orahome\10g<version>\bifoundation\dac\export
deletetriggers.list	OraclelBI\dwrep\DAC_metadata\DAC_Client	Copy this file to the dac\conf folder on the DAC Client machine. For example: C:\orahome\10g<version>\bifoundation\dac\conf.

Table 4–4 DAC Metadata Files That Need to Be Copied to DAC Server Machine

File/Folder Name	Location of File/Folder in BI Applications Installation	File to Be Copied to Following Location
parameterfileOLTP.txt	OraclelBI\dwrep\DAC_metadata\DAC_Server	Copy this file to the dac\Informatica\parameters\input folder on the DAC Server machine. For example, C:\orahome\10g<version>\bifoundation\dac\Informatica\parameters\input.
parameterfileDW.txt	OraclelBI\dwrep\DAC_metadata\DAC_Server	Copy this file to the dac\Informatica\parameters\input folder on the DAC Server machine. For example: C:\orahome\10g<version>\bifoundation\dac\Informatica\parameters\input.

4.11 Logging into DAC for the First Time and Importing Metadata into the DAC Repository

When you log into DAC for the first time, you create a set of stored login information. Also, as part of the first-time login process, DAC prompts you to create the DAC Repository schema. Once the schema is created, you can then import metadata into the DAC Repository.

This section contains the following topics:

- Section 4.11.1, "Logging into DAC and Creating a Connection to the DAC Repository"
- Section 4.11.2, "Importing Metadata into the DAC Repository"

4.11.1 Logging into DAC and Creating a Connection to the DAC Repository

When you log into DAC for the first time, you must first configure a connection to connect to the DAC Repository. DAC stores this connection information for subsequent logins.

After configuring a connection to the DAC Repository and logging in, the DAC will automatically prompt you to upgrade the repository schema.

DAC Repository Database Authentication File

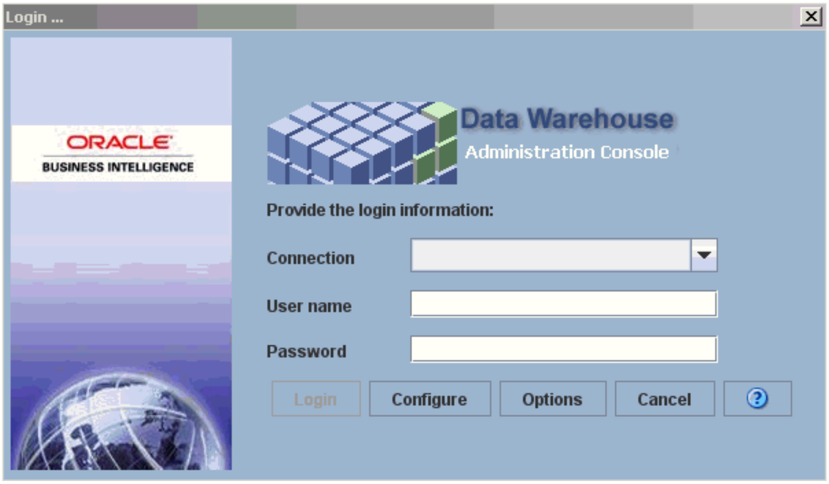
When you configure a connection to the DAC Repository, the configuration process includes creating a new authentication file or selecting an existing authentication file. The authentication file authenticates the database in which the repository resides. If you create a new authentication file, you will specify the table owner and password for the database.

A user with the Administrator role must distribute the authentication file to any user account that needs to access the specified DAC Repository. For information about managing user accounts, see Section A.2, "About DAC User Account Management.".

To log into DAC for the first time and upgrade the repository schema

1. Start the DAC Client by navigating to the $ORACLE_HOME\bifoundation\dac directory and double-clicking the startclient.bat file.

   The Login ... dialog box appears.



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

Field	Required Value
Name	Enter a unique name for the connection to the DAC Repository.
Connection type	Select the type of database in which the DAC Repository will be stored.

Field	Required Value
Connection String, or Database name, or TNS Name, or Instance	Select the database name or database account name of the DAC Repository. If you are using: ■ Oracle (OCI8), use the tnsnames entry. ■ Oracle (Thin), use the instance name. ■ SQL Server, use the database name. ■ DB2-UDB, use the connect string as defined in the DB2 configuration.
Database Host	Enter the name of the machine where the DAC Repository will reside.
Database Port	Enter 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.
Optional URL	Can be used to override the standard URL for this connection.
Optional Driver	Can be used to override the standard driver for this connection.
Authentication File	Click in this field to do one of the following: ■ Select an existing authentication file. ■ Create a new authentication file. Proceed to the next step for detailed instructions.

5. To select an existing authentication file, do the following:
   a. Click in the Authentication File field of the Configuring... dialog box.
   b. In the Authentication File dialog box, select Choose existing authentication file.
   c. Navigate to the appropriate folder, and select the authentication file. Click OK.
   d. In the Configuring... dialog box, click Test Connection to confirm the connection works.
   e. Click Apply, and then click Finish.

   Note: You must distribute this authentication file to all user accounts that need to access this DAC Repository.

6. To create a new authentication file, do the following:
   a. Click in the Authentication File field of the Configuring... dialog box.
   b. In the Authentication File dialog box, select Create authentication file.
   c. Navigate to the folder where you want to save the new authentication file, and click OK.
   d. In the Create Authentication File dialog box, enter a unique name for the authentication file, and click OK.
   e. Enter the Table Owner Name and Password for the database where the repository will reside.

   f. In the Configuring... dialog box, click Test Connection to confirm the connection works.

   g. Click Apply, and then click Finish.

> **Note:** You must distribute this authentication file to all user accounts that need to access this DAC Repository.

7. In the Login... dialog box, do the following:

   a. Select the appropriate Connection from the drop-down list.

   b. Enter Administrator as the User Name.

   c. Enter Administrator as the Password.

   d. Click Login.

8. When prompted to create a repository, click Yes.

   This process creates DAC repository tables.

   Depending on your database type, you may have the option specify a tablespace.

   The Unicode check box is available for a repository on SQL Server or DB2 databases. Check the Unicode check box if your deployment requires a Unicode schema to be created.

4.11.2 Importing Metadata into the DAC Repository

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

> **Note:** The DAC Client can connect to only one DAC Repository at a time.

To import metadata into the DAC Repository

1. In DAC, select Tools, then DAC Repository Management, then Import to display the Import dialog box.

   Make sure that the correct \DAC\export\ directory is displayed at the top of the dialog box (for example, <drive>:\orahome\10g<version>\bifoundation\dac\export). If necessary, use the 'Change import/export directory' button to select the \dac\export\ directory. If the wrong directory is selected, the Applications list will be empty

2. In the Categories area, select the Logical check box and the System check box.

   The information imported by these options is 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.
System	Imports all information contained in the DAC Setup view, except passwords for servers and database connections.

3. In the Applications List table, use the check boxes in the Selected column to specify the source system applications for which you will import the ETL metadata, as follows:

   a. Select the check box next to Universal.

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

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

      Note: Do not select the Data Warehouse option unless Oracle specifically instructs you to do so. This container is reserved for special purposes.

      Tip: Make a note of the Application names that you select. When you create data warehouse tables later in the configuration process, you might need to type in the names exactly as they are displayed here (for more information, see Section 4.12.1, "Creating Data Warehouse Tables").

4. Select the Truncate Repository Tables check box.

   Note: If you are importing DAC metadata for the first time, you should select the Truncate Repository Tables check box.

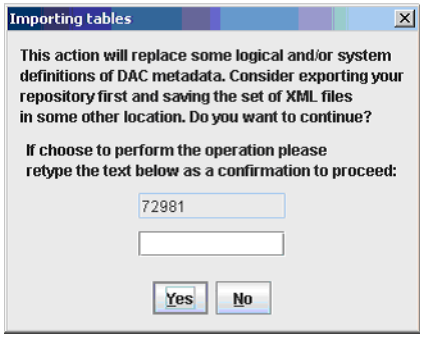
   If you are importing DAC metadata into a repository that already has metadata, do the following:

   1. Back up the current repository by exporting it to an empty folder (use the 'Change import/export directory' button to select an empty folder.
   2. Use the 'Change import/export directory' button to select the \dac\export\ directory.
   3. Select the Truncate Repository Tables check box and the Enable Batch Mode check box, and import the new DAC metadata.
   4. Use the 'Change import/export directory' button to select the directory where you backed up data in Step 1.
   5. Deselect the Truncate Repository Tables check box and re-import the data you backed up.

5. Select the Enable Batch Mode check box.

   Note: If you are using an Oracle 9i Release 2 database, you cannot use batch mode because the Oracle 9i JDBC drivers do not handle array inserts or bulk inserts.

6. Click OK to display the Importing tables dialog box.



This action will replace some logical and/or system definitions of DAC metadata. Consider exporting your repository first and saving the set of XML files in some other location. Do you want to continue?

If choose to perform the operation please retype the text below as a confirmation to proceed:

72981

7. To confirm that you want to import the seed data selected, re-type the text in the text box and click Yes.
8. If prompted to verify that you want to continue with the Import, click Yes.

   When the process is complete, the DAC displays a status box containing a success or failure message. If the process fails, use the \DAC\log\import.log file to diagnose errors.

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

9. Click OK.
10. If your source or target database is a DB2/390 database, run DAC_DB2390_Source.sql or DAC_DB2390_Target.sql immediately after importing the seed data by connecting to the database where the DAC Repository resides. These files are stored in the \DAC directory.

4.12 Creating the Oracle Business Analytics Warehouse Tables

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

Note: Before you start this procedure, you need to create a database for the Oracle Business Analytics Warehouse. For more information, see Section 4.4.1, "Create Databases for Oracle BI Applications and Informatica PowerCenter Components."

The Oracle Business Analytics Warehouse tables are created by the DAC Client. The DAC Client uses ODBC connections to the Oracle Business Analytics Warehouse database for this procedure. Ensure that you have created an ODBC connection to the Oracle Business Analytics Warehouse database as described in Section 4.9.4, "Creating ODBC Database Connections for the DAC Client."

If your Oracle Business Analytics Warehouse database is Oracle, you must use the Oracle Merant ODBC driver to create the ODBC connection. This driver is installed by the Oracle BI Applications. The DAC Client must be installed by running the Oracle BI Applications installer so that the driver is installed on the machine. Create the ODBC DSN to the Oracle Business Analytics Warehouse as described in Section 4.9.4, "Creating ODBC Database Connections for the DAC Client."

Before you perform the procedures described in this section, make sure that the SSE role has been created for the Oracle Business Analytics Warehouse, and that the database user has been associated with the role (for more information, see Section 4.4.1.1, "How to Create the SSE Role").

If your Oracle Business Analytics Warehouse database is not Teradata, refer to Section 4.12.1, "Creating Data Warehouse Tables" to create the data warehouse tables.

If you are using a Teradata database as the Oracle Business Analytics database, the DAC creates a SQL file to create the schema tables, not the tables themselves. Follow the steps in Section 4.12.2, "Creating Data Warehouse Tables on a Teradata Database".

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

4.12.1 Creating Data Warehouse Tables

Use this procedure to create the data warehouse tables.

If the Oracle Business Analytics Warehouse database is Teradata, follow the steps in Section 4.12.2, "Creating Data Warehouse Tables on a Teradata Database."

To create data warehouse tables

1. In DAC, select Tools, then ETL Management, then Configure.

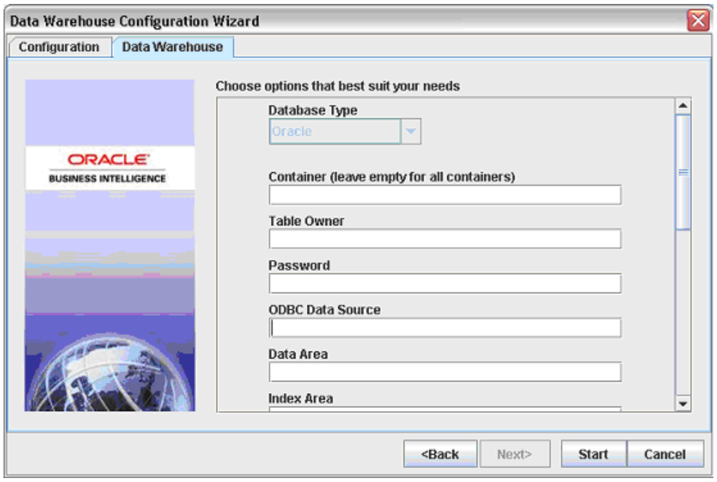
   For more information about logging into DAC, see Section A.1, "How to Log into DAC."

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.



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

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 box (see step 2). If the database type is DB2/390, then check the 390 Database check box.
Container	The name of the source business applications for which you want to create the data warehouse tables. **Note**: You must leave this field blank, unless told to specify a value by Oracle Support. If you leave the Container field blank, DAC creates a container by default for all of the following: ■ The source business applications that you selected when you imported the seed data into the DAC metadata repository earlier (for more information, see Section 4.11.2, "Importing Metadata into the DAC Repository"). ■ Any copies of those source system applications. ■ Any additions made to those source system applications. ■ Any new source system applications that you have custom built. If there are tables that are common to these containers, then only one table will be created. For example, if there is a table called W_ORG_D in Oracle 11.5.9 and Oracle 11.5.10, then DAC will create only one table called W_ORG_D. If columns are different for the same table across containers, then DAC will create a table that has all the columns in the same table. If you only want to deploy a subset of the source business applications for which you imported seed data earlier, then use this field to specify a container name. When you specify a container name, you must enter the names of the applications exactly as they are displayed on the seed data Import dialog box. For example: ■ If you want to deploy Oracle 11.5.9, enter: 'Oracle 11.5.9'. ■ If you want to deploy Oracle 11.5.9 and 11.5.10, enter: 'Oracle 11.5.9,Oracle 11.5.10'. For more information about containers, see *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.
Table Owner	Valid database owner, username, or account that you set up to hold the data warehouse (for more information, see Section 4.4.1, "Create Databases for Oracle BI Applications and Informatica PowerCenter Components").
Password	Valid database user password for the database owner, username, or account that you specified in the Table Owner field (for more information, see Section 4.4.1, "Create Databases for Oracle BI Applications and Informatica PowerCenter Components").
ODBC Data Source	Data Source Name (DSN) for the Oracle Business Analytics Warehouse. You must specify the name of the ODBC connection (for example, Connect_to_OLAP) that you created for the data warehouse. For more information, see Section 4.9.4, "Creating ODBC Database Connections for the DAC Client".
Data Area	(Optional) Tablespace where data warehouse tables are created.
Index Area	(Optional) Indexspace where data warehouse indexes are created (applicable only to Oracle and DB2 databases).
Is Unicode	Specifies whether the data warehouse database is Unicode. The database must be set to support the creation of a Unicode schema. For more information, see Section 4.3.3, "Code Page and Data Movement Requirements."

6. Click Start.

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

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

4.12.2 Creating Data Warehouse Tables on a Teradata Database

This section explains how to create and delete data warehouse tables on a Teradata database. Before you start, make sure that the tables that you create are case specific by setting the session mode to ANSI in your Teradata ODBC configuration (for more information, see Section 3.5, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse").

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

To create data warehouse tables on a Teradata database

1. In DAC, select Tools, then ETL Management, then Configure.

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

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

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

   - If a 'Success' message is displayed, the data warehouse tables have been created. If you want to see log information about the process, use the following log files.

     \OracleBI\DAC\config\generate_ctl.log - A log of the schema definition process, including details of any conflicts between containers.

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

   - If a 'Failure' message is displayed, the data warehouse tables have not been created. Use the log information in \OracleBI\DAC\config\generate_ctl.log to diagnose the error. The createtables.log is not generated.

7. Copy the SQL file created in step 6 from \conf\sqlgen\sql\Teradata into SQL Assistant and execute the SQL.

4.13 Configuring the DAC Server

This section contains instruction for configuring the DAC Server. It contains the following topics:

- Section 4.13.1, "Configuring the Connection Between the DAC Server and DAC Repository"
- Section 4.13.2, "Setting Environment Variables to Enable Communication between the DAC Server on UNIX and Informatica"
- Section 4.13.3, "Starting the DAC Server"
- Section 4.13.4, "Activating Join Indexes for Teradata Databases"

4.13.1 Configuring the Connection Between the DAC Server and DAC Repository

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

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

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

- To configure the DAC Server repository connection using the DAC Client, see Section 4.13.1.1, "How to Configure the DAC Server Repository Connection Using the DAC Client (Windows)".
- To configure the DAC Server repository connection using the serverSetupPrompt scripts, see Section 4.13.1.2, "How to Configure the DAC Server Repository Connection Using serverSetupPrompt Scripts (Windows, UNIX or Linux)".

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

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

To configure the connection between the DAC Server and the DAC Repository using the DAC Client

1. In DAC, select Tools, then DAC Server Management, then DAC Server Setup.

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

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

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

2. Click Yes to display the Server Configuration dialog box.

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

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

Field	Description
Connection type	Select the type of database that you are using to store the DAC metadata repository. Depending on what type you select, the connection details below change (see *Connection fields* below).
Connection fields (for example, Instance, TNS Name, Connection string/Database name.	Specify connection details for the database that stores the DAC metadata repository. ■ If you select Oracle (Thin), you are prompted below for the following information: ■ Instance (for example, mymachinename). ■ Database Host (fully qualified, for example, mymachine.us.company.com). ■ Database Port (for example, 1521). ■ Table owner name, using the account that you created a database to store the DAC Repository (for example, DAC). For more information, see Section 4.4.1, "Create Databases for Oracle BI Applications and Informatica PowerCenter Components". ■ Password (that is, for the above database account). ■ If you select Oracle (OCI8), you are prompted below for a TNS name (for example, mymachinename@host.com). ■ If you select DB2, you are prompted below for a Connection string. ■ If you select MS SQL Server, you are prompted below for a Database name (for example, mydacdatabase).
Table owner name	The database table name or instance (for example, DAC) that you created to store the DAC Repository (for more information, see Section 4.4.1, "Create Databases for Oracle BI Applications and Informatica PowerCenter Components").
Password	The database or instance password (for example, DAC).

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

- When you created a database to store the DAC Repository (for more information, see Section 4.4.1, "Create Databases for Oracle BI Applications and Informatica PowerCenter Components").

- When you created a DAC connection (for more information, see Section 4.11.1, "Logging into DAC and Creating a Connection to the DAC Repository").

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

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

Use the serverSetupPrompt.sh to configure the connection between the DAC Server and the DAC Repository when the DAC Server is installed on UNIX or Linux, as described below. For more information on the DAC Server scripts, see Section A.7, "About the DAC Server Shell Scripts."

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

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

1. Run the serverSetupPrompt script, as follows:
   - On Windows, double-click the serverSetupPrompt.bat located in the \dac directory.
   - On UNIX or Linux, run serverSetupPrompt.sh located in the /dac directory.
2. Enter 1 in the 'Please make your selection' prompt to enter repository connection information.
3. Enter the number for the type of database storing the DAC Repository from the list of connection type choices.
4. Enter the connection information as described in the Connection Information table in section Section 4.13.1.1, "How to Configure the DAC Server Repository Connection Using the DAC Client (Windows)".
5. Enter 2 to test the DAC Repository connection.
6. Enter 5 to save changes.
7. Enter 6 to exit.

4.13.2 Setting Environment Variables to Enable Communication between the DAC Server on UNIX and Informatica

When you install the DAC Server on UNIX, you must define the path for the Informatica Domain file and set environment variables manually. For instructions, see Section A.6.2, "How to Set Environment Variables for DAC Server Communication on UNIX."

4.13.3 Starting the DAC Server

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

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

For more information on starting and stopping the DAC Server on Windows, UNIX or Linux, see Section A.4, "How to Start and Stop the DAC Server."

4.13.4 Activating Join Indexes for Teradata Databases

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

To activate join indexes for Teradata databases

1. In DAC, create a new Execution Plan with the list of Subject Areas that you want to extract.
2. Query for all Tasks whose name starts with 'Teradata Drop' and add them as preceding tasks.
3. Query for all Tasks whose name start with 'Teradata Create' and add them as following tasks.
4. Assemble the Execution Plan parameters in the Parameters tab and configure the parameters.
5. Redesign the Execution Plans.

4.14 Configuring DAC Integration Settings

This section explains how to configure DAC integration settings. It contains the following topics:

- Section 4.14.1, "Setting DAC System Properties"
- Section 4.14.2, "Registering Informatica Services in DAC"
- Section 4.14.3, "Setting Physical Data Sources"

You must start the DAC Server to complete certain steps in the procedures described below. For information on how to start the DAC Server, see Section A.4, "How to Start and Stop the DAC Server."

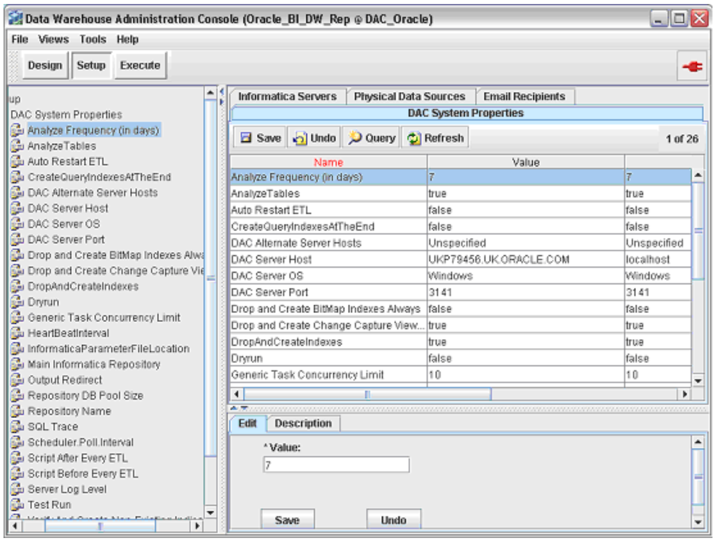
4.14.1 Setting DAC System Properties

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

To set DAC System Properties

1. Log in to DAC.

   For more information about logging into DAC, see Section A.1, "How to Log into DAC."
2. From the Views menu, select Setup, then DAC System Properties.



Name	Value	
Analyze Frequency (in days)	7	7
AnalyzeTables	true	true
Auto Restart ETL	false	false
CreateQueryIndexesAtTheEnd	false	false
DAC Alternate Server Hosts	Unspecified	Unspecified
DAC Server Host	UKP79456.UK.ORACLE.COM	localhost
DAC Server OS	Windows	Windows
DAC Server Port	3141	3141
Drop and Create BitMap Indexes Always	false	false
Drop and Create Change Capture View...	true	true
DropAndCreateIndexes	true	true
Dryrun	false	false
Generic Task Concurrency Limit	10	10

*Value: 7

3. Set values for the following properties:

Property	Value Required
DAC Server Host	Enter the name or IP address of the machine that hosts the DAC Server.
DAC Server OS	Enter the operating system of the machine that hosts the DAC Server. Possible values are Windows, AIX, Solaris, HP-UX, Linux (case sensitive).
InformaticaParameterFile Location	Enter the path to the Informatica source file directory, for example \Informatica\PowerCenter8.6.1\ server\infa_shared\SrcFiles. You can confirm the source file directory for your environment by launching the PowerCenter Administration Console and going to the General Properties area of the Processes tab. The parameter name for the source file directory is $PMSourceFileDir. The value of the $PMSourceFileDir parameter must match the value for the InformaticaParameterFileLocation property.

