



**SIEBEL**<sup>®</sup> 7  
eBusiness

**SIEBEL DATA WAREHOUSE INSTALLATION AND  
ADMINISTRATION GUIDE**

*VERSION 7.7, REV. A  
MARCH 2004*

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404  
Copyright © 2004 Siebel Systems, Inc.  
All rights reserved.  
Printed in the United States of America

No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior agreement and written permission of Siebel Systems, Inc.

Siebel, the Siebel logo, TrickleSync, Universal Agent, and other Siebel names referenced herein are trademarks of Siebel Systems, Inc., and may be registered in certain jurisdictions.

Other product names, designations, logos, and symbols may be trademarks or registered trademarks of their respective owners.

U.S. GOVERNMENT RESTRICTED RIGHTS. Programs, Ancillary Programs and Documentation, delivered subject to the Department of Defense Federal Acquisition Regulation Supplement, are “commercial computer software” as set forth in DFARS 227.7202, Commercial Computer Software and Commercial Computer Software Documentation, and as such, any use, duplication and disclosure of the Programs, Ancillary Programs and Documentation shall be subject to the restrictions contained in the applicable Siebel license agreement. All other use, duplication and disclosure of the Programs, Ancillary Programs and Documentation by the U.S. Government shall be subject to the applicable Siebel license agreement and the restrictions contained in subsection (c) of FAR 52.227-19, Commercial Computer Software - Restricted Rights (June 1987), or FAR 52.227-14, Rights in Data—General, including Alternate III (June 1987), as applicable. Contractor/licensor is Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404.

**Proprietary Information**

Siebel Systems, Inc. considers information included in this documentation and in Siebel eBusiness Applications Online Help to be Confidential Information. Your access to and use of this Confidential Information are subject to the terms and conditions of: (1) the applicable Siebel Systems software license agreement, which has been executed and with which you agree to comply; and (2) the proprietary and restricted rights notices included in this documentation.

# Contents

## **Introduction to Siebel Data Warehouse Installation and Administration Guide**

What's New in Siebel Data Warehouse Installation and Administration Guide, Version 7.7, Rev. A . . . . . 12

## **Chapter 1. Overview of Siebel Data Warehouse**

Siebel Data Warehouse Architecture . . . . . 15  
    Siebel Data Warehouse Architecture Components . . . . . 16  
About the Data Warehouse Application Console . . . . . 17  
    About DAC Repository Objects . . . . . 18  
    About the DAC Process Life Cycle . . . . . 21

## **Chapter 2. Installing and Configuring Siebel Data Warehouse**

Siebel Data Warehouse Installation and Configuration Process . . . . . 24  
Siebel Data Warehouse Deployment Configuration . . . . . 26  
    Operating System, Driver, and Connectivity Requirements for Siebel Data Warehouse . . . . . 27  
Database Parameters and Requirements for Siebel Data Warehouse . . . . . 28  
    IBM DB2 UDB Windows 2000-Specific Database Requirements for Siebel Data Warehouse . . . . . 29  
    IBM DB2 UDB zOS and OS/390-Specific Database Requirements for Siebel Data Warehouse . . . . . 31  
    Oracle-Specific Database Requirements for Siebel Data Warehouse . . . . . 32  
    Additional Suggestions for Optimizing Oracle Performance in Siebel Data

Warehouse . . . . .	34
SQL Server-Specific Database Requirements for Siebel Data Warehouse . . . . .	36
Informatica Server Requirements for Windows 2000 . . . . .	39
Deploying Informatica and Siebel Data Warehouse in Unicode and Non-English Environments . . . . .	40
Unicode to Unicode Parameter Settings for Siebel Data Warehouse . . . . .	40
Code Page (Multibyte/Single-Byte) to Unicode Parameter Settings for Siebel Data Warehouse . . . . .	42
Code Page (Multibyte/Single-Byte) to Code Page (Multibyte/Single-Byte) Parameter Settings for Siebel Data Warehouse . . . . .	44
Siebel Data Warehouse Unicode and Code Page Procedures . . . . .	45
Installing Siebel Data Warehouse-Related Software on Windows . . . . .	48
Installing Siebel Analytics (Data Warehouse Option) on Windows . . . . .	49
Installing the Informatica Server and Repository Server on Windows . . . . .	50
Installing Informatica Client Software on Windows . . . . .	52
Siebel Data Warehouse Data Code Pages Validation . . . . .	53
Installing Java SDK and Drivers . . . . .	53
Setting Up the DAC Client . . . . .	54
Creating ODBC Database Connections . . . . .	55
Creating the SSE Role . . . . .	56
Configuring the DAC Config.bat File . . . . .	56
Starting the DAC Client and Importing Seed Data . . . . .	56
Using the DAC's Data Warehouse Configurator Wizard . . . . .	58
Creating the Informatica Repository (in the DAC) . . . . .	58
Configuring the Informatica Repository (in Repository Server Administration Console) . . . . .	59
Dropping the Informatica Repository . . . . .	60
Configuring the Informatica Repository (in DAC's Data Warehouse Configurator) . . . . .	61
Creating Data Warehouse Tables . . . . .	63
Dropping Data Warehouse Tables . . . . .	64
Creating Delete Triggers . . . . .	65