Note: For a description of all DAC System Properties, see Section A.3, "How to Set DAC System Properties."

4.14.2 Registering Informatica Services in DAC

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

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

- You must register one or more Integration Services service.
- You must register one Repository Service.

> **Tip:** For information about Informatica services properties that you may need to complete this procedure, log into the PowerCenter Administration Console and select the appropriate service. For more information, see Section A.8, "How to Log Into Informatica PowerCenter Administration Console."

To register Informatica Services in DAC

1. In DAC, navigate to the Setup view, and then click the Informatica Servers tab.

   For instructions on logging into DAC, see Section A.1, "How to Log into DAC."

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

   a. Modify the record with Name = INFORMATICA_DW_SERVER by entering the following information in the Edit subtab:

Field	Enter or select
Name	For Informatica version 7.x, logical name for the Informatica Server. For Informatica version 8.x, logical name for the Integration Service.
Type	Informatica.
Service	For Informatica version 7.x, Informatica server host name or IP address. For Informatica version 8.x, name of the Integration Service you created in Section 4.6.4, "Creating the Informatica Integration Service."
Server Port	For Informatica version 7.x only. The port that DAC uses to communicate with the Informatica Server. The default port number is 4006.
Domain	For Informatica version 8.x only. Informatica domain name.
Login	Informatica Repository user name who has appropriate privileges to execute workflows (for example, Administrator). **Note:** DAC must log in to Informatica as an Informatica Repository Administrator user that is configured in the native security domain. For instructions on how to create such a user, see Section 4.8.4, "Creating the Repository Administrator User in the Native Security Domain." For more information about Informatica security domains, see "Chapter4: Managing Users and Groups," in the *PowerCenter Administrator Guide*, which is included on the Informatica PowerCenter DVD provided with Oracle BI Applications.
Password	Informatica Repository user password.
Maximum Sessions	Maximum number of workflows that can be executed in parallel on the Informatica PowerCenter Integration Services service. If the number of sessions is zero or is not specified, the DAC Server assigns the default value of 10.
Repository Name	Name of the Informatica Repository (for example, Oracle_BI_DW_Base.rep).
Inactive	Indicates whether the PowerCenter Integration Services service will participate in the ETL process.

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

      Note: Integration Services must be running.

   c. Click Save to save the details.

3. For the Informatica Repository Service you want to register, do the following:

   a. Modify the record with Name = INFORMATICA_REP_SERVER by entering the following information in the Edit subtab:

Field	Enter
Name	For Informatica version 7.x, logical name for the Repository Server. For Informatica version 8.x, logical name for the Repository Service.
Type	Repository.
Hostname	For Informatica version 7.x, Repository Service host name or IP address. For Informatica version 8.x, Repository Serer host name or IP address.
Server Port	For Informatica version 7.x only. Port where the Repository Service connects for requests. The default port number is 6001.
Login	Informatica Repository user name that has appropriate privileges to execute workflows (for example, Administrator).
Password	Informatica Repository user password (for example, Administrator).
Maximum Sessions	Maximum number of workflows that can be executed in parallel on the Informatica PowerCenter Integration Services service. If the number of sessions is zero or is not specified, the DAC Server assigns the default value of 10.
Repository Name	Name of the Informatica Repository (for example, Oracle_BI_DW_Base.rep).
Inactive	Indicates whether the Repository Service will participate in the ETL process.

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

   Note: The Repository Service must be running.

c. Click Save to save the details.

4.14.3 Setting Physical Data Sources

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

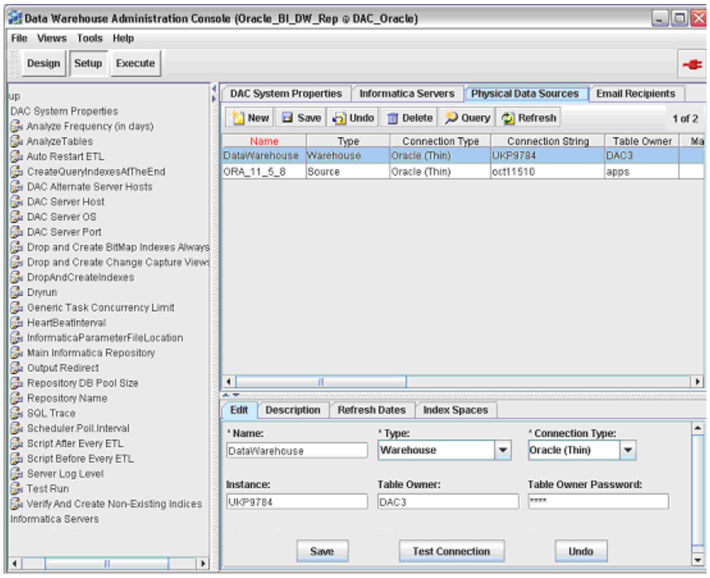
> **Note:** If you have a JD Edwards EnterpriseOne or a JD Edwards World source system hosted on an iSeries DB2 database, see Section 4.14.3.1, "Specifying a Data Source Connection for JD Edwards EnterpriseOne or JD Edwards World Hosted on a DB2 Database."

To specify transactional and data warehouse data sources

1. Log into DAC.

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

2. From the Views menu, select Setup, then Physical Data Sources.



Name	Type	Connection Type	Connection String	Table Owner	Ma
DataWarehouse	Warehouse	Oracle (Thin)	UKP9784	DAC3	
ORA_11_5_8	Source	Oracle (Thin)	oct11510	apps	

* Name: DataWarehouse
* Type: Warehouse
* Connection Type: Oracle (Thin)
Instance: UKP9784
Table Owner: DAC3
Table Owner Password: ****

The Physical Data Sources tab displays a precreated record for the data warehouse with the name DataWarehouse, and one or more records for the OLTP sources. The records that are created by DAC for the OLTP sources depend on the business application source systems you selected when importing the DAC metadata. For more information, see Section 4.11.2, "Importing Metadata into the DAC Repository."

3. For each record, enter the following information in the Edit subtab:

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

Field	Description
Instance or TNS Name or Connection String or Database Name (NOTE: The field name changes based on the Connection Type selection.)	Enter the value appropriate for your database. For an Oracle database instance, enter the database SID (that is the SERVICE_NAME = value in the tnsnames.ora file in \network\admin\. For an Oracle TNS Name, enter the TNS entry name that is specified in the tnsnames.ora file in \network\admin\.
Table Owner	Valid database user.
Table Owner Password	Valid database user password.
Max Num Connections	Maximum number of database connections this connection pool can contain.
DB Host	Machine name or instance where the database resides. For example, databasename.us.company.com.
Port	Port number where the database listens (for example 1521 is the default for an Oracle database).
Dependency Priority	Number used to generate dependencies when designing execution plans.
Data Source Number	Unique number assigned to the data source category so that the data can be identified in the data warehouse. For example, the value '1' is used for Siebel data sources. If you are editing a data source template for a data source type, Oracle recommends that you do not change the default value. If you are specifying a data source without using a pre-defined template, you must use the correct value for that data source category. For example, if you specify an Oracle EBS R12 data source, you must specify the DATASOURCE_NUM_ID value '9'. For a complete list of supported data sources and DATASOURCE_NUM_ID values, see the section entitled, "How to Configure Data Source Num IDs," in *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*. This value is passed as a parameter to the Informatica workflows. If you are using multiple sources, each data source has a unique number. Typically, all source dependent extracts will use this parameter to populate the DATASOURCE_NUM_ID column, and the source independent workflows will carry these values to the final dimension and fact tables.
Default Index Space	(Oracle specific) Specifies the table space in which DAC drops and creates indexes against this database connection To use the default table space, leave this field blank.

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

4.14.3.1 Specifying a Data Source Connection for JD Edwards EnterpriseOne or JD Edwards World Hosted on a DB2 Database

If you have a JD Edwards EnterpriseOne or JD Edwards World source system hosted on an iSeries DB2 database, follow the steps in this section to specify the OLTP connection in DAC.

1. From the IBM site, select and download the appropriate driver based on the version of your iSeries DB2 database.

2. Include the driver in the class path, for example, for IBM DB2 on iSeries V5R4M0, download jt400.jar copied from the iSeries server only.
3. On a machine that hosts the DAC Server and an appropriate drive client access for iSeries, create a data source name (DSN) to connect to the source DB2 database (using an ODBC Connection).
4. Log into DAC.
5. From the Views menu, select Setup, then Physical Data Sources.
6. Select the record for the JD Edwards EnterpriseOne or JD Edwards World source.
7. In the Edit tab, complete the following fields:

Field	Description
Name	This is the logical name for the database connection. It must match the name in the OLTP record that you selected. Do not change the default value.
Type	Enter Source.
Connection Type	Enter DB2.
Instance	Or connection string. Enter the DSN that you created in step 3. You enter the DSN, or host name, because the JDBC driver and URL that you enter below are used to override the connection to the database.
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.
JDBC Driver	The name of the driver. The driver that you enter will override the driver that connects to the iSeries database. The driver that you select must be applicable to the version of iSeries on which JD Edwards EnterpriseOne or JD Edwards World is hosted. For example, for IBM DB2 on iSeries V5R4M0, you would enter this driver: `com.ibm.as400.access.AS400JDBCDriver`
URL	The URL for the driver. The URL that you enter will override the driver URL that connects to the iSeries database. The driver URL that you select must be applicable to the version of iSeries on which JDE source is hosted. For example, for IBM DB2 on iSeries V5R4M0, you would enter this URL: `jdbc:as400://<hostname>/LIBRARY`

8. Click Test Connection to make sure the connection works.
9. Click Save.

4.15 Configuring Relational and Application Connections in Informatica Workflow Manager

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

- Section 4.15.1, "Logging Into Informatica PowerCenter Workflow Manager"
- Section 4.15.2, "Configuring Connections in Informatica PowerCenter Workflow Manager"

4.15.1 Logging Into Informatica PowerCenter Workflow Manager

Note: Before you log into Workflow Manager, start the Informatica services.

To log into Informatica Workflow Manager

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

4.15.2 Configuring Connections in Informatica PowerCenter Workflow Manager

You use Informatica PowerCenter Workflow Manager to configure the relational and application connections that are required for your deployment, as follows:

- For all data sources except PeopleSoft OLTP data sources, configure relational database connections for both OLTP and OLAP data sources by following the steps in Section 4.15.2.1, "How to Configure Relational Connections."
- For PeopleSoft OLTP data sources, configure application connections by following the steps in Section 4.15.2.2, "How to Configure Application Connections for PeopleSoft OLTP Data Sources."
- For Teradata databases, follow the additional steps in Section 4.15.2.3, "(Teradata specific) How to Configure the Informatica Repository for Teradata External Loader Connections."

Note: The Informatica services must be running to perform these tasks.

4.15.2.1 How to Configure Relational Connections

Follow this procedure to configure relational connections:

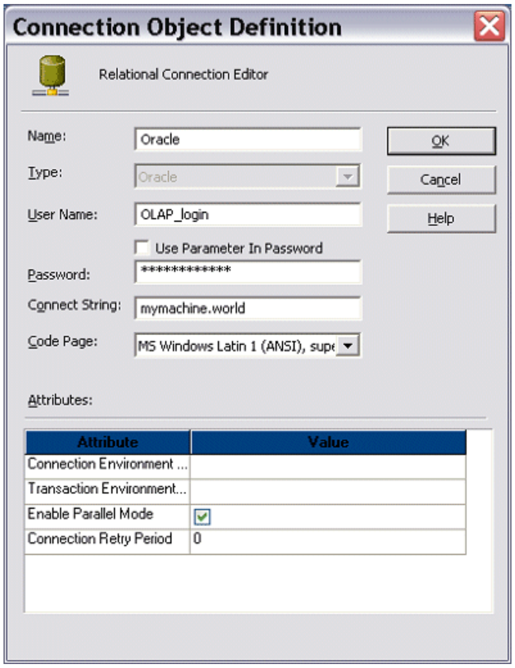
1. In Informatica PowerCenter Workflow Manager, select Connections, then Relational to display the Relational Connection Browser.

   You need to create a connection for each transactional (OLTP) database, and a connection for the Oracle Business Analytics Warehouse (OLAP) database.

2. For each database connection you need to create, do the following:

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

   b. Use the Connection Object Definition dialog box to define the relational connection.



Connection Object Definition

Relational Connection Editor

Name: Oracle

Type: Oracle

User Name: OLAP_login

□ Use Parameter In Password

Password: ************

Connect String: mymachine.world

Code Page: MS Windows Latin 1 (ANSI), supe

Attributes:

Attribute	Value
Connection Environment ...	
Transaction Environment...	
Enable Parallel Mode	☑
Connection Retry Period	0

   c. Click OK to save the details.

   Notes:

   - If the target database is Oracle or DB2, use the following settings:

     – Click New, select the appropriate database type, and then click OK.

     – Name: DataWarehouse, for connection to the Oracle Business Analytics Warehouse.

       You must specify 'DataWarehouse' exactly as it appears in the Physical Data Sources tab in the DAC Setup View. For the connection to the OLTP, specify the name exactly as it appears in the Physical Data Sources tab in the DAC Setup view. For example, if your source system is Siebel release 7.8.x, then you name this connection as SEBL_78. For more information, see Section 4.14.3, "Setting Physical Data Sources".

  - User Name: Database user name with the appropriate read and write database permissions to access the database.
  - Password: Password for the user name.
  - Connect string: Connect string used to communicate with the database (refer to the Informatica Help for information about specifying this value).
  - Code Page: Code page compatible with the code page of the database client. If NLS_LANG (for Oracle database) or DB2CODPAGE (for DB2 database) has been set, then the Code Page value should be a code page compatible with the language set by these variables. For more information, see Section 4.3.3, "Code Page and Data Movement Requirements".

- If the target database is SQL Server, use the following settings:
  - Click New and select the type as SQL Server, then click OK.
  - Name: DataWarehouse, for the connection to the Oracle Business Analytics Warehouse. You must specify 'DataWarehouse' exactly as it appears in the Physical Data Sources tab in the DAC Setup View.
  - For the connection to the OLTP, specify the name exactly as it appears in the Physical Data Sources tab in the DAC Setup View. (For example, if your source system is Siebel version 7.8.x then you name this connection as SEBL_78). For more information, see Section 4.14.3, "Setting Physical Data Sources".
  - User Name: Database user name with the appropriate read and write database permissions to access the database.
  - Password: Password for the user name.
  - Code Page: Code page compatible with the code page of the database client.
  - Database name: Name of the database.
  - Server name: Database server name.
  - Domain name: Name of the domain.
- If the target database is Teradata, use the following settings:
  - Click New and select the type as Teradata, then click OK.
  - Name: DataWarehouse, for the connection to the Oracle Business Analytics Warehouse. You must specify 'DataWarehouse' exactly as it appears in the Physical Data Sources tab in the DAC Setup View.
  - For the connection to the OLTP, specify the name exactly as it appears in the Physical Data Sources tab in the DAC Setup View. (For example, if your source system is Siebel version 7.8.x then you name this connection as SEBL_78). For more information, see Section 4.14.3, "Setting Physical Data Sources".
  - User Name: Database user name with the appropriate read and write database permissions to access the database.
  - Use Parameter in Password: Indicates the password for the database user name is a session parameter, $ParamName. Define the password in the workflow or session parameter file, and encrypt it using the pmpasswd CRYPT_DATA option.

- Password: Password for the database user name. For Teradata connections, this overrides the database password in the ODBC entry. Passwords must be in 7-bit ASCII.
- Code Page: Code page compatible with the code page of the database client.
- Database name: Name of the database. For Teradata connections, this overrides the default database name in the ODBC entry. If you do not enter a database name for a Teradata connection, the Integration Service uses the default database name in the ODBC entry.
- Data Source name: Name of the Teradata ODBC data source.
- Domain name: Name of the domain.

4.15.2.2 How to Configure Application Connections for PeopleSoft OLTP Data Sources

PeopleSoft OLTP data sources use Application Connections that implement Informatica's PowerConnect for PeopleSoft. PeopleSoft OLTP data sources do not use relational connections. Therefore, you need to use Informatica Workflow Manager to define Application Connections for the PeopleSoft OLTP data sources as described below.

Note: You need to configure relational connections for OLAP databases with PeopleSoft adapters (for more information, see Section 4.15.2.1, "How to Configure Relational Connections.").

To configure Application Connections for PeopleSoft OLTP data sources:

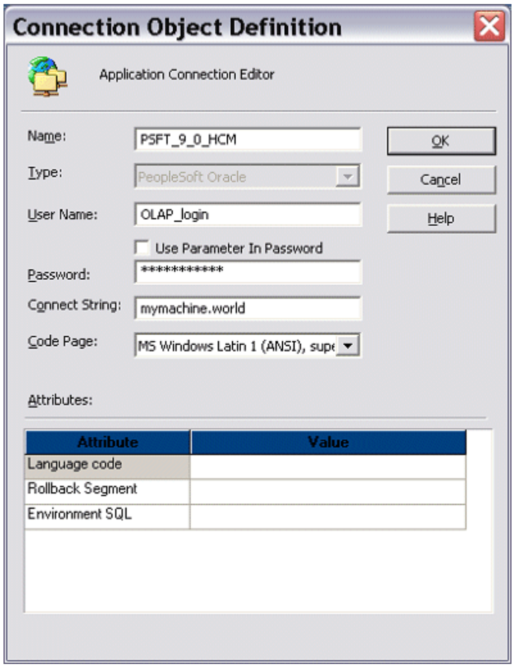
1. In Informatica PowerCenter Workflow Manager, select Connections, then Application, to display the Application Connection Browser dialog.

   You need to create an Application Connection for each PeopleSoft transactional (OLTP) data source.

2. For each Application Connection that you need to create, do the following.

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

   b. Use the Connection Object Definition dialog to define an Application Connection (for example, a connection named PSFT_9_0_HCM).



Name: PSFT_9_0_HCM

Type: PeopleSoft Oracle

User Name: OLAP_login

☐ Use Parameter In Password

Password: ************

Connect String: mymachine.world

Code Page: MS Windows Latin 1 (ANSI), supe

Attributes:

Attribute	Value
Language code	
Rollback Segment	
Environment SQL	

c. Click OK to save the details.

Notes:

- You must specify the **Name** value exactly as it appears in the Physical Data Sources tab in the DAC Setup View. For example, if your source system is named PSFT_9_0_HCM in DAC, then you must name this connection as PSFT_9_0_HCM. For more information, see Section 4.14.3, "Setting Physical Data Sources".
- Connect string: Connect string for the database (refer to the Informatica Help for information about specifying this value).

4.15.2.3 (Teradata specific) How to Configure the Informatica Repository for Teradata External Loader Connections

To configure the Informatica Repository for Teradata, you need to do the following:

- Specify Loader Connections for Teradata (for more information, see Section 4.15.2.3.1, "How to Specify Loader Connections for Teradata").
- For each workflow, specify Loader Connection details at the session level (for more information, see Section 4.15.2.3.2, "How to Specify Teradata Details at the Workflow Level").

4.15.2.3.1 How to Specify Loader Connections for Teradata Follow this procedure to configure the Informatica Repository in Informatica PowerCenter Workflow Manager for Teradata external loader connections.

1. In Informatica PowerCenter Workflow Manager, select Connections, then Loader to display the Loader Connection Browser.
2. In the Objects list, select Teradata_Tpump_Upsert, and then click Edit to display the Connection Object Definition dialog box.
3. Edit the User Name, Password, TDPID, Database Name, Error Database, and Log Table Database, and other attributes as necessary. Please note that if you do not enter the Error Database and Log Table Database, Informatica will default it to the same as the Tables Database.

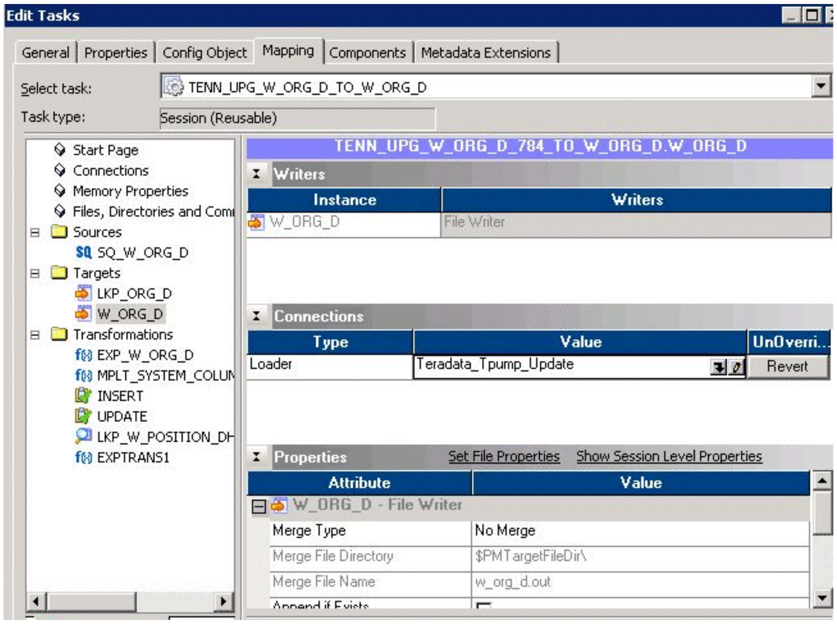
4. Repeat these steps for the following objects:
   - Teradata_Tpump_Update
   - Teradata_Tpump_Insert
   - Teradata_Tpump_Delete
   - Teradata_Tpump_Upsert
5. For Unicode environments, append -c UTF8 to the value for the External Loader Executable attribute for each external loader.

   For example:
   - tpump -c UTF8
6. Click Close to close the Loader Connection Browser.

4.15.2.3.2 How to Specify Teradata Details at the Workflow Level

For each session, you need to specify Teradata details, as follows:

1. In Informatica PowerCenter Workflow Manager, go to the workflow and open the session in the Task Developer pane.
2. Double-click on the session to display the Edit Tasks dialog box.
3. Display the Mapping tab.



Select task: TENN_UPG_W_ORG_D_TO_W_ORG_D

Task type: Session (Reusable)

TENN_UPG_W_ORG_D_784_TO_W_ORG_D.W_ORG_D

Writers

Instance	Writers
W_ORG_D	File Writer

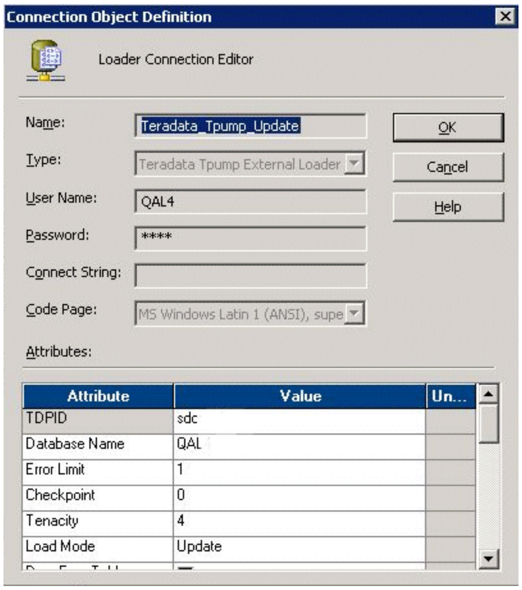
Connections

Type	Value	UnOverri...
Loader	Teradata_Tpump_Update	Revert

Properties

Attribute	Value
W_ORG_D - File Writer	
Merge Type	No Merge
Merge File Directory	$PMTargetFileDir\
Merge File Name	w_org_d.out
Append if Exists	□

4. On the Targets node on the left navigator panel, select the Teradata table name.
5. In the Writers area, select "File Writer" from the Writers drop-down list for the target table.
6. In the Connections area, select "Teradata_Tpump_Upsert" or other "Teradata_Tpump_XXX" as the loader.
7. Click on the edit icon (that is, the pencil icon) next to the Connections - Value field to display the Connection Object Definition dialog box.



Name: Teradata_Tpump_Update

Type: Teradata Tpump External Loader

User Name: QAL4

Password: ****

Connect String:

Code Page: MS Windows Latin 1 (ANSI), supe

Attributes:

Attribute	Value	Un...
TDPID	sdc	
Database Name	QAL	
Error Limit	1	
Checkpoint	0	
Tenacity	4	
Load Mode	Update	

8. Enter the User Name and Password to log in to the Teradata database.

   Note: In the Attributes list, make sure that the following attributes are specified:

   - TDPID
   - Database name
   - Error Database
   - Log Table Database

   If the above attribute values are not inherited from the attributes defined at the connection level, you must specify them.

9. In the Attributes list, specify values for Error Table and Log Table.

4.16 Configuring the Oracle BI Repository Connections

When you first install and setup Oracle Business Intelligence Applications, you must configure the predefined repository connections and variables in the Oracle BI repository. This section explains the predefined connection pools and variables, and how to configure them using Oracle BI Administration Tool, and contains the following sections:

- Section 4.16.1, "About the Predefined Connection Pools in the Oracle Business Analytics Warehouse"
- Section 4.16.2, "How to Configure the Oracle BI Repository Connections"
- Section 4.16.3, "How to Configure Oracle BI Repository Variables"
- Section 4.16.4, "How to Configure Dynamic Data Source Names"
- Section 4.16.5, "Modifying the Oracle BI Repository for Siebel Applications version 6.3 or version 7.5"

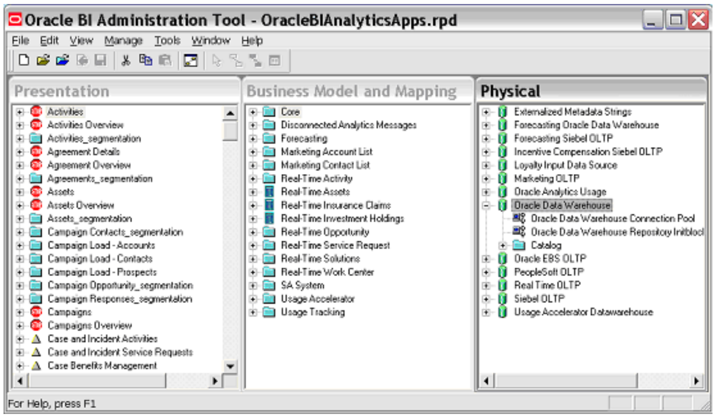
4.16.1 About the Predefined Connection Pools in the Oracle Business Analytics Warehouse

The Oracle BI repository contains the following predefined databases:

- Oracle Data Warehouse
- Oracle EBS OLTP
- PeopleSoft OLTP
- Siebel OLTP
- Loyalty Input Data Source

You configure the connection pools for databases in the Oracle Business Analytics Warehouse from the Physical layer in Oracle BI Administration Tool.

Figure 4–2   BI Repository Physical Layer in Oracle BI Administration Tool


Oracle BI Administration Tool - OracleBIAnalyticsApps.rpd
File Edit View Manage Tools Window Help
Presentation
Activities
Activities Overview
Activities_segmentation
Agreement Details
Agreement Overview
Agreements_segmentation
Assets
Assets Overview
Assets_segmentation
Campaign Contacts_segmentation
Campaign Load - Accounts
Campaign Load - Contacts
Campaign Load - Prospects
Campaign Opportunity_segmentation
Campaign Responses_segmentation
Campaigns
Campaigns Overview
Case and Incident Activities
Case and Incident Service Requests
Case Benefits Management
Business Model and Mapping
Core
Disconnected Analytics Messages
Forecasting
Marketing Account List
Marketing Contact List
Real-Time Activity
Real-Time Assets
Real-Time Insurance Claims
Real-Time Investment Holdings
Real-Time Opportunity
Real-Time Service Request
Real-Time Solutions
Real-Time Work Center
SA System
Usage Accelerator
Usage Tracking
Physical
Externalized Metadata Strings
Forecasting Oracle Data Warehouse
Forecasting Siebel OLTP
Incentive Compensation Siebel OLTP
Loyalty Input Data Source
Marketing OLTP
Oracle Analytics Usage
Oracle Data Warehouse
Oracle Data Warehouse Connection Pool
Oracle Data Warehouse Repository Initbloc
Catalog
Oracle EBS OLTP
PeopleSoft OLTP
Real Time OLTP
Siebel OLTP
Usage Accelerator Datawarehouse
For Help, press F1

The Oracle Data Warehouse physical database has two predefined connection pools:

- **Oracle Data 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 Data Warehouse Repository Initblocks Connection Pool.** You need to configure the Oracle Data 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.

For information about configuring configuration pools, see Section 4.16.2, "How to Configure the Oracle BI Repository Connections".

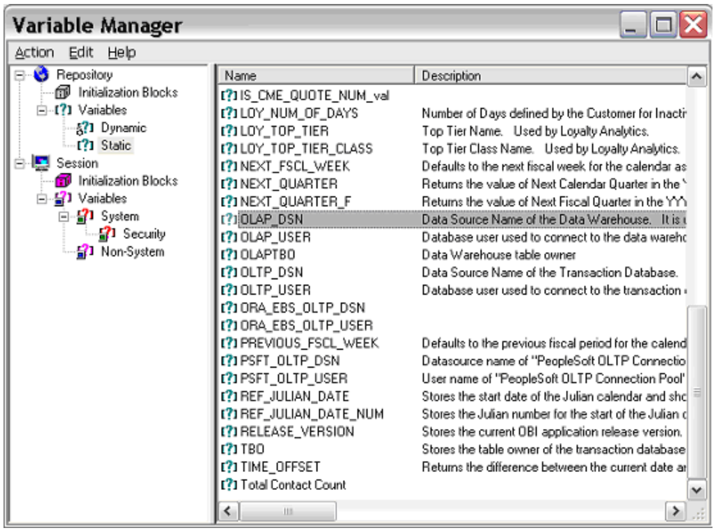
You can also set up dynamic data source names, which allow an Administrator to set one instance of Analytics server to connect to different data warehouses depending on the user. For more information about how to set up dynamic data source names, see Section 4.16.4, "How to Configure Dynamic Data Source Names".

You also need to configure the following Static variables:

- **OLAP_DSN.** The value of the OLAP_DSN static variable is set to the data source name for the warehouse database.
- **OLAP_USER.** The value of the OLAP_USER static variable is set to the database user name for the warehouse database.
- **OLAPTBO.** The value of the OLAPTBO static variable is set to the database table owner for the data warehouse database.

You configure Static variables in Oracle BI Administration Tool using the Variable Manager (choose Manage, then Variables, and expand the Variables\Static node). For more information about configuring Static variables, see Section 4.16.3, "How to Configure Oracle BI Repository Variables".

Figure 4–3 Variable Manager in Oracle BI Administration Tool



Name	Description
IS_CME_QUOTE_NUM_val	
LOY_NUM_OF_DAYS	Number of Days defined by the Customer for Inacti
LOY_TOP_TIER	Top Tier Name. Used by Loyalty Analytics.
LOY_TOP_TIER_CLASS	Top Tier Class Name. Used by Loyalty Analytics.
NEXT_FSCL_WEEK	Defaults to the next fiscal week for the calendar as
NEXT_QUARTER	Returns the value of Next Calendar Quarter in the `
NEXT_QUARTER_F	Returns the value of Next Fiscal Quarter in the YY\
OLAP_DSN	Data Source Name of the Data Warehouse. It is (
OLAP_USER	Database user used to connect to the data wareh(
OLAPTBO	Data Warehouse table owner
OLTP_DSN	Data Source Name of the Transaction Database.
OLTP_USER	Database user used to connect to the transaction (
ORA_EBS_OLTP_DSN	
ORA_EBS_OLTP_USER	
PREVIOUS_FSCL_WEEK	Defaults to the previous fiscal period for the calend
PSFT_OLTP_DSN	Datasource name of "PeopleSoft OLTP Connectio
PSFT_OLTP_USER	User name of "PeopleSoft OLTP Connection Pool"
REF_JULIAN_DATE	Stores the start date of the Julian calendar and shc
REF_JULIAN_DATE_NUM	Stores the Julian number for the start of the Julian (
RELEASE_VERSION	Stores the current OBI application release version.
TBO	Stores the table owner of the transaction database
TIME_OFFSET	Returns the difference between the current date ar
Total Contact Count	

The PeopleSoft OLTP, Siebel OLTP, and Oracle EBS OLTP databases each have two predefined connection pools. The actual databases in the RPD will depend on the modules licensed by the customer. The connection pools and their functions within each database are listed below.

- Oracle EBS OLTP:
  - **Oracle EBS OLTP DBAuth Connection Pool.** The Oracle EBS OLTP DBAuth Connection Pool is used if database authentication is required.
  - **Oracle EBS OLTP Connection Pool.** The Oracle EBS OLTP Connection Pool is used to connect to the Oracle EBS OLTP system.

  You also need to configure the following Static variables:
  - ORA_EBS_OLTP_DSN. The value of the ORA_EBS_OLTP_DSN static variable is set to the data source name for the Oracle EBS OLTP database.
  - ORA_EBS_OLTP_USER. The value of the ORA_EBS_OLTP_USER static variable is set to the database user name for the Oracle EBS OLTP database.
- Siebel OLTP
  - **Siebel OLTP DBAuth Connection Pool.** The Siebel OLTP DBAuth Connection Pool is used if database authentication is required.
  - **Siebel OLTP Connection Pool.** The Siebel OLTP Connection Pool is used to connect to the Siebel OLTP system.

  You also need to configure the following Static variables:
  - OLTP_DSN. The value of the OLTP_DSN static variable is set to the data source name for the Siebel OLTP database.
  - OLTP_USER. The value of the OLTP_USER static variable is set to the database user name for the Siebel OLTP database.

- PeopleSoft OLTP
  - **PeopleSoft OLTP DBAuth Connection Pool.** The PeopleSoft OLTP DBAuth Connection Pool is used if database authentication is required.
  - **PeopleSoft OLTP Connection Pool.** The PeopleSoft OLTP Connection Pool is used to connect to the PeopleSoft OLTP system.

  You also need to configure the following Static variables:

  - PSFT_OLTP_DSN. The value of the OLTP_DSN static variable is set to the data source name for the PeopleSoft OLTP database.
  - PSFT_OLTP_USER. The value of the OLTP_USER static variable is set to the database user name for the PeopleSoft OLTP database.
- Loyalty Input Data Source:
  - **Loyalty Input Connection Pool.** The Loyalty Input Connection Pool is used to connect to the Loyalty OLTP system.

4.16.2 How to Configure the Oracle BI Repository Connections

The section explains how to configure the repository connections used by Oracle Business Intelligence Applications. At a minimum, you need to configure the following:

- the connection pool for the Oracle Data Warehouse.
- the connection pool for each OLTP data source that you want to deploy (for example, Oracle EBS OLTP, PeopleSoft OLTP, Siebel OLTP).

For each connection pool that you configure, you need to:

- Specify the database type for the connection pool (for more information, see Section 4.16.2.1, "How to specify the database type for connection pools").
- Specify the connection details for the connection pool (for more information, see Section 4.16.2.1, "How to specify the database type for connection pools").

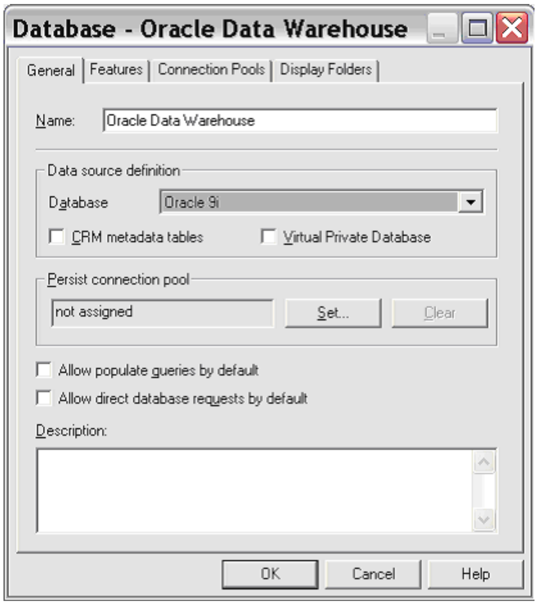
> **Note:** No additional configuration is required for Oracle's JD Edwards EnterpriseOne and JD Edwards World, as both use the standard Oracle Data Warehouse physical connection.

4.16.2.1 How to specify the database type for connection pools

You need to specify the database type for the Oracle Data Warehouse connection pool, and the connection pool for each OLTP data source that you wish to deploy.

To specify the database type for connection pools

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the \OracleBI\Server\Repository folder.
2. In the Physical pane, double-click the Oracle Data Warehouse object.
3. Display the General tab.



Database - Oracle Data Warehouse

Name: Oracle Data Warehouse

Data source definition

Database: Oracle 9i

□ CRM metadata tables □ Virtual Private Database

Persist connection pool: not assigned

□ Allow populate queries by default

□ Allow direct database requests by default

Description:

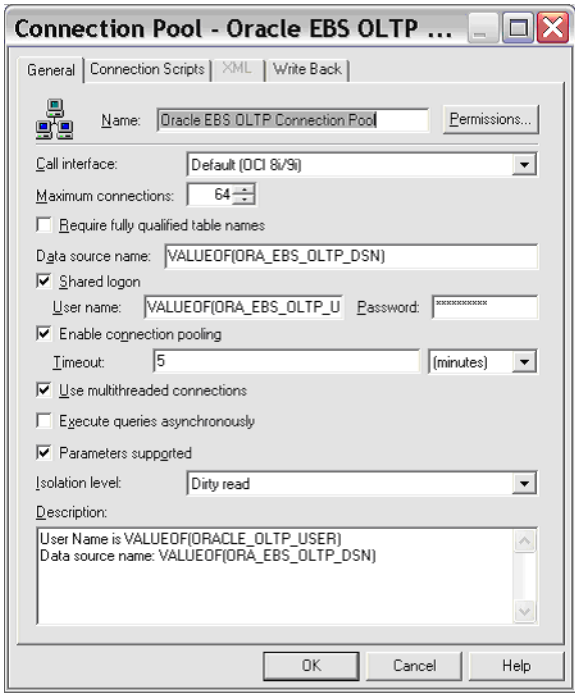
4. Use the **Database** field to specify your database type.
5. Save the repository.
6. Click Yes to Check Global Consistency.
7. Repeat steps 1-6 for each OLTP data source that you want to deploy (for example, Oracle EBS OLTP, PeopleSoft OLTP, and Siebel OLTP).
8. Click OK when the Warnings are displayed.

4.16.2.2 How to specify connection details for connection pools

You need to specify the connection details for the Oracle Data Warehouse connection pool, and the connection pool for each OLTP data source that you want to deploy.

To configure the Oracle BI Repository connection pools

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the \OracleBI\Server\Repository folder.
2. In the Physical pane:
   a. Expand the Oracle Data Warehouse node, and double-click the Oracle Data Warehouse Connection Pool to display the Connection Pool dialog.



Name: Oracle EBS OLTP Connection Pool

Call interface: Default (OCI 8i/9i)

Maximum connections: 64

☐ Require fully qualified table names

Data source name: VALUEOF(ORA_EBS_OLTP_DSN)

☑ Shared logon

User name: VALUEOF(ORA_EBS_OLTP_U Password: **********

☑ Enable connection pooling

Timeout: 5 (minutes)

☑ Use multithreaded connections

☐ Execute queries asynchronously

☑ Parameters supported

Isolation level: Dirty read

Description:

User Name is VALUEOF(ORACLE_OLTP_USER)
Data source name: VALUEOF(ORA_EBS_OLTP_DSN)

b. Use the **Data source name** field to specify 'VALUEOF(ORACLE_OLTP_USER)'.

c. Use the **User name field** to specify 'VALUEOF(ORA_EBS_OLTP_DSN)'.

d. Use the **Password** field to specify your database password.

Note: The variable values specified in the VALUEOF functions are specified in the Variable Manager. For more information about specifying variable values, see Section 4.16.3, "How to Configure Oracle BI Repository Variables".

3. Repeat Steps a. to d. above for the other connection pools that you want to deploy.
4. Save the repository.
5. Click Yes to Check Global Consistency.
6. Click OK when the Warnings are displayed.

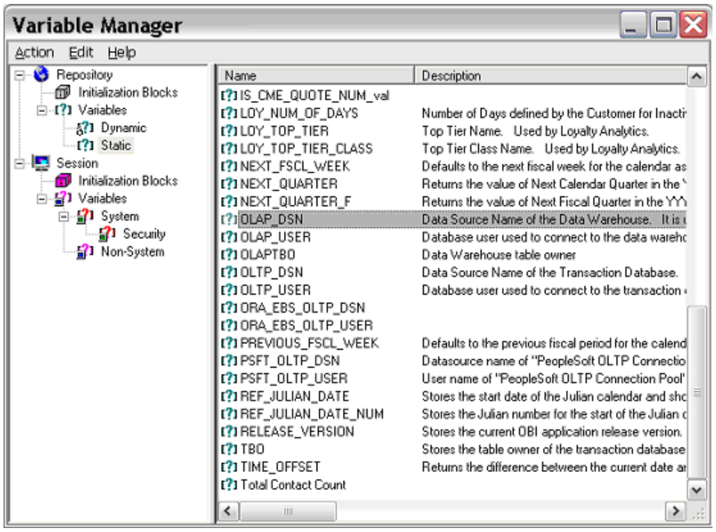
4.16.3 How to Configure Oracle BI Repository Variables

The section explains how to configure the Oracle BI Repository variables.

Note: If you want to deploy multi-calendar with Oracle Financial Analytics, you must have enabled a number of Initialization Blocks that are disabled out-of-the-box. For more information, see Chapter 3 in *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*.

To configure the Oracle BI Repository variables

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the \OracleBI\Server\Repository folder.
2. On the Manage menu, click Variables to display the Variable Manager.
3. Expand the Variables\Static node.



Name	Description
IS_CME_QUOTE_NUM_val	
LOY_NUM_OF_DAYS	Number of Days defined by the Customer for Inacti
LOY_TOP_TIER	Top Tier Name. Used by Loyalty Analytics.
LOY_TOP_TIER_CLASS	Top Tier Class Name. Used by Loyalty Analytics.
NEXT_FSCL_WEEK	Defaults to the next fiscal week for the calendar as
NEXT_QUARTER	Returns the value of Next Calendar Quarter in the
NEXT_QUARTER_F	Returns the value of Next Fiscal Quarter in the YY
OLAP_DSN	Data Source Name of the Data Warehouse. It is
OLAP_USER	Database user used to connect to the data wareho
OLAPTBO	Data Warehouse table owner
OLTP_DSN	Data Source Name of the Transaction Database.
OLTP_USER	Database user used to connect to the transaction
ORA_EBS_OLTP_DSN	
ORA_EBS_OLTP_USER	
PREVIOUS_FSCL_WEEK	Defaults to the previous fiscal period for the calend
PSFT_OLTP_DSN	Datasource name of "PeopleSoft OLTP Connectio
PSFT_OLTP_USER	User name of "PeopleSoft OLTP Connection Pool"
REF_JULIAN_DATE	Stores the start date of the Julian calendar and sho
REF_JULIAN_DATE_NUM	Stores the Julian number for the start of the Julian
RELEASE_VERSION	Stores the current OBI application release version.
TBO	Stores the table owner of the transaction database
TIME_OFFSET	Returns the difference between the current date ar
Total Contact Count	

4. Edit the following variables, which apply to all deployments:

   - GLOBAL_CURRENCY1
   - GLOBAL_CURRENCY2
   - GLOBAL_CURRENCY3
   - OLAP_DSN
   - OLAP_USER
   - OLAPTBO

   Note: The global currency values must match the Source System Parameter values in DAC.

5. Edit the following variables that are specific to your deployment:

   For Oracle EBS OLTP data sources, edit the following variables:

   - ORA_EBS_OLTP_DSN (Oracle EBS-specific)
   - ORA_EBS_OLTP_USER (Oracle EBS-specific)

   For Peoplesoft OLTP data sources, edit the following variables:

   - PSFT_OLTP_DSN (PeopleSoft-specific)
   - PSFT_OLTP_USER (PeopleSoft-specific)

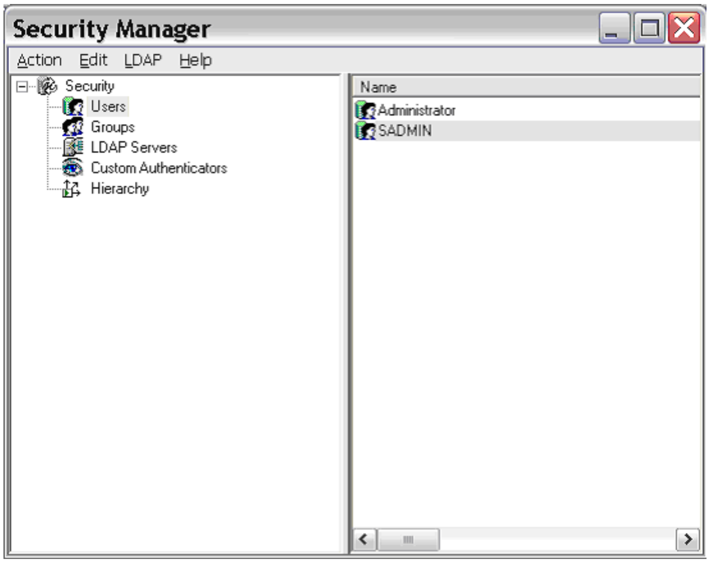
   For Siebel OLTP data sources, edit the following variables:

   - OLTP_DSN (Siebel-specific)
   - OLTP_USER (Siebel-specific)

6. Close the Variables Manager dialog.

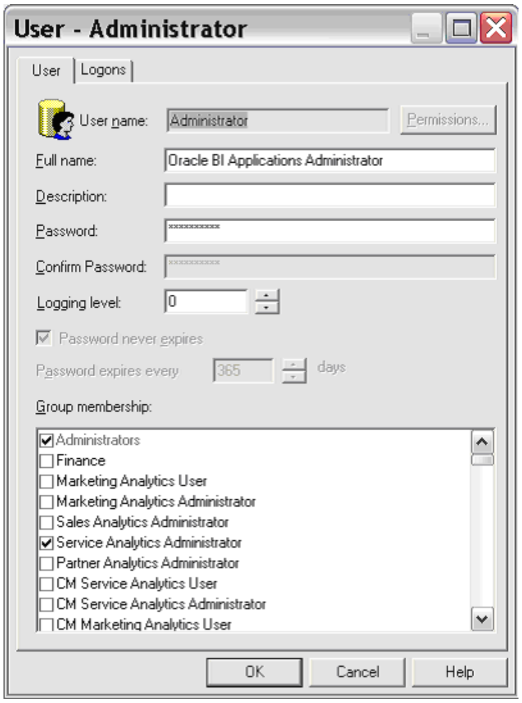
To reset the Oracle BI Repository password

1. On the Manage menu, click Security, to display the Security Manager.



2. In the Security Manager dialog, click Users, and then:

   a. Double-click on Administrator user to display the User dialog.



User name: Administrator

Full name: Oracle BI Applications Administrator

Description:

Password: **********

Confirm Password: **********

Logging level: 0

☑ Password never expires

Password expires every 365 days

Group membership:

☑ Administrators
☐ Finance
☐ Marketing Analytics User
☐ Marketing Analytics Administrator
☐ Sales Analytics Administrator
☑ Service Analytics Administrator
☐ Partner Analytics Administrator
☐ CM Service Analytics User
☐ CM Service Analytics Administrator
☐ CM Marketing Analytics User

   b. Use the **Password** field to change the password.

   c. Repeat steps a- b for the SADMIN user.

   d. Save and close the Security Manager.

3. Save the repository.

4. Click Yes to Check Global Consistency.

5. Click OK when the Warnings are displayed.

4.16.4 How to Configure Dynamic Data Source Names

This sections explains how to create and configure dynamic data source names.

Dynamic data source names allow the Administrator to set one instance of Analytics server to connect to different data warehouses depending on the user. For this you need to have your user authentication based on an external system (like LDAP), and add the following to your repository:

1. Create new session variables: Session_OLAP_DSN and Session_OLAP_USER.
2. Create a Session Init Block which uses 'Oracle Data Warehouse Repository Initblocks Connection Pool' to populate these session variables based on the user login.
3. Add this Initialization Block to the Execution Precedence list of the Authorization Initialization block.
4. Modify the values of Data Source Name and User Name fields in 'Oracle Data Warehouse Connection Pool' to be VALUEOF(Session_OLAP_DSN) and VALUEOF(Session_OLAP_USER) respectively.

   For information about configuring connection pools, see Section 4.16.2.2, "How to specify connection details for connection pools".
5. Update the field password with the same value as of User Name.

4.16.5 Modifying the Oracle BI Repository for Siebel Applications version 6.3 or version 7.5

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

To modify the Oracle BI Repository for Siebel applications version 6.3 and version 7.5

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
   - FINS Cols

4.17 Configuring the SiebelUnicodeDB Custom Property

> **Note:** This procedure is not specific to the Siebel source system. It must be performed for all types of source systems.

If your source to target data movement configuration is Unicode to Unicode, you need to create a custom property called SiebelUnicodeDB on Integration Services. If your source to target data movement configuration is either Code Page to Code Page or Code Page to Unicode, you do not need to create this property. For more information on supported source to target configuration modes, and how to determine the source to target configuration for your environment, see Section 4.3.3, "Code Page and Data Movement Requirements."

To create and set the SiebelUnicodeDB custom property on Integration Services

1. Log into Informatica PowerCenter Administration Console. For information on how to log into PowerCenter Administration Console, see Section A.8, "How to Log Into Informatica PowerCenter Administration Console."
2. Select the Integration Service.
3. In the Properties tab, scroll down to the Custom Properties area, and click Edit.
4. In the Name field, enter the following:

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

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

   Where:

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

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

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

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

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

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

For more information about Relational Connections, see Section 4.15.2, "Configuring Connections in Informatica PowerCenter Workflow Manager".

4.18 Setting Up DAC to Receive Email Notification

This section describes how to set up DAC to receive email notification. It includes the following topics:

- Section 4.18.1, "Configuring Email Recipients in DAC"
- Section 4.18.2, "Configuring Email in the DAC Server"

4.18.1 Configuring Email Recipients in DAC

Follow this procedure to configure email recipients.

To configure email recipients in DAC

1. In DAC, navigate to the Setup view.

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

2. Click the Email Recipients tab.
3. Click New.
4. In the Edit tab below, enter the following information:

Field	Description
Name	Logical name of the user to be notified.
Email Address	Email address where the notification is sent.
Notification Level	The notification levels are as follows: ■ 10 -- Notifies recipient of success or failure of each task. ■ 5 -- Notifies recipient of success of failure of the entire ETL process. ■ 1 -- Notifies recipient that ETL completed successfully.
Inactive	Indicates whether the selected email activation is active or inactive.
Needs Authentication	Read only value that specifies whether the corporate email server requires authentication (this value is set by choosing Tools, then DAC Server Setup).

4.18.2 Configuring Email in the DAC Server

Follow this procedure to configure the email administrator account in 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.

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

To configure the email administrator account in the DAC Server

1. In the DAC menu bar, select Tools, then DAC Server Management, then DAC Server Setup.

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

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

2. Click Yes.

   The Server Configuration dialog box appears.

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

4. Click Save.

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

4.19 Additional Configuration Tasks

This section provides configuration steps that may be required depending on your specific environment. All the sections may not be applicable for your environment.

For mandatory, Siebel-specific configuration tasks, see Section 4.19.4, "Siebel-Specific Configuration Tasks."

For mandatory, Teradata-specific configurations tasks, see Section 4.19.5, "Teradata-Specific Configuration Tasks."

> **Note:** After you complete the tasks in this section and before you run the first ETL load process, you may need to perform additional configuration steps depending on your environment. For additional information about source system-specific and application-specific mandatory configuration tasks, see Section 2.1, "High-Level Overview of Configuring Oracle BI Applications," in *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*.

This section contains the following topics:

- Section 4.19.1, "Setting DAC Source System Parameters"
- Section 4.19.2, "Creating Stored Procedures for DB2-UDB"
- Section 4.19.3, "Configuring Security-Related Initialization Blocks"
- Section 4.19.4, "Siebel-Specific Configuration Tasks"
  - Section 4.19.4.1, "Updating Siebel Transactional Database Schema Definitions"
- Section 4.19.5, "Teradata-Specific Configuration Tasks"
  - Section 4.19.5.1, "Setting Up the Code Page File Property for Unicode Environments on Teradata Databases"
  - Section 4.19.5.2, "Setting Up the HOSTS File for Teradata Installations on Windows"
  - Section 4.19.5.3, "Setting Up the HOSTS File for Teradata Installations on UNIX"
  - Section 4.19.5.4, "Teradata-Specific Installation Checklist"

4.19.1 Setting DAC Source System Parameters

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

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

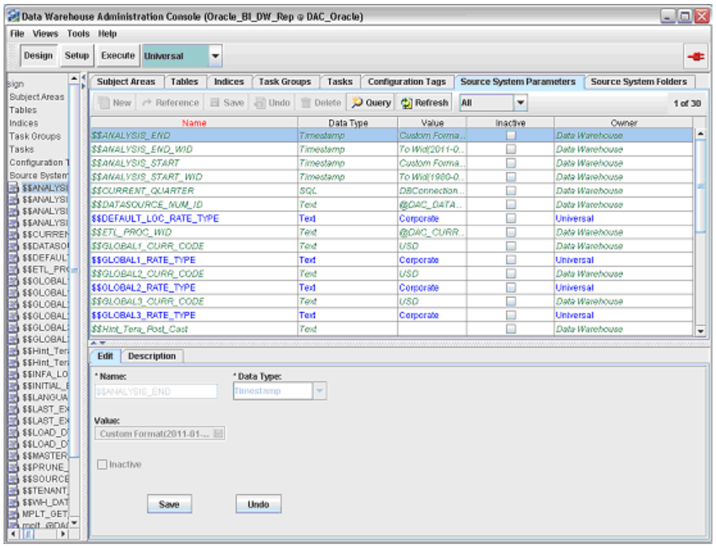
To set DAC source system parameters

1. In DAC, go to the Design view.

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

2. Make sure that you have selected the correct container from the containers drop-down list. You cannot edit preconfigured containers. Make a copy of an existing container in order to make edits.

3. Display the Source System Parameters tab.



Name	Data Type	Value	Inactive	Owner
$$ANALYSIS_END	Timestamp	Custom Forma...	□	Data Warehouse
$$ANALYSIS_END_WID	Timestamp	To Wid(2011-0...	□	Data Warehouse
$$ANALYSIS_START	Timestamp	Custom Forma...	□	Data Warehouse
$$ANALYSIS_START_WID	Timestamp	To Wid(1980-0...	□	Data Warehouse
$$CURRENT_QUARTER	SQL	DBConnection...	□	Data Warehouse
$$DATASOURCE_NUM_ID	Text	@DAC_DATA...	□	Data Warehouse
$$DEFAULT_LOC_RATE_TYPE	Text	Corporate	□	Universal
$$ETL_PROC_WID	Text	@DAC_CURR...	□	Data Warehouse
$$GLOBAL1_CURR_CODE	Text	USD	□	Data Warehouse
$$GLOBAL1_RATE_TYPE	Text	Corporate	□	Universal
$$GLOBAL2_CURR_CODE	Text	USD	□	Data Warehouse
$$GLOBAL2_RATE_TYPE	Text	Corporate	□	Universal
$$GLOBAL3_CURR_CODE	Text	USD	□	Data Warehouse
$$GLOBAL3_RATE_TYPE	Text	Corporate	□	Universal
$$Hint_Tera_Post_Cast	Text		□	Data Warehouse

* Name: $$ANALYSIS_END

* Data Type: Timestamp

Value: Custom Format(2011-01-...

□ Inactive

Note: For information about recommended settings for specific databases, see Chapter 3, "Preinstallation and Predeployment Requirements for Oracle BI Applications".

---

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

5. Click Save.

4.19.2 Creating Stored Procedures for DB2-UDB

DAC uses siebstat and siebtrun stored procedures when running ETL processes. Typically, these stored procedures are available in your transactional database; they might not be available on the data warehouse database.

If you need to install the DB2 stored procedures manually, they are located in the installation directory \OracleBI\dwrep\siebproc\db2udb\. There is a sub-directory for each platform. For example, stored procedures for the Windows platform are stored in the sub-directory \OracleBI\dwrep\siebproc\db2udb\win32\.

Each platform-specific directory contains the following sub-directories:

- \siebproc\ (containing stored procedures for a 32-bit DB2 environment)
- \siebproc64\ (containing stored procedures for a 64-bit DB2 environment)

These directories also contain the files siebproc.sql and sqlproc.ksh, which are used to create the function. For more information, see Section 4.19.2.2, "How to Create DB2 Stored Procedures").

4.19.2.1 How to Verify the Existence of Stored Procedures

This section describes how to verify the existence of stored procedures.

To verify the existence of the stored procedures

- From a DB2 command prompt or control center, issue the following SQL scripts:

```
db2 => create table test_siebproc(id int);
DB20000I  The SQL command completed successfully.

db2 => insert into test_siebproc(id) values (1);
DB20000I  The SQL command completed successfully.

db2 => call siebtrun('TEST_SIEBPROC');
SQL0805N  Package "NULLID.SYSSH200" was not found.  SQLSTATE=51002
```

> **Note:** If you get an error message, you do not have the required stored procedures.

Creating stored procedures must be performed by the database administrator (for more information, see Section 4.19.2.2, "How to Create DB2 Stored Procedures").

4.19.2.2 How to Create DB2 Stored Procedures

This section describes how to create DB2 stored procedures.

To create DB2 stored procedures

1. Copy the DB2 stored procedure directory (i.e. \siebproc\ or \siebproc64\) from the appropriate platform-specific directory to a directory on the DB2 server side.

   For example, for a 32-bit DB2 environment on a Windows platform, you might copy the directory \OracleBI\dwrep\siebproc\db2udb\win32\siebproc\ to the directory d:\Program Files\SQLLIB\function\ on the DB2 server side.

   > **Note:** For more information about the location of DB2 stored procedures, see Section 4.19.2, "Creating Stored Procedures for DB2-UDB").

2. If you copied stored procedures for a 64-bit DB2 environment, on the DB2 server side, rename the \siebproc64\ directory to \siebproc\.

   For example, if you copied stored procedures to d:\Program Files\SQLLIB\function\siebproc64\, rename this directory to d:\Program Files\SQLLIB\function\siebproc\.

Once these procedures are created, you can verify that they exist. After the test is complete, you can drop the table TEST_SIEBPROC.

4.19.3 Configuring Security-Related Initialization Blocks

You may need to configure the security-related initialization blocks that are provided with Oracle BI Applications to work with your particular source system. For information about the Oracle BI Applications security model, see *Oracle Business Intelligence Applications Security Guide*. For information about configuring initialization blocks, see *Oracle Business Intelligence Server Administration Guide*.

4.19.4 Siebel-Specific Configuration Tasks

This section contains tasks you are required to complete if your source system is Siebel. This section contains the following topics:

- Section 4.19.4.1, "Updating Siebel Transactional Database Schema Definitions"

4.19.4.1 Updating Siebel Transactional Database Schema Definitions

> **Note:** The steps in this section are required if are using a Siebel source system.

This section explains how to update Siebel transactional database schema definitions. It contains the following sections:

- Section 4.19.4.1.1, "How to Run the DDLIMP Tool From The Command Line"
- Section 4.19.4.1.2, "How to Apply Siebel CRM Schema Changes For Oracle, DB2/UDB, MSSQL"
- Section 4.19.4.1.3, "How to Apply Siebel CRM Schema Changes for DB2 on OS/390 and z/OS"
- Section 4.19.4.1.4, "How to Apply the Siebel CRM Image Table to a Siebel Transactional Database"
- Section 4.19.4.1.5, "About Delete Triggers"
- Section 4.19.4.1.6, "How to Verify Siebel (CRM) Schema Changes"

4.19.4.1.1 How to Run the DDLIMP Tool From The Command Line When you use the DDLIMP utility from a command line to update schema definitions, refer to the following notes:

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

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

Note: Where <ODBC_DSN> is the ODBC connection created as described in section Section 4.9.4, "Creating ODBC Database Connections for the DAC Client".

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

Additionally you can use the following commands:

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

/Z Y (if the OTTP database is DB2 and Unicode).

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

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

/Y Storage File for DB2/390.

You can obtain a complete list of DDLIMP parameters by running DDLIMP in a command line. DDLIMP is located in the siebsrvr\bin directory. You must have Siebel Server installed to access DDLIMP.

4.19.4.1.2 How to Apply Siebel CRM Schema Changes For Oracle, DB2/UDB, MSSQL To enable change capture for Oracle's Siebel adapters, you use the ddlimp control file to apply Siebel CRM schema changes, which updates the required image tables in the OLTP.

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

To apply Siebel CRM schema changes for Oracle, DB2/UDB and MSSQL databases

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 release 6.3, specify DDL_OLTP_63.CTL as the /F parameter instead of DDL_OLTP.CTL.

   For example:

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

   Notes:

   – /P <PASSWORD> - The password for Oracle's CRM OLTP.

   – /C <ODBC connect string> - The name of the ODBC connect string.

   – For Oracle databases, use the Oracle Merant ODBC Drivers (installed with Oracle BI Applications).

   – In addition, you can use the following commands:

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

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

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

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

/Y - Storage File for DB2/390.

2. Restart all servers.

4.19.4.1.3 How to Apply Siebel CRM Schema Changes for DB2 on OS/390 and z/OS

1. Edit the following parameters in the Storage control files located in \OracleBI\dwrep\Storeage_DDL_OLTP.CTL:

   - %1 - Replace with a 2 character database name.
   - %indBufPool - Replace it with a index buffer pool name.
   - %4kBulfPool - Replace it with a 4k TBS buffer pool name.
   - %32kBufPool - Replace it with a 32K TBS Buffer Pool name.

2. On the machine where Oracle 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.

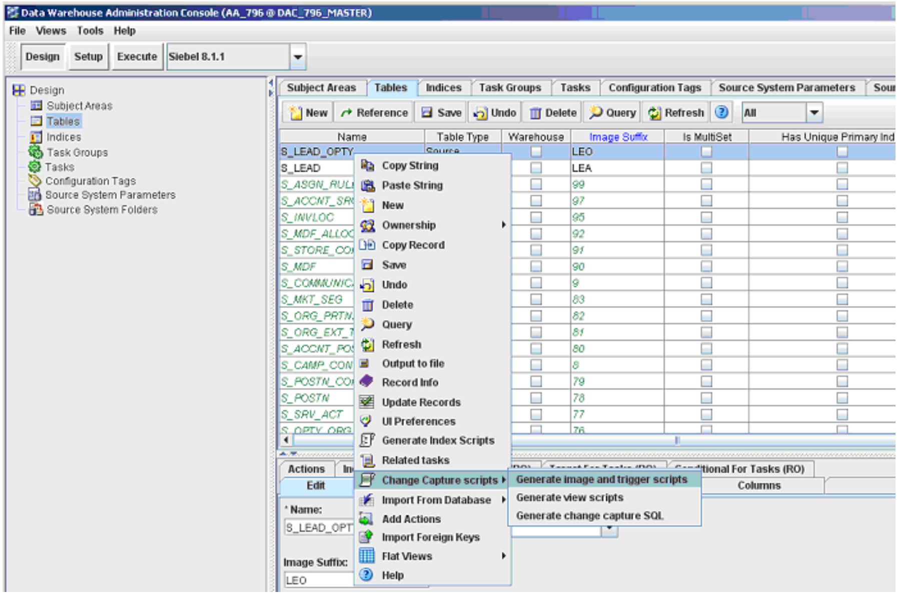
     /Y - Storage File for DB2/390.

3. To create indexes for the tables created, run the following SQL Script from the DB2 command line utility connected to your OLTP Schema:

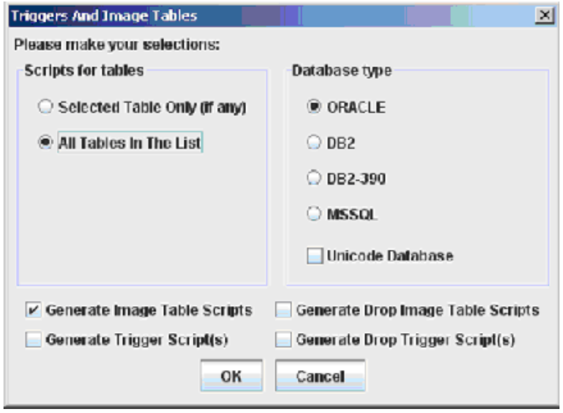
   \OracleBI\Dwrep\Create_OLTP_Db2390_index.sql

4.19.4.1.4 How to Apply the Siebel CRM Image Table to a Siebel Transactional Database

1. In DAC, select Design, then Tables, and query for tables where the **Image Suffix** value is not null.



2. Right click over the returned tables, then select 'Change Capture scripts', then 'Generate image and trigger scripts' to display the Triggers And Image Tables dialog.


Triggers And Image Tables

Please make your selections:

Scripts for tables

○ Selected Table Only (if any)

● All Tables In The List

Database type

● ORACLE

○ DB2

○ DB2-390

○ MSSQL

□ Unicode Database

☑ Generate Image Table Scripts

□ Generate Drop Image Table Scripts

□ Generate Trigger Script(s)

□ Generate Drop Trigger Script(s)

3. At the Triggers And Image Tables dialog, do the following:

   - Select the **All Tables In The List** radio button.
   - Select the **Generate Image Table Scripts** check box.
   - Select the appropriate **Database type**.

4. Click OK to generate the database scripts.

   DAC generates the scripts in a pop-up page.

5. Execute the scripts in your OLTP database.

4.19.4.1.5 About Delete Triggers

Note: Delete triggers are only used with Siebel CRM databases.

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:

DAC can create delete triggers on source tables (refer to *Oracle Business Intelligence Data Warehouse Administration Console User's 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 data warehouse table.

4.19.4.1.6 How to Verify Siebel (CRM) 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.

1. Use a SQL tool to make sure that the following tables were created in the transactional database:
   - S_ETL_R_IMG_xxx
   - S_ETL_I_IMG_xxx
   - S_ETL_D_IMG_xxx
   - S_ETL_PARAM
   - S_ETL_PRD_ATTR
   - S_ETL_PRD_REL

4.19.5 Teradata-Specific Configuration Tasks

This section contains tasks you are required to complete if you are using a Teradata database in your Oracle BI Applications environment.

This section contains the following topics:

- Section 4.19.5.1, "Setting Up the Code Page File Property for Unicode Environments on Teradata Databases"
- Section 4.19.5.2, "Setting Up the HOSTS File for Teradata Installations on Windows"
- Section 4.19.5.3, "Setting Up the HOSTS File for Teradata Installations on UNIX"
- Section 4.19.5.4, "Teradata-Specific Installation Checklist"

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

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

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

1. In Informatica PowerCenter Workflow Manager, drag a Workflow into the Workflow Designer pane.
2. In the Workflow Designer pane, double click the Task (for example, SDE_PSFT_APTermsDimension) to display the Edit Tasks dialog box.
3. Display the Mappings tab.
4. Select Target in the left pane. In the Properties section, click on the **Set File** link.
5. In the Flat Files - Targets dialog box, in the File Properties area, click Advanced.

6. In the Fixed Width Properties - Targets dialog box, in the Code Page area, select UTF-8 encoding of Unicode.

4.19.5.2 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 PowerCenter Integration Services service is installed by adding an entry in the HOSTS file.

To set up the HOSTS file for Teradata installations on Windows

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

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

   For example:

   ```
   172.20.176.208 tdatsvr tdatsvrCOP1
   ```

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

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

4.19.5.3 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 on UNIX

1. How to Set Up the HOSTS File for Teradata Installations on UNIX.
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, refer to the Teradata documentation.

4.19.5.4 Teradata-Specific Installation Checklist

Table 4–5 provides a list of Teradata-specific installation and configuration steps that are performed during the Oracle BI Applications installation and configuration process. You should review this list to make sure that you have performed all of the required Teradata-specific steps.

Table 4–5 Teradata-Specific Installation Checklist for Windows

Action	Link to Topic
Review Teradata-specific database guidelines.	Section 3.5, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse".
Create or drop data warehouse tables.	Section 4.12.2, "Creating Data Warehouse Tables on a Teradata Database"
Create or restore the Informatica Repository using the Informatica Repository Manager. You cannot create or restore the Informatica Repository using DAC. You must use Informatica PowerCenter Administration Console.	Section 4.7, "Restoring the Prebuilt Informatica Repository"
Configure the source and data warehouse database relational connections in Informatica PowerCenter Workflow Manager. You cannot use the DAC's Data Warehouse Configurator to configure relational connections.	Section 3.5, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse"
Set the Code Page File Property for Unicode Environments.	Section 4.19.5.1, "Setting Up the Code Page File Property for Unicode Environments on Teradata Databases"
Set up the HOSTS file.	Section 4.19.5.2, "Setting Up the HOSTS File for Teradata Installations on Windows"

4.20 About Running A Full Load ETL

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

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

For an example of using DAC to run a full load ETL, see Section 4.20.1, "An Example of Running a Full Load ETL."

4.20.1 An Example of Running a Full Load ETL

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

1. In the DAC menu bar, select File, then New Source System Container, to display the New Source System Container dialog box, and specify details of the new container.

Note: You cannot make any changes to the preconfigured containers. You must make a copy of a container before you can make any changes to it.

For more information about logging into DAC, see Section A.1, "How to Log into DAC." For more information about the source system container functionality in DAC, see *Oracle Data Warehouse Administration Console User's Guide*.

2. Select the Create as a Copy of Existing Container radio button, and select the appropriate container from the Existing Containers drop-down list, then click OK.
3. In the Design view, display the Source System Parameters tab and set the parameters appropriately.
4. In the Setup view, display the DAC System Properties tab and set the properties appropriately.
5. Display the Execute view, and display the Execution Plan tab.
6. Click New, display the Edit tab, and use the Name field to specify a name for the ETL process.
7. Display the Subject Areas tab and click Add/Remove to display the Choose Subject Areas dialog box.
8. Select the new container that you created in step 4 from the container drop-down list at the top of the Choose Subject Areas dialog box.
9. Select Financials - Receivables, click Add, then click OK.
10. Display the Parameters tab, and click Generate.
11. On the Parameters tab, edit the parameters as follows:
    - Edit the value of DBConnection_OLAP and set it to the same value as the name of the OLAP database that you specified in the Physical Data Source dialog box (for example, DataWarehouse).
    - Edit the value of DBConnection_OLTP and set it to the same value as the name of the OLTP database that you specified in the Physical Data Source dialog box (for example, ORA_11_5_8).
    - If there is a FlatFileConnection parameter, edit the value of FlatFileConnection and set it to the same value as the name of the flat file data source that is specified in the Physical Data Source dialog box (for example, ORA_11_5_8_Flatfile).
12. On the Execution Plans tab, click Build.
13. On the Execution Plans tab, click Run Now.

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

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

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

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

Part III

Appendices

Part III contains the following sections:

- Appendix A, "Supporting Tasks for DAC and Informatica PowerCenter"
- Appendix B, "Localizing Oracle Business Intelligence Deployments"
- Appendix C, "Integrating Interactive Dashboards and Operational Applications Data"
- Appendix D, "Configuring Metadata for Oracle Business Intelligence Applications"
- Appendix E, "Using Oracle Business Analytics Warehouse Exception Reports"
- Appendix F, "About the Versioned Informatica Repository"

Note: For a high-level road map for installation, configuration, and customization steps for Oracle BI Applications, see Section 2.4, "Roadmap to Installing and Setting Up Oracle BI Applications."

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

A

Supporting Tasks for DAC and Informatica PowerCenter

This section contains additional tasks relating to the DAC Client and DAC Server, and Informatica PowerCenter Client Tools. It contains the following topics:

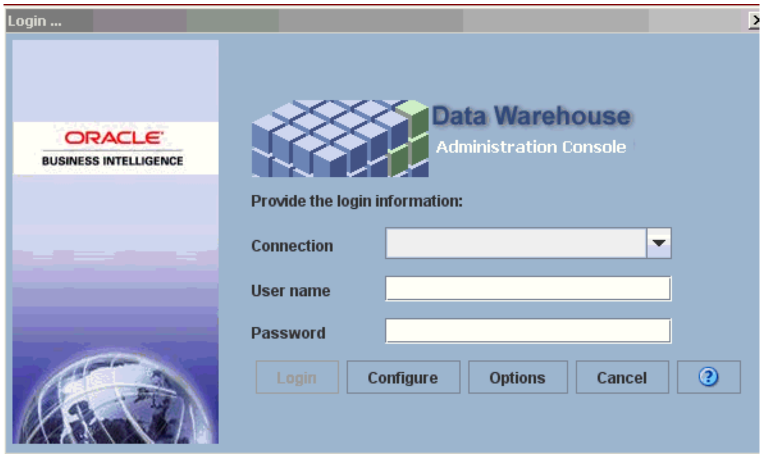
A.1 How to Log into DAC

Before you can log into the DAC, you first need to have created a DAC connection, which is a set of stored login details. For information about creating a DAC connection, see Section 4.11.1, "Logging into DAC and Creating a Connection to the DAC Repository."

To log into DAC

1. Launch the DAC Client by doing one of the following:
   - Double-clicking the DAC Client icon on your desktop
   - Navigating to the \bifoundation\dac directory and double-clicking the startclient.bat file
   - On the Windows taskbar, click Start, then Programs, then Data Warehouse Administration Console, and then Client.

   The Login... dialog is displayed.



2. In the Login... dialog, select a connection for the required DAC Repository from the Connection drop-down list.

   For instructions on creating a connection to the DAC Repository, see Section 4.11.1, "Logging into DAC and Creating a Connection to the DAC Repository."

3. In the Table owner name field, enter the database user name for the DAC Repository database.

4. In the Password field, enter the database password for the DAC Repository database.

5. Click Login to launch the DAC.

   The DAC Client launches and connects to the DAC Repository.

For more information about using the DAC Client, see *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

A.2 About DAC User Account Management

The User Management feature includes three roles: Administrator, Developer, and Operator. As shown in Table A–1, each role has a set of permissions that determines what DAC functionality the role can access.

The User Management dialog box enables a user with the Administrator role to create user accounts. A user account includes a unique identifier, password, and one or more roles. The Administrator can also inactivate a user account. For instructions on managing user accounts, see Section A.2.1, "Creating, Deleting and Inactivating User Accounts."

Upon the initial login to a new DAC installation, a user account with the Administrator role is automatically created. This default user account name is Administrator, and the default password is Administrator. It is recommended that after the initial login, the user change the default password.

Note: A user with the Administrator role must distribute the DAC Repository database authentication file to user accounts that need to access the DAC Repository. For information about the authentication file, see "DAC Repository Database Authentication File".

Table A–1 User Account Roles and Permissions

Role	Permissions
Administrator	Read and write permission on all DAC tabs and dialog boxes.
Developer	Read and write permission on the following: ▪ All Design view tabs ▪ All Setup view tabs ▪ Export dialog box ▪ New Source System Container dialog box ▪ Rename Source System Container dialog box ▪ Delete Source System Container dialog box
Operator	Read and write permission on all Setup view tabs

A.2.1 Creating, Deleting and Inactivating User Accounts

The User Management feature enables a user with the Administrator role to create, delete, and inactivate user accounts.

To create a user account

1. From the toolbar, select File, then User Management.
2. In the User Management dialog box, click New.
3. In the new record field, do the following:
   a. Enter a unique Name and Password.
   b. Click in the Roles field, and then select the roles you want to associate with this user account.
4. Click Save.
5. Click Close to exit the User Management dialog box.
6. Distribute the authentication file for the database where the DAC Repository resides to the user account.

   For more information about authentication files, see "DAC Repository Database Authentication File".

To delete a user account

1. From the toolbar, select File, then User Management.
2. In the User Management dialog box, select the user account you want to delete.
3. Click Delete.
4. Click Close to exit the User Management dialog box.

To inactivate a user account

1. From the toolbar, select File, then User Management.
2. In the User Management dialog box, select the user account you want to inactivate.
3. Click the Inactive check box.
4. Click Save.
5. Click Close to exit the User Management dialog box.

A.3 How to Set DAC System Properties

This section provides instructions for setting the DAC System Properties.

To set DAC System Properties

1. Log in to DAC.

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

2. From the Views menu, select Setup, then DAC System Properties tab.

3. Set the appropriate values for the following system properties:

Property	Value Required
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.
DAC Alternate Server Hosts	Host name of the machine where the alternate DAC server resides. The alternate DAC server is used for failover purposes. The DAC client cannot talk to the alternate server unless the main DAC server is not running.
DAC Server Hosts	Host name of the machine where the DAC server resides. You cannot use an IP address for this property. 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. This property also takes the value localhost. 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, or AIX. 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 serverSetupPrompt.sh.
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.

Property	Value Required
Drop and Create Change Capture Views	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, using the following rules: ■ In full mode: During the change capture phase, views will be dropped and created as full views. During the change capture sync process, incremental views will be generated. ■ In incremental mode: If the view exists, it will not be dropped and created. If the view does not exist, the incremental view will be created.
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 indexes, and analyze statements. This option should be used for debugging purposes only and not used in a production environment.
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.
InformaticaFileParameterLocation	Directory where the Informatica parameter file is stored.
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. If the server is launched as a Windows service, the logging information is directed to the service log. If the server is launched with the command shell not visible, all logging information is deleted
Repository DB Pool Size	Indicates the maximum number of connections to the DAC repository that the server will maintain.
Scheduler.Poll.Interval	Frequency (in seconds) the DAC server polls for changes in the schedule configuration.

Property	Value Required
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.

A.4 How to Start and Stop the DAC Server

This section explains how to start and stop the DAC Server.

To start or stop the DAC Server on Windows

1. If you installed the DAC Server with the DAC installer, choose the Windows Start menu, then Programs, then Oracle Business Intelligence, then Oracle DAC, and then Start Server or Stop Server.

   Alternatively, navigate to the \bifoundation\dac folder and double-click the startserver.bat or stopserver.bat file, depending on which action you want to perform.

2. If you installed the DAC Server by copying the \DAC\ directory to a machine, run the \DAC\startserver.bat script or \DAC\stopserver.bat script.

To start the DAC Server on UNIX and Linux

1. In bash-related shells, issue the following command:

   ```
   ./startserver.sh
   ```

2. On AIX, use startserver_aix.sh.

To run the DAC Server in the background

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

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

  The nohup command allows the DAC Server to run in the background on UNIX even if the DAC Client is disconnected. **Note**: To stop DAC Server running as a background process, use stopserver.sh or stopserver.csh.

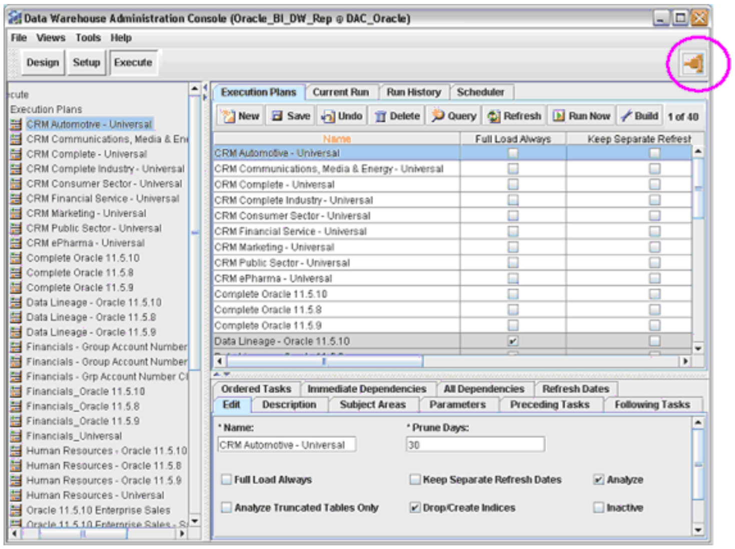
To stop the DAC Server on UNIX and Linux

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

  ```
  ./stopserver.sh
  ```

  Note: When you execute ./stopserver.csh or ./stopserver.sh, the server will issue a warning about the shutdown request. When the server receives the request, it will shut down even if there is an ETL in progress. The statuses for the ETL run will not be set correctly. The next time the DAC Server starts, it will set the status to Failed for any uncompleted run.

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


Data Warehouse Administration Console (Oracle_BI_DW_Rep @ DAC_Oracle)
File
Views
Tools
Help
Design
Setup
Execute
cute
Execution Plans
CRM Automotive - Universal
CRM Communications, Media & En
CRM Complete - Universal
CRM Complete Industry - Universal
CRM Consumer Sector - Universal
CRM Financial Service - Universal
CRM Marketing - Universal
CRM Public Sector - Universal
CRM ePharma - Universal
Complete Oracle 11.5.10
Complete Oracle 11.5.8
Complete Oracle 11.5.9
Data Lineage - Oracle 11.5.10
Data Lineage - Oracle 11.5.8
Data Lineage - Oracle 11.5.9
Financials - Group Account Number
Financials - Group Account Number
Financials - Grp Account Number Cl
Financials_Oracle 11.5.10
Financials_Oracle 11.5.8
Financials_Oracle 11.5.9
Financials_Universal
Human Resources - Oracle 11.5.10
Human Resources - Oracle 11.5.8
Human Resources - Oracle 11.5.9
Human Resources - Universal
Oracle 11.5.10 Enterprise Sales
Oracle 11.5.10 Enterprise Sales - S
Execution Plans
Current Run
Run History
Scheduler
New
Save
Undo
Delete
Query
Refresh
Run Now
Build
1 of 40
Ordered Tasks
Immediate Dependencies
All Dependencies
Refresh Dates
Edit
Description
Subject Areas
Parameters
Preceding Tasks
Following Tasks

Name	Full Load Always	Keep Separate Refresh
CRM Automotive - Universal	□	□
CRM Communications, Media & Energy - Universal	□	□
CRM Complete - Universal	□	□
CRM Complete Industry - Universal	□	□
CRM Consumer Sector - Universal	□	□
CRM Financial Service - Universal	□	□
CRM Marketing - Universal	□	□
CRM Public Sector - Universal	□	□
CRM ePharma - Universal	□	□
Complete Oracle 11.5.10	□	□
Complete Oracle 11.5.8	□	□
Complete Oracle 11.5.9	□	□
Data Lineage - Oracle 11.5.10	☑	□

* Name: CRM Automotive - Universal

* Prune Days: 30

□ Full Load Always
□ Keep Separate Refresh Dates
☑ Analyze
□ Analyze Truncated Tables Only
☑ Drop/Create Indices
□ Inactive

A.5 Enabling DAC Client Communication with Informatica PowerCenter

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

In order for the DAC Client to be able to use the pmrep and pmcmd programs, the path of the Informatica Domain file 'domains.infa' must be defined in the environment variables on the DAC Client machine.

When you install DAC using the DAC installer, the Informatica Domain file is defined in the environment variables on the DAC Client machine. Therefore, if you installed DAC using the DAC installer, you do not need to perform the procedures in this section.

You should only perform the procedures in this section if you installed the DAC Client by copying the \DAC folder from one machine to another.

This section includes the following topics:

- Section A.5.1, "How to Define the Informatica Domains File Path in the DAC Client Environment Variables"

- Section A.5.2, "How to Verify the DAC Client Is Able to Use pmrep and pmcmd"

A.5.1 How to Define the Informatica Domains File Path in the DAC Client Environment Variables

In order for the DAC Client to be able to use the pmrep and pmcmd programs, the path of the Informatica Domain file 'domains.infa' must be defined in the environment variables on the DAC Client machine.

Note: When you use the DAC installer to install the DAC Client, this configuration is done automatically.

To define the Informatica Domains File path in the DAC Client environment variables

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

   For example, <drive>:\Informatica\PowerCenter8.6.1.

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

   - On Windows, display the Windows Environment Variables dialog box (that is, from the Windows Control Panel, select System, then Advanced, then Environment Variables), and create a System variable with the following values:

     – Variable name: INFA_DOMAINS_FILE

     – Variable value: <*directory path of domains file*>\domains.infa

     The path should include the name of the file. For example, '<drive>:\Informatica\PowerCenter8.6.1\domains.infa'.

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

   In the Windows System Properties > Environment Variables dialog box, add the path of the Informatica \Client\bin directory to the PATH environment variable. For example: <drive>:\Informatica\PowerCenter8.6.1\client\bin.

A.5.2 How to Verify the DAC Client Is Able to Use pmrep and pmcmd

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

If pmrep and pmcmd are not recognized, then:

- Ensure Hotfix 6 has been applied to Informatica PowerCenter 8.6.1.
- Verify that the INFA_DOMAINS_FILE variable points to the domains.infa file located in the Informatica directory.
- Verify that the PATH variable includes the path to the Informatica binaries (\Informatica\PowerCenter\client\bin), and that pmrep.exe and pmcmd.exe exist in the \bin folder.

A.6 Enabling DAC Server Communication with Informatica PowerCenter

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

- pmrep is used to communicate with PowerCenter Repository Services.
- pmcmd is used to communicate with PowerCenter Integration Services to run the Informatica workflows.

The pmrep and pmcmd programs are installed during the PowerCenter Services installation in the bin folder of the server directory.

For the DAC Server to be able to communicate with PowerCenter Services using the pmrep and pmcmd, the path of the Informatica Domain file 'domains.infa' must be defined and certain environment variables must be set. When you install the DAC Server on Windows or Linux using the DAC installer, this configuration is done automatically. Therefore, if you installed the DAC Server using the DAC installer, you do not need to perform the procedures in this section.

If you installed the DAC Server on UNIX, you need to perform the procedure in Section A.6.2, "How to Set Environment Variables for DAC Server Communication on UNIX."

This section includes the following topics:

- Section A.6.1, "How to Set Environment Variables for DAC Server Communication on Windows"
- Section A.6.2, "How to Set Environment Variables for DAC Server Communication on UNIX"
- Section A.6.3, "How to Verify Java JDK Availability and Version"

A.6.1 How to Set Environment Variables for DAC Server Communication on Windows

Follow this procedure to set environment variables on Windows.

Note: When you use the DAC installer to install the DAC Server, this configuration is done automatically.

To set the environment variables on Windows

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

   For example, D:\Informatica\PowerCenter8.6.1.

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

   - On Windows, display the Windows Environment Variables dialog box (that is, from the Windows Control Panel, select System, then Advanced, then Environment Variables), and create a System variable with the following values:

     – Variable name: INFA_DOMAINS_FILE

     – Variable value: <<*directory path of domain file*>\domains.infa>

       The Variable value should include the domains.infa file name. For example, D:\Informatica\PowerCenter8.6.1\domains.infa.

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

In the Windows System Properties > Environment Variables dialog box, add the path of the Informatica \server\bin directory to the PATH environment variable. For example: <drive>:\Informatica\PowerCenter8.6.1\server\bin.

How to Verify the DAC Server Is Able to Use pmrep and pmcmd

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

If pmrep and pmcmd are not recognized, then:

- Ensure Hotfix 6 has been applied to Informatica PowerCenter 8.6.1.
- Verify that the INFA_DOMAINS_FILE variable points to the domains.infa file located in the Informatica directory.
- Verify that the PATH variable includes the path to the Informatica binaries (\Informatica\PowerCenter\server\bin).

A.6.2 How to Set Environment Variables for DAC Server Communication on UNIX

This procedure is required for DAC Server deployments on UNIX.

Use the dac_env.sh file to set the appropriate environment variables on UNIX. For more information on the DAC Server scripts, see Section A.7, "About the DAC Server Shell Scripts."

Note: When you use the DAC installer to install the DAC Server on Linux, this configuration is done automatically.

To set environment variables for DAC Server Communication on UNIX

1. Navigate to the /DAC directory.
2. Open the dac_env.sh file for editing.

   Note: The files dac_env_714.sh and dac_env_811.sh are backup files used for Information PowerCenter deployments prior to version 8.6.1.
3. Set the value for each instance of the variable `%INFORMATICA_SERVER_LOCATION%` to the directory that contains the Informatica server directory. (Do not include the server directory.)

   For example, change the following line from:

   ```
   DAC_PMCMD_PATH=%INFORMATICA_SERVER_LOCATION%/server/bin
   ```

   to

   ```
   DAC_PMCMD_PATH=Informatica/PowerCenter8.6.1/server/bin
   ```
4. Set the value for each instance of the variable `%DOMAINS.INFA_FILE_LOCATION%` to the directory that contains the domains.infa file. (Include the file name in the value.)

   For example, change the following line from:

   ```
   export INFA_DOMAINS_FILE=%DOMAINS.INFA_FILE_LOCATION%
   ```

   to

   ```
   export INFA_DOMAINS_FILE=Informatica/PowerCenter8.6.1/domains.infa
   ```
5. If necessary, uncomment the locale settings.

How to Verify the DAC Server on UNIX or Linux Is Able to Use pmrep and pmcmd

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

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

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

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

```
export JAVA_HOME=/opt/java1.6
```

To:

```
JAVA_HOME=/opt/java1.6
export JAVA_HOME
```

A.6.3 How to Verify Java JDK Availability and Version

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

1. Invoke config.sh to set environment.

   For example:

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

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

   ```
   $JAVA -version
   ```

   The Java version is returned.

   If you receive a 'command not found' error message, or the Java version is lower than 1.6, then the JAVA_HOME parameter in config.sh is pointing to a non-existent or incorrect Java JDK location.

A.7 About the DAC Server Shell Scripts

Shell scripts are provided in the *.sh format. Table A–2 lists the available shell scripts and their usage. These files contain comments that provide information about how to configure the scripts.

Table A–2 Oracle Business Analytics Warehouse Scripts for Bash and C Shells

Script	Purpose
config.sh	Configures the environment variables for DAC_HOME and JAVA_HOME.
dacCmdLine.sh	Invokes DAC commands on the DAC Server. See the file for usage information.
dac_env.sh	Configures environment variables for the DAC Server.
serverSetupPrompt.sh	Configures DAC metadata repository connection information.

Table A–2 (Cont.) Oracle Business Analytics Warehouse Scripts for Bash and C Shells

Script	Purpose
serverinfo.sh	Displays the DAC Server version.
startserver.sh	Starts the DAC Server on machines running Solaris or Linux, and HP.
startserver_aix.sh	Starts the DAC Server on AIX machines.
stopserver.sh	Shuts down the DAC Server.

Note: The files listed in Table A–2 need to have read, write, and execute permissions. If the files do not have these permissions, modify them using the chmod command.

When you copy across these files, use a MS-DOS to UNIX conversion tool, convert the script files to UNIX format (that is, remove the carriage return and line feed characters). There are many MS-DOS to UNIX conversion tools that are freely available for download on the Internet. Alternatively, you can manually remove the carriage return and line feed characters from the script files.

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

```
export JAVA_HOME=/opt/java1.6
```

To:

```
JAVA_HOME=/opt/java1.6
export JAVA_HOME
```

Follow these procedures to start and stop the DAC Server on UNIX. Before you can start the DAC Server, you must have already configured the config.sh and dac_env.sh files.

To start the DAC Server

1. In bash-related shells, issue the following command:

   ```
   ./startserver.sh
   ```

To run the DAC Server in the background

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.

To stop the DAC Server

1. In bash-related shells, issue the following command:

   ```
   ./stopserver.sh
   ```

> **Note:** When you execute ./stopserver.sh, the server will issue a warning about the shutdown request. When the server receives the request, it will shut down even if there is an ETL in progress. The statuses for the ETL run will not be set correctly. The next time the DAC Server starts, it will set the status to Failed for any uncompleted run.

A.8 How to Log Into Informatica PowerCenter Administration Console

Informatica PowerCenter Administration Console is installed on the machine that hosts the gateway node for the PowerCenter domain. For a single machine install of PowerCenter Services as described in this chapter, PowerCenter Administration Console is installed along with PowerCenter Services.

To log into PowerCenter Administration Console

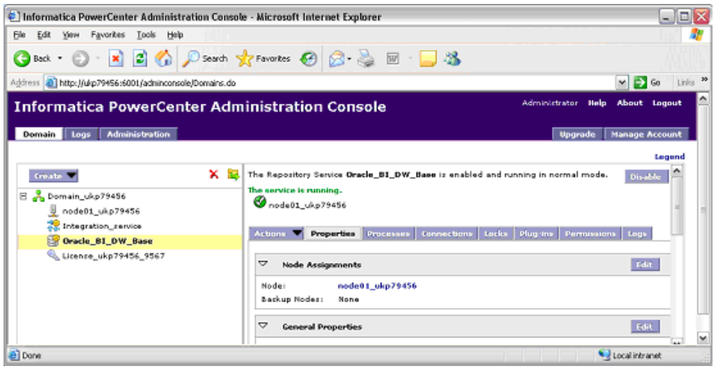
1. In Microsoft Internet Explorer or Mozilla Firefox, access the following URL:

   http://<*gateway host*>:<*domain port*>

   Where:

   - <*gateway host*> is the name of the machine on which the gateway node has been configured; for a single-machine installation of PowerCenter Services it is the name of the machine on which PowerCenter Services has been installed.
   - <*domain port*> is the port number for the gateway. The default port number is 6001.

2. In the login page, enter the domain username and password that was specified when you created the domain during installation of PowerCenter Services.



Informatica PowerCenter Administration Console - Microsoft Internet Explorer

Address: http://ukp79456:6001/adminconsole/Domains.do

Informatica PowerCenter Administration Console

Domain_ukp79456
node01_ukp79456
Integration_service
Oracle_BI_DW_Base
License_ukp79456_9567

The Repository Service **Oracle_BI_DW_Base** is enabled and running in normal mode.

The service is running.
node01_ukp79456

Node Assignments

Node: node01_ukp79456
Backup Nodes: None

General Properties

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:

- Section B.1, "Process of Maintaining Translation Tables for Oracle BI"
- Section B.2, "About Translating Presentation Services Strings"
- Section B.3, "Changing the Default Currency in Analytics Applications"

B.1 Process of Maintaining Translation Tables for Oracle BI

The Oracle Business Intelligence Presentation layer supports multiple translations for any column name. When working with Oracle BI Answers or rendering a dashboard, users see their local language strings in their reports. For example, English-speaking and French-speaking users would see their local language strings in their reports. There are two kinds of application strings requiring translation in Oracle Business Intelligence:

- Metadata

  Metadata strings are analytics-created objects in the Oracle Business Intelligence repository such as subject areas, 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 XML caption files. 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:

- Section B.1.1, "Upgrading Oracle Business Intelligence Seed Data for Non-English Locales"
- Section B.1.2, "Externalizing Customer Metadata Strings"
- Section B.1.3, "Adding Custom Translations to the W_LOCALIZED_STRING_G Table"

B.1.1 Upgrading Oracle Business Intelligence Seed Data for Non-English Locales

If Oracle Business Intelligence data in your deployment is to be viewed in a language other than English, you must also import Locale seed data into a data warehouse table called W_LOCALIZED_STRING_G. This process must be performed once for each language the application users might select to run their web client.

During the Oracle Business Intelligence installation, a folder named `$INSTALLDIR\SeedData` was created, which contains a sub folder for each language. Within each language sub folder is a .dat file (the data to be imported) and an .inp file (the WHERE clause governing the import).

B.1.1.1 Importing Locale Seed Data Into The Translation Table (W_LOCALIZED_STRING_G)

If the primary language being used is not English, you may have to import additional locale seed data (depending on the number of languages you use) as shown in the following procedure. Note: This procedure requires the use of the dataimp utility, which can only be used on 32-bit operating systems.

> **Note:** This procedure should be performed only by a BI Administrator.

To verify creation of Translation Table (W_LOCALIZED_STRING_G) and corresponding indexes:

1. Verify that the Business Analytics Warehouse contains the W_LOCALIZED_STRING_G table.
2. Lookup the definitions of the indexes in DAC and create them manually in the Business Analytics Warehouse. The names of the indexes are as follows:
   - W_LOCAL_STRING_G_U1
   - W_LOCAL_STRING_G_P1
   - W_LOCAL_STRING_G_M1
   - W_LOCAL_STRING_G_M2

Note: It is better to add these indexes to W_LOCALIZED_STRING_G prior to importing the locale seed data in the next section, in order to safeguard against inadvertently duplicating the data in the table.

To import Locale seed data into the Translation Table (W_LOCALIZED_STRING_G)

1. Open a command window and navigate to `$INSTALLDIR\SeedData\bin` folder.
2. Run the import command in step 3 after replacing these connection parameters with the values appropriate to your database environment:
   - UserName
   - Password
   - ODBCDataSource
   - DatabaseOwner
3. Run the import command:

```
$INSTALLDIR\SeedData\bin\dataimp /u $UserName /p $Password /c "$ODBCDataSource"
/d $DatabaseOwner /f $INSTALLDIR\SeedData\<XX>\analytics_seed_<XXX>.dat /w y /q
100 /h Log /x f /i $INSTALLDIR\SeedData\<XX>\metadata_upgrade_<XXX>_
<DBPlatform>.inp /l metadata_upgrade_<XXX>.log
```

> **Note:** Replace the XX with the Oracle Business Intelligence two-letter language code (_fr, _it) and the XXX with the Siebel Systems three-letter code (FRA, ITA).

4. When you have finished importing the Locale seed data into the Translation Table (W_LOCALIZED_STRING_G), configure the initialization block in the Oracle BI Repository using the Oracle BI Administration Tool to connect to the database where this table resides.

> **Note:** Unicode connectivity can be used to access databases that do not support Unicode.

B.1.2 Externalizing Customer Metadata Strings

Metadata Strings are loaded by the Oracle BI Server from a database table. In the case of Oracle Business Intelligence applications, this table is W_LOCALIZED_STRING_G in the data warehouse. The initialization block 'Externalize Metadata Strings' loads the strings for the Server. It is recommended that you run a test to make sure that this initialization block runs successfully. An example of the translation table is shown in Table B–1.

Table B–1 Example of W_LOCALIZED_STRING_G Translation Table

MSG_NUM	MSG_TEXT	LANG_ID
CN_Customer_Satisfaction	Customer Satisfaction	ENU
CN_Customer_Satisfaction	Kundenzufriedenheit	DEU
CN_Customer_Satisfaction	Satisfação do cliente	PTB

By default, the Oracle Business Intelligence repository is configured to run in English only. To deploy in any other language, you must externalize the metadata strings, as described in the following procedure.

To externalize metadata strings in the Oracle Business Intelligence repository

1. Stop the Oracle BI Server.
2. Using the Oracle BI Administration Tool in offline mode, open OracleBIAnalyticsApps.rpd.
3. Select the entire Presentation layer and right-click the mouse to display the menu.
   - From the pop-up menu, select Externalize Display Names. (A check mark appears next to this option the next time you right-click on the Presentation layer.)
   - Unselect the Presentation layer.

> **Note:** When Externalize Display Names is checked, all metadata strings are read from the W_LOCALIZED_STRING_G table in the data warehouse.

4. In the Physical layer, select the Externalized Metadata Strings database icon. Expand the tree.
5. Double-click Internal System Connection Pool.

   In the Connection Pool dialog General tab, the field Data source name should point to the data warehouse.
6. Click OK and exit the Oracle BI Administration Tool.
7. Restart the Oracle BI Server.

B.1.3 Adding Custom Translations to the W_LOCALIZED_STRING_G Table

When you add custom objects to the metadata and choose to externalize these objects (by right-clicking the object and checking the Externalize Display Name option), the Oracle BI Server looks for the translations (including those for the native language) in the W_LOCALIZED_STRING_G table.

If you do not externalize the display names, you do not need to perform the following procedures.

> **Note:** The custom Presentation layer objects show up only in the native language of the metadata (the language in which you added these new objects).

B.1.3.1 Adding String Translations for Analytics Metadata

The following procedure describes how to add string translations for Oracle Business Intelligence metadata to the W_LOCALIZED_STRING_G table. This task occurs in any database administration tool, and in the Oracle BI Administration Tool.

To add string translations for Analytics metadata

1. Open a database administration tool and connect to your data warehouse database.
2. Query for the table named W_LOCALIZED_STRING_G and add a new record to the table, as defined below in steps 4 to 8.
3. Obtain the Message Key from the Oracle BI Administration Tool as follows:
   - In the Oracle BI Administration Tool, right-click on the new Presentation layer metadata object and select Properties from the menu.
   - The Message key is displayed in the dialog under Custom Display Name. The Message key is the part that starts with CN_.

     For example, double-click the Pipeline catalog folder in the Presentation layer. The Custom Display name is Valueof(NQ_SESSION.CN_Pipeline). CN_ Pipeline is the Message Key.
4. Enter your deployment language in the new record.
5. Enter the Message Type required (for example, Metadata, FINS_Metadata).
6. Select the Message Level *AnalyticsNew*, then do the following:

- In the Message Text column, add the translation of the object.
- Check the flags (set to Yes) for the Translate and Active columns.
- Set the Error Message # column to 0.

7. Enter the required Message Facility (for example, HMF, FIN).
8. Repeat Step 3 through Step 7 for each new metadata object string.
9. Exit the database administration tool, then restart the Oracle BI Server.

B.2 About Translating Presentation Services Strings

The translations for such Presentation Services objects as report and page names are stored in the xxxCaptions.xml files available in OracleBIData\web\res\<language abbreviation>\Captions directories. In multiple language deployment mode, if you add any additional Presentation Services objects, such as reports and new dashboard pages, you also need to add the appropriate translations. Add these translations using the Catalog Manager tool. For more information on using this utility, see *Oracle Business Intelligence Presentation Services Administration Guide*.

B.3 Changing the Default Currency in Analytics Applications

In Oracle Business Intelligence Applications, you may see a dollar sign used as the default symbol when amounts of money are displayed. In order to change this behavior, you must edit the currencies.xml file using the following procedure. The currencies.xml file is located in the following directories:

- Windows:

  ```
  $INSTALLDIR\OracleBI\Web\config\
  ```

- UNIX:

  ```
  $INSTALLDIR/OracleBI/web/config
  ```

To change the default currency in Analytics Applications

1. In a text editor, open the currencies.xml file.
2. Look for the currency tag for the warehouse default (tag="int:wrhs"):

   ```
   <Currency tag="int:wrhs" type="international" symbol="$" format="$#" digits="2"
   displayMessage="kmsgCurrencySiebelWarehouse">
      <negative tag="minus" format="-$#" />
   </Currency>
   ```

3. Replace the symbol, format, digits and negative information in the warehouse default with the information from the currency tag you want to use as the default.

   For example, if you want the Japanese Yen to be the default, replace the contents of the warehouse default currency tag with the values from the Japanese currency tag (tag="loc:ja-JP"):

   ```
   <Currency tag="loc:ja-JP" type="local" symbol="¥" locale="ja-JP" format="$#"
   digits="0">
      <negative tag="minus" format="-$#" />
   </Currency>
   ```

   When you are finished, the default warehouse currency tag for Japanese should look like the following example:

```
<Currency tag="int:wrhs" type="international" symbol="¥" format="$#" digits="0"
displayMessage="kmsgCurrencySiebelWarehouse">
   <negative tag="minus" format="-$#" />
</Currency>
```

4. Save and close the currencies.xml file.

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 a Siebel CRM application.

> **Tip:** See *Oracle Business Intelligence Server Administration Guide* before performing any of the tasks in this section.

The integration of Oracle Business Intelligence with a Siebel operational application involves two general processes:

- Section C.1, "Importing Oracle's Siebel Industry Applications Seed Data"
- Section C.2, "Completing the Initialization in the Siebel Operational Application"

C.1 Importing Oracle's Siebel Industry Applications Seed Data

Oracle Business Intelligence seed data is not installed with Oracle's Siebel Industry Applications. You must import the seed data into your database after the Oracle Business Intelligence installation is completed. You only need to perform this procedure if you are using Oracle's Siebel Industry Applications.

To import Oracle Business Intelligence seed data into a transactional database

1. Obtain the required language.inp and .dat files from the Oracle Business Intelligence language folder `\OracleBI\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 folder OracleBI\server\bin.
3. From the command prompt in \OracleBI\SeedData\bin, run the following command:

   ```
   dataimp /u $USERNAME /p $PASSWORD /c "$ODBCDatasource" /d $Tableowner /f
   analytics_seed_<XXX>.dat /i metadata_upgrade_<XXX>_<DBPlatform>.inp /w y
   ```

   Replace the *XXX* with the three-letter code (FRA, ITA) and the *DBPlatform* with the abbreviation for the database platform being used. For example:

   ```
   dataimp /u sadmin /p sadmin /c JPN_CRMDEV1 /d siebel /f analytics_seed_JPN.dat
   /i metadata_upgrade_JPN_db2.inp /w y
   ```

For information about merging content into Oracle Business Intelligence Presentation Services, see the topics about using the Catalog Manager in *Oracle Business Intelligence Presentation Services Administration Guide*.

C.2 Completing the Initialization in the Siebel Operational Application

Once you have configured your Oracle BI Server and are able to access the dashboards, you need to update the Siebel operational application to view Analytics dashboards from within the Siebel operational application. Completing the initialization in the Siebel operational application involves the following processes:

- Changing the operational application host name to the host name of the machine that runs Oracle Business Intelligence Presentation Services. See the following topics:
  - Section C.2.1, "Updating the Siebel Operational Application"
  - Section C.2.2, "Reapplying Customized Style Sheets"
  - Section C.2.3, "How to Configure Oracle Business Intelligence with Oracle's Siebel Web Extension (SWE)"
  - Section C.2.4, "Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services"
  - Section C.2.5, "Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE"
  - Section C.2.6, "How to Configure Oracle BI Action Links"
  - Section C.2.7, "Testing Action Links"
- Customizing the operational application home page, dashboards, or content, or add views to a dashboard. See the following topics:
  - Section C.2.8, "Accessing Optional Analytics Applications"
  - Section C.2.9, "Process of Customizing Oracle BI Application Home Page and Dashboards"
    - Section C.2.9.1, "Customizing Oracle BI Content on the Siebel Operational Application Home Page"
    - Section C.2.9.2, "Determining the Oracle BI Report Path Argument"
    - Section C.2.9.3, "Adding Views for Custom Oracle BI Interactive Dashboards"
    - Section C.2.9.4, "Verifying NQHOST Setup for Pharma Disconnected Analytics Client"
    - Section C.2.9.5, "Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client"
    - Section C.2.9.6, "Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client"

C.2.1 Updating the Siebel Operational Application

The following task changes the Siebel operational application host name to the host name of the machine that runs Oracle Business Intelligence Presentation Services.

To update the Siebel operational application

1. Open your Siebel operational application and login as SADMIN.
2. Navigate to View, then Site Map.
3. Click Integration Administration screen.
4. Click Host Administration view.

5. Query for NQHOST in the Virtual Name column:

   – In Windows, change the host name from <*AnalyticsServerName*> to the host name of the machine that runs Oracle BI Presentation Services.

   – In AIX or Solaris, change the host name from <*AnalyticsServerName*> to the name of the port.

     For example, `servername.siebel.com:8080`

     > **Note:** For UNIX platforms only: You may need to add the domain name suffix to the server name in order to make sure that action links work on the Oracle Business Intelligence user interface.

6. Log out of the application and log back in.

C.2.2 Reapplying Customized Style Sheets

For Oracle's Siebel Business Analytics versions 7.7 and later, new styles have been appended to the following style sheets:

- Go.css
- PortalBanner.css
- PortalContent.css
- Views.css

These new classes are identified in the style sheets. In this version of Oracle Business Intelligence, new styles and files must be added to the underlying style sheet (for example, to s_Siebel7). For complete functionality, any custom styles require similar updating. In addition, views2.css and some other files have been added to the s_ directory. For custom column formatting to work properly, references to font sizes and families should also be removed from the TD Styles section in PortalBanner.css, PortalContent.css, Views.css.

Because of these changes to styles, review your deployment's customizations manually, reapply them, and test them thoroughly to ensure that there are no problems.

After you have tested them, but before you copy the default views.css files back to the implementation server, perform the following task on the server to clear the server caches.

To clear the server caches and restore your default views

1. Shut down Oracle Business Intelligence Server, Oracle BI Presentation Service and IIS.

2. Remove your custom views.css from the directory where it has been installed.

   For example:

   ```
   $INSTALL\web\app\res\s_Siebel7\b_mozilla_4
   ```

   or

   ```
   $INSTALL\OracleBIData\Web.
   ```

3. Clear the Oracle BI Presentation Services Server Cache.

   In the `C:\WINNT\Temp` directory, delete the nQs_*.temp files.

4. Clear the Browser Cache.

   From the Internet Explorer menu, navigate to Tools, then Internet Options, then Settings, then View Files, and delete all the files in these folder.

5. Restore the default views.css files to the appropriate folder.

6. Restart the Analytics Server, Oracle BI Presentation Services and IIS.

C.2.3 How to Configure Oracle Business Intelligence with Oracle's Siebel Web Extension (SWE)

Whenever you run Oracle Business Intelligence and Siebel Web Extension (SWE) on separate machines, you must perform additional configuration steps in order for action links and interactive charts to work. If, for example, you plan to run the SWE and Oracle Business Intelligence Presentation Services on different Web servers, you must use some kind of networking or load balancing mechanism to create a single logical domain (or virtual IP address) for the two machines.

Caution: When Oracle BI Applications Server and Oracle BI Presentation Services are installed on different machines and load balancing software is used to create a single virtual IP address, the action links feature works only if you log onto the Siebel application using a virtual Web browser. Action links fail on a dedicated client if the SWE and Oracle BI Presentation Services servers are on separate machines. Either use a virtual Web browser, or install the Oracle BI Applications Web Server and Oracle BI Presentation Services on the same machine.

When one virtual IP address is created for two machines, the Web browser accesses one IP address and is still routed to different physical machines, based on the port accessed. From the browser, it appears that both servers are running on the same IP address.

You can use any of several physical methods to create a single logical domain, such as running SWE and Oracle Business Intelligence Presentation Services on a single machine if you are not load balancing the SWE, or using a router to do the mapping, or using load balancing software. Your company must determine the best mechanism to accomplish this routing given the topology being used.

Configuring Oracle Business Intelligence to work with Siebel Web Extension includes the following tasks:

- Section C.2.4, "Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services"
- Section C.2.5, "Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE"

C.2.4 Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services

You create a virtual IP address for the Siebel Web Engine (SWE) and Oracle BI Presentation Services in order to make it appear that all servers are running on the same virtual machine. The easiest way to do this is to configure Oracle BI Presentation Services to run on a different port (for example, port 84) from SWE (which usually runs on port 80).

For example, SWE is load-balanced across <*machine1*>:port 80 and <*machine2*>: port 80, and Oracle BI Presentation Services is running on <*machine3*>:port 84, and the virtual address is defined as `http://siebel.company.com`. Therefore, the network or load-balancing software should be configured to route requests like http://siebel.company.com to <*machine1*> and <*machine2*>, and to route requests like `http://siebel.company.com:84 to <machine3>`.

To create a virtual IP address for Siebel Web Engine and Oracle BI Presentation Services

1. On the network, set up CSS to direct requests from <*virtual domain*> to <*physical Siebel Web Server Extension machine*>*:*
   - Where the acronym CSS represents the load-balancer or router used to do the virtual IP configuration.
   - Where <*virtual domain*> is the virtual IP prefix that users enter to navigate to the Siebel applications (in the preceding example, this is `http://siebel.company.com`).
2. On the network, set up CSS to direct requests from <*virtual domain*>:84 to <*physical Oracle BI Presentation Services machine*>:84.
3. In the Siebel application, using the Siebel Administration screen, set the NQHost parameters for Oracle Business Intelligence Symbolic URLs to point to the <*virtual domain*>:84, instead of directly to the Oracle Business Intelligence Presentation Services server physical machine.

   > **Note:** If you are running Oracle's Siebel Business Analytics 7.5.3 instead of version 7.7 or later, perform the following additional step.

4. In the Siebel application on the Oracle Business Intelligence Presentation Services server machine, locate the registry setting `\SOFTWARE\Siebel Systems, Inc.\Siebel Analytics\Web\7.5\Charts`.
5. Add a new key, ForceFileBasedPainter, and enter TRUE into the Data string.

C.2.5 Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE

Use a client browser to verify that Oracle Business Intelligence and SWE work when accessed directly through a physical machine address, using the following procedure. For <*virtualdomain*>, substitute the Virtual IP address you created in Section C.2.4, "Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services".

To test the Virtual IP configuration with Oracle Business Intelligence and SWE

1. In a client browser, type `<virtual domain>:84/analytics`. The Oracle Business Intelligence logon appears.
2. In a client browser, type `<virtual domain>/callcenter` (or other Siebel application). The SWE appears.
3. Navigate to an Analytics screen within the Siebel application to see if Oracle Business Intelligence appears.
4. Interact with Oracle Business Intelligence charts and action links.

C.2.6 How to Configure Oracle BI Action Links

Oracle BI Applications supports action links in reports and dashboards to allow users to navigate from a record in an Oracle Business Intelligence dashboard to the same record in a Siebel operational applications.

For example, you can drill down directly from an Order Management Business Intelligence dashboard to the same record in a Sales view. The link is based on the row-ID column contained in a report. You can also create new action links for any existing report. For how to create Interactive Oracle Dashboards action links, see *Oracle Business Intelligence 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:

- Section C.2.7, "Testing Action Links"

C.2.7 Testing Action Links

This task is part of the Section C.2.6, "How to Configure Oracle BI Action Links".

Use the following procedure to make sure that action links you create work correctly.

To test an Oracle Business Intelligence action link

1. Log on using your system.

   Go to Service Analytics screen, then Employee Analytics, then To Do List.

2. Click the action link on the pie chart in Overdue Activities.

   You should be in Overdue Activities Detail in Analytics.

3. Click any ROW_ID action link.

   You should be in the Employee screen in Oracle's Siebel Call Center operational application.

C.2.8 Accessing Optional Analytics Applications

Depending on the options you purchased with your Siebel operational application, you must perform additional steps in order to access the corresponding Oracle Business Intelligence options. Table C–1 shows the additional options for Oracle Business Intelligence.

Table C–1 Oracle Business Intelligence Applications Options

Siebel Application	Options
Sales	Sales Analytics
Service	Service Analytics, Contact Center Telephony Analytics
Marketing	Marketing Analytics

By default, the dashboards and reports contained in these optional areas are hidden. If, for example, you purchased Sales Analytics with your Sales application, you must perform the additional steps shown in the following procedure to access the Sales Analytics.

To turn on options for Sales Analytics

1. Log in to Oracle Business Intelligence as Administrator.
2. Navigate to Answers, then Oracle BI Presentation Services Administration and select the option to manage Presentation Services groups and users.
3. Locate the Web Group corresponding to your option.

   The Web Group options are shown in the following list.

   - No Forecasting
   - No Forecasting Lite
   - No Universal Queuing
   - No Email Response
   - No Service Agreements
   - No Partner Marketing
   - No Partner ERM
   - No Partner ISS
4. Click on the Edit icon.
5. Under the Group Membership section, click the delete icon (X) to delete Analytics Users from this group.
6. Click Finished and log out of the application.
7. Log in again to access the additional optional dashboards and reports.

C.2.9 Process of Customizing Oracle BI Application Home Page and Dashboards

The process of customizing your Oracle Business Intelligence application's home page and dashboards may include the following tasks:

- Section C.2.9.1, "Customizing Oracle BI Content on the Siebel Operational Application Home Page"
- Section C.2.9.2, "Determining the Oracle BI Report Path Argument"
- Section C.2.9.3, "Adding Views for Custom Oracle BI Interactive Dashboards"
- Section C.2.9.4, "Verifying NQHOST Setup for Pharma Disconnected Analytics Client"
- Section C.2.9.5, "Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client"
- Section C.2.9.6, "Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client"

C.2.9.1 Customizing Oracle BI Content on the Siebel Operational Application Home Page

Oracle Business Intelligence applications are integrated with Siebel operational applications using the symbolic URL infrastructure. The following task describes how to use symbolic URLs to link a new report to a Siebel operational application home page and how to add new Analytics Dashboards to the Siebel operational application. The symbolic URL specifies how the HTTP request to the external application should be constructed and to defines any arguments and values to be sent as part of the request.

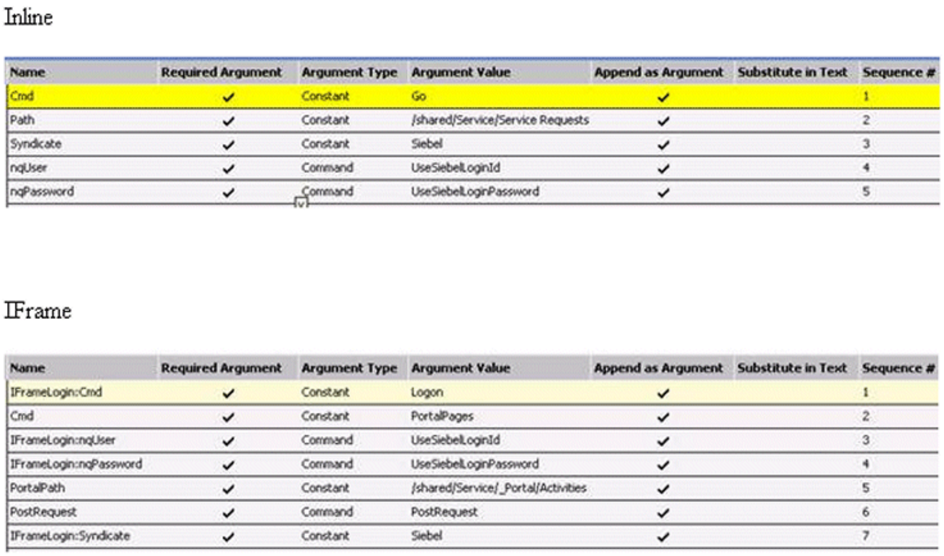
> **Note:** For more information on the symbolic URL infrastructure, see Siebel Portal Framework Guide. This topic assumes that you have successfully built a symbolic URL to link some external content, using the instructions given in Siebel Portal Framework Guide.

For each Analytical report on a Siebel operational application home page, there is a symbolic URL record defined that links the home page to the Analytics report. If you have not already done so, you need to create a new home page and set it up to use a symbolic URL.

> **Note:** For Oracle Business Intelligence Disconnected settings, see Section C.2.9.4, "Verifying NQHOST Setup for Pharma Disconnected Analytics Client".

The figure below shows example Inline and IFrame symbolic URL arguments.

Figure C–1 Examples of Symbolic URL Arguments.



Inline

Name	Required Argument	Argument Type	Argument Value	Append as Argument	Substitute in Text	Sequence #
Cmd	✓	Constant	Go	✓		1
Path	✓	Constant	/shared/Service/Service Requests	✓		2
Syndicate	✓	Constant	Siebel	✓		3
nqUser	✓	Command	UseSiebelLoginId	✓		4
nqPassword	✓	Command	UseSiebelLoginPassword	✓		5

IFrame

Name	Required Argument	Argument Type	Argument Value	Append as Argument	Substitute in Text	Sequence #
IFrameLogin:Cmd	✓	Constant	Logon	✓		1
Cmd	✓	Constant	PortalPages	✓		2
IFrameLogin:nqUser	✓	Command	UseSiebelLoginId	✓		3
IFrameLogin:nqPassword	✓	Command	UseSiebelLoginPassword	✓		4
PortalPath	✓	Constant	/shared/Service/_Portal/Activities	✓		5
PostRequest	✓	Command	PostRequest	✓		6
IFrameLogin:Syndicate	✓	Constant	Siebel	✓		7

To configure a new Home Page to use a symbolic URL

1. Make sure the symbolic URL has been set up.
2. Launch the Siebel operational application and navigate to the Integration Administration, then Symbolic URL Administration view.
3. Query for the symbolic URL that has been set up for Analytics.

   The name of this symbolic URL should be exactly the same as the calculated value of the field that was added to the Business Component. For example, you may have a symbolic URL named HomePageAnalytics.
4. In the URL field, enter the Web URL. For example:

   ```
   http://NQHOST/Analytics/saw.dll
   ```

   For the Host Name, choose the Analytics Server Name from the drop-down list. The following table shows the other parameters for reports.

Parameter	Value
NQHOST	A virtual name in the URL that points to the Oracle BI Presentation Services machine
Fixup Name	Inside Applet
SSO Disposition	Inline
Oracle BI Presentation Services application name	Select from the drop-down list

5. Create the appropriate symbolic URL Arguments.

   These arguments depend upon the particular Analytics report that you are trying to display. The Argument Values should be the same for any Analytics report you work with, except for the Path Argument Value. Use the procedure in Section C.2.9.2, "Determining the Oracle BI Report Path Argument" to determine the path to the Analytics report.

C.2.9.2 Determining the Oracle BI Report Path Argument

The Path argument tells the symbolic URL the path to the report on the Oracle BI Presentation Services. (For example, `/shared/Sales/Pipeline/Overview/Top 10 Deals`.) Use the following procedure to determine the path to the Analytics report.

To determine the path to the report

1. Log on to your Oracle BI Presentation Services as an Administrator.
2. In the Siebel operational application, navigate to Answers, then Oracle BI Presentation Services Administration.
3. Select Manage Analytics Catalog, and then navigate to your report.
4. Add this path name to the Symbolic URL argument.

   The following table shows the symbolic URL path arguments for reports.

Name	Type	Path Argument Value	Append	Sequence #
Cmd	Constant	GO	Y	1
Path	Constant	/shared/Sales/Pipeline/Overview/Top 10 Deals	Y	2
Syndicate	Constant	Siebel	Y	3
nQUser	Command	UseSiebelLoginId	Y	4
nQPassword	Command	UseSiebelLoginPassword	Y	5
PostRequest	Command	PostRequest	Y	6

C.2.9.3 Adding Views for Custom Oracle BI Interactive Dashboards

This task is similar to that of adding Oracle Business Intelligence reports to the home page:

- Using Oracle's Siebel Tools, set up a new view.
- In the Siebel operational application, define a symbolic URL for that view.

For more information on how to set up a view to use Symbolic URLs, see Siebel Portal Framework Guide.

To configure the symbolic URL for Analytics dashboards

1. Define a Symbolic URL.
   a. Navigate to Site Map, then Integration Administration, then Symbolic URL Administration.
   b. In the Symbolic URL Administration view, add a new record.
2. Define Symbolic URL arguments.
   a. Navigate to Symbolic URL Administration.
   b. In the Symbolic URL Administration list, select the Symbolic URL you want to configure.

C.2.9.4 Verifying NQHOST Setup for Pharma Disconnected Analytics Client

Pharma Disconnected Analytics is a prebuilt Disconnected Analytics application for Oracle's Siebel Pharma Sales. When you run Oracle Business Intelligence Disconnected Client, your client application runs against a local instance of the database on your machine. You need to confirm that NQHOST is using the correct port number and DLL files to work with the Pharma Disconnected Analytics Client.

To verify the NQHOST setup

1. Run the Disconnected Client application against your local database.
2. Log in as Administrator.

   You must have access to the following Integration Administration Screen views:

   – WI Host Admin View

   – WI Symbolic URL Admin View
3. Navigate to Administration – Integration.
4. Click WI Symbolic URL List, and search for NQHOST in Virtual Name field.

   The Host Name value should be localhost:9762.
5. Select the Host Administration View from the drop-down list and choose Symbolic URL Administration.
6. The following table shows Disconnected Client symbolic URLs to search for.

   For each of the URLs, the URL field is:

   ```
   http://NQHOST/Analytics/sawd.dll
   ```

   The Disconnected Client Symbolic URL Names are:

   – ePharmaHomePageAnalytics1

   – ePharmaHomePageAnalytics2

   – ePharmaHomePageAnalytics3

   – ePharmaHomePageAnalytics4

   – SISMLSDistrictManagerDashboard1

   – SISMLSSalesRepDashboard1

   – SiebelAnswers

   – SiebelDelivers

The following table shows the symbolic URL path arguments for Oracle BI Answers and Oracle BI Delivers.

Name	Type	Path Argument Value	Append	Sequence #
Cmd	Constant	Answers for SiebelAnswers Delivers for SiebelDelivers	Y	1
nQUser	Command	UseSiebelLoginId	Y	2
nQPassword	Command	UseSiebelLoginPassword	Y	3

C.2.9.5 Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client

If you have licensed both the Siebel Pharma operational application and Oracle Business Intelligence Disconnected Client, you must change part of the operational application configuration file in order to access Analytics reports on the LS Analytics Home Page and to allow Create Target List integration.

To allow integration of Pharma Analytics Home Page with Pharma Disconnected Analytics

1. On the machine where you have installed the Siebel operational application, navigate to `$INSTALLDIR\sea77\siebsrvr\BIN\ENU\epharma.cfg`.
2. Using a text editor, open the file epharma.cfg.
3. In the Local section, find the parameter UseCachedExternalContent.

   The default value for this parameter is TRUE.

   > **Note:** If this parameter remains set to TRUE, Analytics reports on the Home Page return an error.

4. Set the UseCachedExternalContent parameter to FALSE.
5. Save and close the file.

C.2.9.6 Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client

If you have licensed both the Siebel Pharma operational application and Oracle Business Intelligence Disconnected Client, you must change part of the operational application configuration file in order to allow you to import the contacts from the Pharma Analytics Create Target List function into the Disconnected Analytics Client.

To allow importation of Target List contacts into Oracle Business Intelligence Disconnected Client

1. On the machine where you have installed the Siebel operational application, navigate to `$INSTALLDIR\sea77\siebsrvr\BIN\ENU\epharma.cfg`.
2. Using a text editor, open the file epharma.cfg.
3. In the Siebel Client section, find the parameter EnableFQDN.

   The default value for this parameter is TRUE.

> **Note:** If this parameter remains set to TRUE, the Create Target List functionality is disabled in the Pharma Disconnected Analytics Client.

4. Set the EnableFQDN parameter to FALSE.
5. Save and close the file.
    a. In the Symbolic URL Arguments list, enter the arguments that need to be sent to the external host.

       The following table shows the symbolic URL argument definitions for dashboards.

Parameter	Value
Fixup Name	Default
SSO Disposition	IFrame
Oracle BI Presentation Services application name	Select from the drop-down list.

The following table shows the symbolic URL path arguments for dashboards.

Name	Type	Value	Append	Sequence #
Cmd	Constant	Dashboard	Y	1
PortalPath	Constant	/	Y	2
Page	Constant	Sales	Y	3
Syndicate	Constant	Siebel	Y	4
nQPassword	Command	UseSiebelLoginPassword	Y	5
nQUser	Command	UseSiebelLoginId	Y	6

6. In Oracle Business Intelligence, create the dashboards.

> **Note:** Only an Oracle BI Presentation Services administrator can perform this step.

7. In Oracle Business Intelligence, set up the dashboards to your requirements.

C.3 Viewing Information About Preconfigured Dashboards

If your organization has prebuilt applications installed, you can use Catalog Manager to locate and view information about preconfigured dashboards. You may first have to expose the dashboards and requests.

C.3.1 Exposing Dashboards and Requests

Depending on the Oracle Business Intelligence options your organization purchased, you may need to expose these options before the associated dashboards and requests can be viewed in Oracle BI Presentation Services and in Catalog Manager. This applies to sites that have the following prebuilt applications:

- Sales Analytics

- Service Analytics
- Contact Center Analytics
- Marketing Analytics
- Partner Analytics

C.3.2 Locating Dashboards, Page Names and Reports

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 > page name. The path to locate reports is Presentation Catalog > shared folder > prebuilt application name > Folder name > Report name.

D

Configuring Metadata for Oracle Business Intelligence Applications

This appendix describes configuration necessary for the Oracle Business Intelligence metadata for Siebel CRM sources. This configuration includes administrative tasks for metadata setup.

D.1 Metadata Setup Steps for Siebel CRM Sources

This section includes metadata setup steps you may need to perform if you are using Siebel as a source.

This section includes the following topics:

- Section D.1.1, "Updating Oracle Financial Services Analytics Logical Table Sources"
- Section D.1.2, "Developing and Deploying Predictive Scores"
- Section D.1.3, "Business Intelligence Metadata Requirements for Oracle's Siebel Industry Applications"

D.1.1 Updating Oracle Financial Services Analytics Logical Table Sources

In the Oracle BI repository file, the FACT - CRM - Asset logical table has the following logical table sources active:

- W_ASSET_F
- W_ASSET_F_FINS

If you are using any of the Oracle BI Applcations listed below, keep the W_ASSET_F_FINS logical table source active and deactivate the W_ASSET_F logical table source. For instructions on activating and deactivating logical table sources, see the procedure below.

- Oracle Finance Sales Analytics
- Oracle Finance Service Analytics
- Oracle Finance Marketing Analytics
- Oracle Finance Institutional Analytics
- Oracle Finance Retail Analytics
- Oracle Insurance Partner Manager Analytics
- Oracle Insurance Sales Analytic

- Oracle Insurance Service Analytics
- Oracle Insurance Marketing Analytics
- Oracle Insurance Partner Manager Analytics

If you are not using any of Oracle BI Applications listed above, keep the W_ASSET_F logical table source active and deactivate the W_ASSET_F_FINS logical table source.

To activate and deactivate logical table sources

1. Shut down the Oracle BI Server.
2. Using the Oracle BI Administration Tool, open the Oracle BI Repository (OracleBIAnalyticsApps.rpd).
3. Go to the Business Model and Mapping dialog box (the logical layer dialog box) and open the Core folder.
4. Scroll down to the Fact - CRM - Asset logical table and open its Sources folder.
5. To activate a logical table source:
    a. In the list of logical table sources, right-click the logical table source you want to activate.
    b. Select Properties.
    c. Click the General tab in the Properties dialog and make sure that the Active check box is checked. If it is not, check it.
6. To deactivate a logical table source:
    a. In the list of logical table sources, right-click the logical table source you want to deactivate.
    b. Select Properties.
    c. Click the General tab in the Properties dialog and make sure that the Active check box is deselected.
7. Click OK and save the repository.
8. Restart Oracle BI Server.

D.1.2 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 CRM application.

2. Integrate the scores into the Oracle Business Analytics Warehouse.

Once this is completed, scores may be viewed in the Siebel operational application by accessing the Accounts, then Profiles, then Loyalty Profile view.

3. Load the integrated scores into the Oracle Business Analytics Warehouse during the extraction, transformation, and loading (ETL) process.
4. After the scores are loaded into the Oracle Business Analytics Warehouse, map them to the following Oracle Business Intelligence metadata fields:
   - Churn Score
   - Customer Lifetime Value Score
   - Upsell Score
   - Cross-Sell Score
   - Financial Risk Score

   In conjunction with other associated metadata, these fields are primarily used to populate the Loyalty Management dashboard.

D.1.3 Business Intelligence Metadata Requirements for Oracle's Siebel Industry Applications

Some metadata needs to be set up properly in the Oracle BI Repository for it to be displayed accurately in Oracle Business Intelligence. The following topics describe the metadata structure for each of the following Oracle's Siebel Industry Applications:

- Section D.1.3.1, "Oracle Telecom Sales Analytics, Telecom Service Analytics and Telecom Marketing Analytics"
- Section D.1.3.2, "Oracle Pharma Sales Analytics Dimensions"
- Section D.1.3.3, "Dimensions Specific to Subject Areas in Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics"

D.1.3.1 Oracle Telecom Sales Analytics, Telecom Service Analytics and Telecom Marketing Analytics

Oracle Telecom Sales Analytics, Oracle Telecom Service Analytics and Oracle Telecom Marketing Analytics make use of order management functionality configured for CME. For these Business Intelligence applications to fully reflect the information collected by CME order management functionality, some extensions to the Telecom Analytics application may be required. This topic explains these potential extensions.

Oracle's Siebel Sales Orders include complex products and simple products.

Complex Products. A series of products related by a product hierarchy. The highest product in the hierarchy is the root product, and the lower level products are the child products. In complex products, revenue figures are summed and roll up to the root product using the ROLLUP_NET_PRI field. For a complex product, Oracle Business Intelligence examines only the root product when computing revenue. Child products are disregarded because their revenue is already reflected in the root.

Simple Products. A root product. Oracle Business Intelligence examines this root product when computing revenue, and nothing more.

Oracle's Siebel Communications, Media and Energy order management functionality supports products which have recurring charges over time (for example, $20 per month for 12 months), one-time charges (for example, one-time purchase price of equipment), and usage charges (for example, 15 cents per minute).

The revenue attributed to a product with recurring charges is valued by taking the product's net price and multiplying it by the number of months that product is anticipated to be active, as represented by the Number of Revenue Occurrences field. This field, contained in Quote Item and Order Item records, is contained in the Oracle Business Analytics Warehouse by the following fields:

- W_QUOTEITEM_F.NUM_OCCURRENCE
- W_ORDERITEM_F.NUM_OCCURRENCE

In Oracle's CME family of products (Oracle Communications, Media and Energy Sales Analytics, Oracle Communications, Media and Energy Service Analytics, Oracle Communications, Media and Energy Marketing Analytics), revenue metrics do not automatically account for all recurring charges, and do not consider the NUM_OCCURRENCE fields. Instead, Oracle's CME family of products revenue metrics incorporate one-time charges, one-month's worth of recurring charges, and no usage charges. To incorporate the anticipated value of all recurring charges, the W_QUOTEITEM_F.NUM_OCCURRENCE and W_ORDERITEM_F.NUM_OCCURRENCE fields may need to be incorporated into revenue calculations made during the Extraction, Transformation and Load (ETL) process for order item and line item records.

Alternatively, these fields in the Oracle Business Analytics Warehouse, representing the aggregated recurring and one-time product charges, may be used and incorporated into the ETL processes:

- `S_ORDERITEM.PER_MTH_CHG_SUBTOT`
- `S_ORDERITEM.ONETIME_CHG_SUBTOT`
- `S_QUOTEITEM.PER_MTH_CHG_SUBTOT`
- `S_QUOTEITEM.ONETIME_CHG_SUBTOT`

Each CME Order line item and Quote line item contains an Action Type of Add, Update, or Delete. Because Oracle Business Intelligence only looks at root product line items, only the Action Types associated with the root product are considered during analysis. Therefore, while all line items for a complex product may collectively include a combination of various Action Types, only the Action Type for the root product are considered during analysis. This is of special importance if a filter or query criteria in analysis is based on the Action Type field, which it is for most Account Management and Revenue Management dashboard reports.

Similarly, each CME Order line item and Quote line item is associated with a product of a particular Price Type. Because Oracle Business Intelligence considers root products only, only the Price Type associated with the root product is considered during analysis. Again, this is important if a filter or query criteria is based on Price Type. Such filter criteria apply to most Account Management and Revenue Management dashboard reports.

D.1.3.2 Oracle Pharma Sales Analytics Dimensions

Although the following dimensions are used in all subject areas, this topic describes the configuration necessary for Pharma Analytics applications. For more information, please refer to *Siebel Life Sciences Guide Version 8.0 Appendix B: Configuring Data for Siebel Pharma Analytics*.

D.1.3.2.1 Positions Dimension A sales territory is defined in Group Administration–Positions by a Siebel position. Creating parent positions creates the sales force hierarchy. Up to 10 levels of sales force hierarchy are supported by the application. Employees should be assigned to positions to populate employee hierarchy.

Position Types need to be set up according to compensation type (Rx or sales) only at the sales territory level. A district manager does not need to have a Position Type assigned to it. Sales Allocation needs to be exposed on the list to enter script compensation percentages (Rx or Sales) associated with each territory. For example, if all sales representatives receive 100% of the Rx on a ZIP Code, no action is needed or Position Type = Sales Representative can be assigned to the position.

Seed data on the Position Type list of values has been enhanced to include types for mirror, job share, and swat. Typically, both mirror and job share represent a position that receives less than 100% of the total scripts on a ZIP Code.

D.1.3.2.2 Alignments Dimension A sales territory alignment is the relationship of ZIP Code-to-territory or brick-to-territory. The alignment relationship is created in Oracle's Siebel Assignment Manager under Assignment Administration–Territories, as shown in Table D–1.

Table D–1 Sales Territory Alignment

Relationship	Criteria	Comments
Contact ZIP to Territory	Contact ZIP Code	Use contact primary address ZIP Codes. Do not use ranges of ZIP Codes (that is, enter unique ZIP Codes as low and high values). Do not enter duplicate ZIP Codes.
Account ZIP to Territory	Account ZIP Code	Do not use ranges of ZIP Codes (that is, enter unique ZIP Codes as low and high values). Do not enter duplicate ZIP Codes.
Contact Brick to Territory	Contact Brick	Use contact primary address brick. Do not use ranges of bricks (that is, enter unique bricks as low and high values). Do not enter duplicate ZIP Codes.
Account Brick to Territory	Account Brick	Do not use ranges of bricks (that is, enter unique bricks as low and high values). Do not enter duplicate ZIP Codes.
Account to Territory	Account	Do not enter duplicate accounts.
Contact to Territory	Contact	Do not enter duplicate contacts.

D.1.3.2.3 Products Dimension The product hierarchy requires customer products (products of the company who licensed the software) to have predefined product types as shown in Table D–2.

Table D–2 Customer Products Predefined Product Types

Product Level	Product Type	Example
3	Sample	Aracid 400 MG
2	Detail	Aracid
No Level	Sub Market	COPD
1	Market	Asthma

> **Note:** Competitive products should use the product type Competitor. Competitor product hierarchies are set up using parent product relationships exclusively and should not have product levels assigned to them.

D.1.3.2.4 Product Costs Dimension

Product costs for customer products (that is, products of the company that licensed the software) require population in the Product Administration, Product Form, as shown in Table D–3.

Table D–3 Product Costs For Customer Products

Product Type	Field to be Populated
Sample	Sample Cost
Detail	Avg. Promo Cost
Promotional Item Cost	Sample Cost

D.1.3.3 Dimensions Specific to Subject Areas in Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics

This section discusses the subject areas used by Pharma Analytics. For more information, please refer to *Siebel Life Sciences Guide Version 8.0, Appendix B: Configuring Data for Siebel Pharma Analytics*.

D.1.3.3.1 Pharma Sales Effectiveness

This subject area is focused on syndicated data analytics.

The specific configuration required for the syndicated data depends on your data types, and the Analytics application and reports that you have licensed. The Data Loading Matrix table is the basis of prebuilt reports. The syndicated data loading matrix populates both base and derived metrics used in Pharmaceutical Sales Analytics.

D.1.3.3.2 Pharma Product Categories

Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics supports custom and prebuilt product category trees to allow roll-up of syndicated data by alternative hierarchies. To populate a custom category, first create a Catalog in Catalogue Administration, and create categories and subcategories as part of the catalogue. Table D–4 lists the categories that need to have the Usage Type field populated in the Catalog Admin Category Detail list.

Table D–4 Hierarchy Categories to be Populated in Pharma Analytics

Usage Type Code	Hierarchy Category
ATC	Anatomical Therapeutic Class
Chemical	Chemical
Application Form	Product application
USC	User-defined codes and custom hierarchies

D.1.3.3.3 Pharma Promotional Effectiveness

This subject area combines call activity data with syndicated data to analyze effectiveness of call activity.

Call Activity analysis records are derived from submitted call activity records stored in S_EVT_ACT in the Oracle Business Analytics Warehouse, where they are stamped with the ZIP Code or brick where the activity took place—that is, the Contact primary address's ZIP code/brick or the Account ZIP Code/brick. Allocation of these ZIP Code/brick records should be done by Assignment Manager rules to make sure that they are correctly allocated. Assignment Manager rules must match the Contact or Account primary address ZIP Codes or bricks. Otherwise, data integrity is not maintained.

Only calls that have status Submitted on the Pharma Professional Call Form are brought over from the Oracle Business Analytics Warehouse to the Oracle Business Analytics Warehouse.

D.1.3.3.4 Pharma Medical Education Effectiveness This subject area combines measures from MedEd and Syndicated Data to measure effectiveness of medical education events used on Medical Education Analytics.

Only MedEd events with the status Completed on the Pharma ME Event List are extracted from Oracle Business Analytics Warehouse to populate the Oracle Business Analytics Warehouse.

MedEd Event costs are based on costs of activities in the Pharma ME Event Activity List. Costs are allocated based on MedEd Team cost allocation, and promoted products Cost Allocation on the MedEd event.

Costs are solely based on physician invitees with the status Attended in the Pharma ME Event Professional Invitee Session List.

Control groups are based on physicians who have the same contact ranking as attendee physicians within the same sales territory at the time of the event, but who did not attend the event.

D.1.3.3.5 Pharma Objectives Achievement` This subject is used to measure achievement and results for pharma call activity and Rx/sales targets. It is based on Pharma Objectives.

Objectives need to have a Unit populated in Retail Objective Form. Actual target numbers per contact and account need to be populated in the Pharma Campaign Target Account List or the Pharma Campaign Target Professional List Toggle.

E

Using Oracle Business Analytics Warehouse Exception Reports

Note: Exception Reports are only supported with Siebel CRM sources.

The exception reports covered in this chapter identify inconsistencies with ETL source data (used for ETL processes) that can lead to erroneous results in the Oracle Business Analytics Warehouse or may cause data loss during the ETL process. These reports point out some of the known problematic areas, but they should not be relied upon to find all potential data inconsistencies in the source data.

This chapter includes the following topics:

- Section E.1, "Understanding Oracle Business Analytics Warehouse Exceptions"
- Section E.2, "Executing Oracle Business Analytics Warehouse Exception Reports"

E.1 Understanding Oracle Business Analytics Warehouse Exceptions

Exception reports are defined for the following components:

- **List of Values (LOV).** Identifies gaps and overlaps for certain LOV types.
- **Cost Lists.** Identifies products for which the cost lists have not been defined, or where the cost lists for a specified product and currency have overlapping time periods.
- **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.

E.1.1 List of Values Exceptions

List of Values include High and Low values that can be used as bucket values in categories for effective analysis. If these values are not contiguous (such as gaps or overlaps in defined values), the ETL process cannot accurately categorize the values.

An example of List of Values exceptions is shown in Table E–1.

Table E–1 List of Values Exceptions (Example)

Type	Name	Low	High
MY_TYPE	< 100	10	100
MY_TYPE	80 – 200	80	200
MY_TYPE	250 – 300	250	300
MY_TYPE	350 – 400	350	400

Notice that, in the example for Type MY_TYPE, there are overlaps and gaps between records. There is an overlap of ranges between the first and second row. There is a gap between second and third row, and between third and fourth rows.

The following LOV types are analyzed for List of Values 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

E.1.2 Cost List Exceptions

Cost Lists for specified products and currency should not have overlapping time periods. If multiple cost lists are defined for a product and currency during a given time period, then the cost for the product may not be computed correctly in the Oracle Business Analytics Warehouse.

An example of Cost List exceptions is shown in Table E–2.

Table E–2 Cost List Exceptions (Example)

Cost List	Product Name	Currency	Start Date (MM-DD-YYYY)	End Date (MM-DD-YYYY)	Cost
Cost List 1	Product 1	USD	01-01-2000	12-31-2000	10.00
Cost List 2	Product 1	USD	06-01-2000	06-01-2001	12.00
Cost List 3	Product 1	USD	06-01-2001	06-01-2002	13.00

In the example, Cost List 1 and 2 have definitions of cost overlapping over 06-01-2000 to 12-31-2000.

E.1.3 Products Without a Cost List

During the ETL process, the costs of the products are calculated based on the Cost List table. If the cost lists are not defined correctly, the cost of the products cannot be calculated correctly in the Oracle Business Analytics Warehouse. This exception mapping queries the product table and looks for a minimum of one cost list to be defined. The products with no cost list definition are flagged as exceptions.

E.1.4 Exchange Rate Exceptions

The Oracle Business Analytics Warehouse supports transactions in many different currencies. Oracle Business Intelligence converts all currencies in the Oracle Business Analytics Warehouse to a single currency for analysis purposes. The ETL Base Exchange Currency parameter in System Preferences, indicates the currency to which all the financial amounts will be converted. The Exchange rates are derived from the Exchange Rate tables in the Oracle Business Analytics Warehouse. If the currency exchange rates do not change for a period of 30 days, then Oracle Business Intelligence flags it as an exception.

If there are time period gaps in the exchange rate data, the ETL process defaults to the most recent recorded exchange rate. If the actual exchange rate is significantly more or less favorable than what is recorded in the database, the outdated exchange rate distorts the true value of currency amounts in the Oracle Business Analytics Warehouse.

Note: Exchange rates are derived from records that are of type 'Daily' in the Oracle Business Analytics Warehouse. If any other types have been defined, they are not handled without some customization.

E.1.5 Invalid Hierarchy Exceptions

Accounts, divisions, products, and opportunities can all have hierarchical relationships. These entities are denormalized within the Oracle Business Analytics Warehouse database to a fixed number of levels. Oracle Business Intelligence supports up to ten hierarchies levels in the Oracle Business Analytics Warehouse. If the depth of hierarchies extends beyond this number, results become inconsistent or incomplete.

E.1.6 Circular Hierarchy Exceptions

Circular Hierarchies arise when the parent-child relationship has circular references. See Table E–3 for an example.

Table E–3 Circular Hierarchy Example 1

Child	Parent
A1	A2
A2	A1

Oracle Business Intelligence flags exceptions for two levels. Circular references over two hierarchies are not flagged. See Table E–4 for an example.

Table E–4 Circular Hierarchy Example 2

Child	Parent
A1	A2

Table E–4 (Cont.) Circular Hierarchy Example 2

Child	Parent
A2	A3
A3	A1

These produce infinite levels of hierarchies. The same records will be captured under the Invalid Hierarchy exceptions as their hierarchy depths will increase beyond 10 levels.

E.2 Executing Oracle Business Analytics Warehouse Exception Reports

Before loading the Oracle Business Analytics Warehouse for the first time and for the subsequent refreshes, you should plan to spend time cleansing your transactional database data using the exception reports. The process is iterative, and requires coordination with other team members who have responsibility for data in the transactional database, such as the Siebel database administrator. After the initial cleansing, you should generate the exception reports on a scheduled basis to maintain the integrity of your data.

> **Note:** Rerunning the exception reports overwrites all data contained in this view.

To cleanse the Oracle Business Analytics Warehouse, repeat these actions until the exception report is empty:

- In DAC, run the Exception Reports execution plan.
- In Oracle's Siebel application that you are using, navigate to Analytics Administration, then Exception Reports.

> **Note:** In Siebel Financial Services, this screen is called DataMart Administration.

- For every line in the ETL Exception Reports list, fix the cause of the problem. For information on fixing problems, see Section E.2.1, "Cleansing Data".

E.2.1 Cleansing Data

Use a combination of the Exception Reports and the Diagnostic views to assess changes that need to be made external to the Oracle Business Analytics Warehouse, and changes to the Oracle Business Analytics Warehouse directly.

The ETL Exception Reports list contains one record for each exception flagged in the Oracle Business Analytics Warehouse. The ETL Exception Explanation form, located below the ETL Exception Reports list, describes the selected exception, its effect on the Oracle Business Analytics Warehouse building process, and offers suggestions for repairing the data.

To fix an exception

1. Select an exception record.
2. Read and understand the text in the ETL Exception Explanation form.

3. Click the report link.

   The object's data appears. (For example, if the object is an account, then the Account form appears. If the object is a cost list, then the Cost List list appears.)

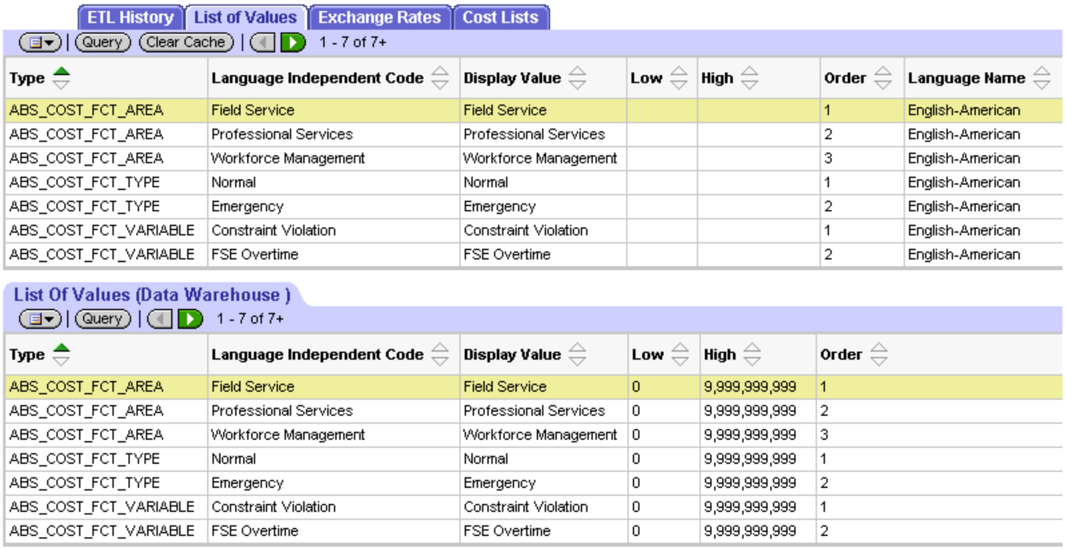
4. Repair the problem, using the text in the ETL Exception Explanation form as a guide.

5. Return to the ETL Exception Reports list and place a check mark in the Fixed column to indicate to others that this exception has now been fixed.

E.2.2 Using the List of Values View

Use the List of Values view, shown in the figure below, to visually compare how the list of values data extracted from the Oracle Business Analytics Warehouse coordinates with the values loaded into the Oracle Business Analytics Warehouse. The ETL process removes duplicates and overlaps and fills data gaps. Values are extended to span the List of Values (LOV) minimum and maximum values. Duplicates, Range Gaps, and Overlaps are flagged by the exception reports.

Figure E–1 List of Values View



Type	Language Independent Code	Display Value	Low	High	Order	Language Name
ABS_COST_FCT_AREA	Field Service	Field Service			1	English-American
ABS_COST_FCT_AREA	Professional Services	Professional Services			2	English-American
ABS_COST_FCT_AREA	Workforce Management	Workforce Management			3	English-American
ABS_COST_FCT_TYPE	Normal	Normal			1	English-American
ABS_COST_FCT_TYPE	Emergency	Emergency			2	English-American
ABS_COST_FCT_VARIABLE	Constraint Violation	Constraint Violation			1	English-American
ABS_COST_FCT_VARIABLE	FSE Overtime	FSE Overtime			2	English-American

List Of Values (Data Warehouse)

Type	Language Independent Code	Display Value	Low	High	Order
ABS_COST_FCT_AREA	Field Service	Field Service	0	9,999,999,999	1
ABS_COST_FCT_AREA	Professional Services	Professional Services	0	9,999,999,999	2
ABS_COST_FCT_AREA	Workforce Management	Workforce Management	0	9,999,999,999	3
ABS_COST_FCT_TYPE	Normal	Normal	0	9,999,999,999	1
ABS_COST_FCT_TYPE	Emergency	Emergency	0	9,999,999,999	2
ABS_COST_FCT_VARIABLE	Constraint Violation	Constraint Violation	0	9,999,999,999	1
ABS_COST_FCT_VARIABLE	FSE Overtime	FSE Overtime	0	9,999,999,999	2

The top List of Values list shows values from the Oracle Business Analytics Warehouse and the bottom List of Values (Data Warehouse) list shows the data that is to be used in ETL process. You can edit the Oracle Business Analytics Warehouse data directly in this view, but the Oracle Business Analytics Warehouse list is read-only.

> **Note:** The List of Values is extracted into the Oracle Business Analytics Warehouse where the language is the same as the ETL Default Language set in the DAC Source System Parameters, or whose translate flag is set to 'N,' or those that are active. For more information about setting DAC Source System Parameters, see.

E.2.3 Using the Exchange Rates View

Use the Exchange Rates view to diagnose currency translation issues in the Oracle Business Analytics Warehouse. The ETL process removes duplicates, fills gaps, and removes overlaps. The ETL process computes exchange rates based on commutative and associative properties, such as product and reverse rates.

The top Exchange Rates list shows currencies, the middle Exchange Rates list shows the Oracle Business Analytics Warehouse values for active currencies and their exchange rates, and the bottom Exchange Rates (Data Warehouse) list shows the values loaded into the Oracle Business Analytics Warehouse for the selected currency in the upper Exchange Rates list to the ETL Base Exchange Currency. The Exchange Rates (Data Warehouse) list is read-only.

- The Active Currencies predefined query restricts the list to the active currencies in the Oracle Business Analytics Warehouse.
- The exception reports flag any exchange rates to the ETL Base Exchange Currency that have not been defined within a specified period (30 days) in the DAC Source System Parameters.

E.2.4 Using the Cost List View

Use the Cost List view to display the cost lists from the Oracle Business Analytics Warehouse from the point of view of the product, and a read-only view of the values to be loaded into the Oracle Business Analytics Warehouse. The ETL process removes duplicates, overlaps, and fills gaps.

The Cost List list (top) shows products, and the Cost List Line Items list (middle) shows the cost lists associated with the selected product. The Cost Lists (Data Warehouse) list (bottom) shows the data as it is transformed for the Oracle Business Analytics Warehouse.

- The exception reports flag products that do not appear in the Cost List list or have Cost List time gaps and overlaps.
- The Oracle Business Analytics Warehouse contains only one Cost List for a product and a currency at a time.

E.2.5 Using the ETL History View

After all of the exceptions are corrected, the building of the data warehouse can be initiated. This view lists the history of the ETL processes and their statuses. When each ETL batch starts, the name of the process along with the timestamp is set, the status is set to STARTED. When the batch completes, its status is updated to COMPLETED.

E.2.6 Additional Exceptions

Additional exceptions include:

- The Analysis start and end date in the DAC Source System Parameters must span the entire period of time during which the transactions have occurred. For example, you may want to choose an early and late date range to cover the entire time period you are analyzing. These dates in the DAC Source System Parameters are crucial for the building of Day Dimension, flattening of Exchange Rates, Cost Lists, and KPI (Key Performance Indicator fact) calculations.
- The DAC Source System Parameters — ETL Date Format, ETL Analysis Start, ETL Analysis End parameters, and the List of Values — ETL_UNSPEC_DATE must be defined in the same data format. If one is changed, the others must be changed accordingly.

- List of Values must be defined appropriately. If there is no appropriate entry in List of Values, the strings that depend on List of Values in the Oracle Business Analytics Warehouse will not be translated.
- There must be exchange rates defined for the currencies your organization deals with. If the appropriate exchange values are not found, the ETL process uses the ETL Unknown Exchange Rate defined in the DAC Source System Parameters.

F

About the Versioned Informatica Repository

This section explains the versioned Informatica Repository that is included in Oracle BI Applications, and contains the following topics:

- Section F.1, "Summary of Versioning in the Informatica Repository"
- Section F.2, "Customization of Repository Objects Using Check Out and Check In"

F.1 Summary of Versioning in the Informatica Repository

The current version of Oracle BI Applications ships the following two Informatica Repository files:

- Oracle_BI_DW_Base.rep
- Oracle_BI_DW_Teradata.rep

In releases 7.9.0, 7.9.1 and 7.9.2, these were non-versioned files provided in the standard application. In Oracle BI Applications version 7.9.3 and later, these Informatica Repository files are now versioned. Because of this change, you need the Team based Development option. This option is included in the OEM license provided to you by Oracle. Also, if you extend or modify the standard mappings, you need to check out and check changes to the Informatica Repository. DAC will now only pick up and execute valid and checked-in objects from the Informatica Repository.

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

F.2 Customization of Repository Objects Using Check Out and Check In

When working with an Informatica Repository that is versioned, you must check out a repository object to modify and then check in the change. To extend or modify the standard mappings, you must check out a repository object that is to be modified, make the desired changes, and then check in the changes. This process is described below, and is illustrated in the screenshot below where the red arrow represents the action flow.

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

- Developers check out an object for modification.

Note the green arrow on the object in the screenshot below.

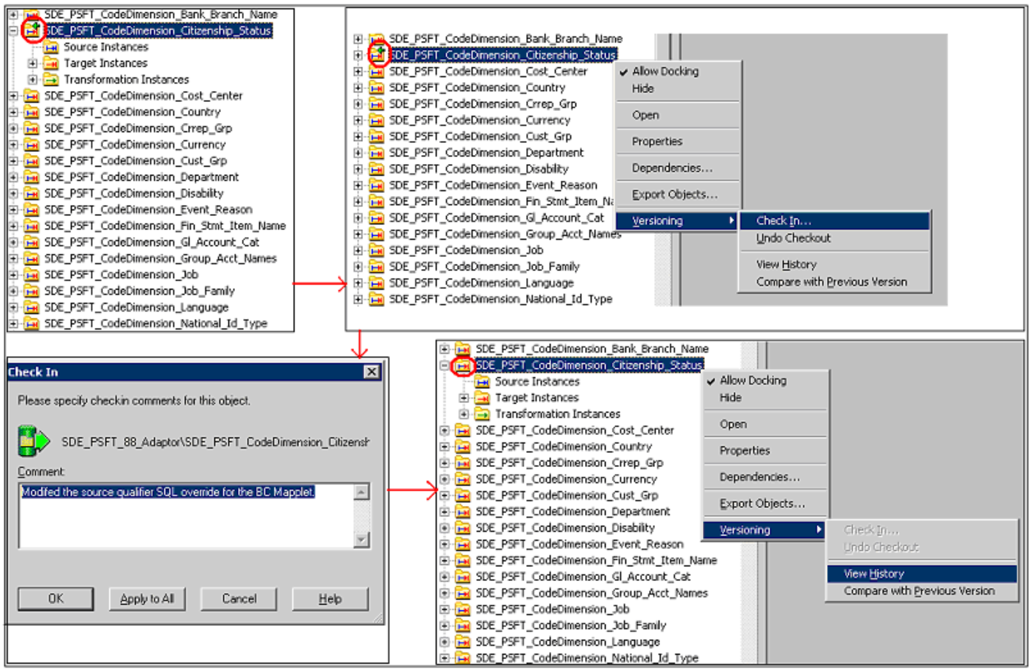
- Developers make changes, validate and save the object, and are ready to check their changes in.

  The green arrow is still displayed to indicate this is still a checked out object.

- Developers provide mandatory check in comments and finally check in.
- The object is checked in now.

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

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


SDE_PSFT_CodeDimension_Bank_Branch_Name
SDE_PSFT_CodeDimension_Citizenship_Status
Source Instances
Target Instances
Transformation Instances
SDE_PSFT_CodeDimension_Cost_Center
SDE_PSFT_CodeDimension_Country
SDE_PSFT_CodeDimension_Crrep_Grp
SDE_PSFT_CodeDimension_Currency
SDE_PSFT_CodeDimension_Cust_Grp
SDE_PSFT_CodeDimension_Department
SDE_PSFT_CodeDimension_Disability
SDE_PSFT_CodeDimension_Event_Reason
SDE_PSFT_CodeDimension_Fin_Stmt_Item_Name
SDE_PSFT_CodeDimension_Gl_Account_Cat
SDE_PSFT_CodeDimension_Group_Acct_Names
SDE_PSFT_CodeDimension_Job
SDE_PSFT_CodeDimension_Job_Family
SDE_PSFT_CodeDimension_Language
SDE_PSFT_CodeDimension_National_Id_Type
Allow Docking
Hide
Open
Properties
Dependencies...
Export Objects...
Versioning
Check In...
Undo Checkout
View History
Compare with Previous Version

Check In

Please specify checkin comments for this object.

SDE_PSFT_88_Adaptor\SDE_PSFT_CodeDimension_Citizensh

Comment:

Modfed the source qualifier SQL override for the BC Mapplet.

OK | Apply to All | Cancel | Help

Index