Configuring Informatica Server and Workflow Manager . . . . . 66

    Configuring the Informatica Service . . . . . 66

    Configuring the Informatica Repository Server (in the Repository Administration Console) . . . . . 67

    Verifying the Informatica Server Host Name . . . . . 69

    Configuring the Informatica Repository Using Workflow Manager . . . . . 70

    Starting the Informatica Service on Windows . . . . . 71

    Adding the PATH Environment Variable for the DAC Server . . . . . 72

    Configuring Database Parameter Files . . . . . 72

Setting DAC ETL Preferences . . . . . 73

Setting Up DAC System Properties and Resources . . . . . 75

    Setting Up DAC System Properties . . . . . 76

    Registering Informatica Servers . . . . . 79

    Creating Database Connections . . . . . 80

    Configuring Email Recipients . . . . . 82

Setting Up the DAC Server . . . . . 83

Installing and Configuring Siebel Data Warehouse on UNIX . . . . . 84

    Setting Up Environment Parameters on UNIX (Prerequisite) . . . . . 84

    Installing Informatica Software on UNIX . . . . . 84

    Setting the Environment Variables for Informatica Server on UNIX . . . . . 90

    Configuring Informatica Repository from Informatica Workflow Manager on UNIX . . . . . 91

    Configuring the DAC Client and Server on Solaris and AIX . . . . . 92

    Starting the Informatica Repository Server on UNIX . . . . . 93

    Starting and Stopping the Informatica Server on UNIX . . . . . 93

Updating Siebel OLTP Schema Definitions . . . . . 95

    Backing Up the Siebel Repository . . . . . 95

    Importing the SIF File into the Siebel OLTP . . . . . 95

    Verifying the SIF Patch Application . . . . . 96

Creating Siebel Stored Procedures for DB2-UDB . . . . . 97

## **Chapter 3. DAC Quick Start**

Navigating the DAC User Interface . . . . .	100
Main Toolbar of the DAC . . . . .	100
DAC Toolbars and Right-Click Menu Commands . . . . .	101
Menu Bar of the DAC . . . . .	105
DAC Server Monitor Icon . . . . .	109
DAC Navigation Tree . . . . .	110
DAC Editable Lists . . . . .	111
Logging In to the DAC . . . . .	111
Tailoring Execution Plans for Your Needs . . . . .	112
Pruning Execution Plans in the DAC . . . . .	114
Pruning Subject Areas in the DAC . . . . .	114
Pruning Table Groups in the DAC . . . . .	114
Pruning Tasks and Dependencies in the DAC . . . . .	117
Disabling DAC Repository Entities . . . . .	117
Using the DAC Preview Functionality to View Tasks and Dependencies . . . . .	117
Distributing DAC Metadata . . . . .	120
Using the DAC Query Functionality . . . . .	122
DAC Query Commands and Operators . . . . .	122
DAC Query Examples . . . . .	123
Common DAC Query Procedures . . . . .	124
Executing an ETL with the DAC . . . . .	125
Handling ETL Failures with the DAC . . . . .	125

## **Chapter 4. DAC Functional Reference**

About the DAC Design View . . . . .	129
About the DAC Execution Plans Tab . . . . .	130
About the DAC Subject Areas Tab . . . . .	134
About the DAC Table Groups Tab . . . . .	135
About the DAC Tables Tab . . . . .	136

About the DAC Indices Tab . . . . .	139
About the DAC Task Definitions Tab . . . . .	142
Using SQL Files as an Execution Type in the DAC . . . . .	148
About the DAC Setup View . . . . .	150
About the DAC System Properties Tab . . . . .	150
About the DAC Informatica Servers Tab . . . . .	150
About the DAC Database Connections Tab . . . . .	151
About the DAC Email Recipients Tab . . . . .	151
About the DAC Execute View . . . . .	152
About the DAC Execution Plans (Read Only) Tab . . . . .	152
About the DAC Current Run Tab . . . . .	152
About the DAC Run History Tab . . . . .	156
About the DAC Scheduler Tab . . . . .	156
About the DAC Validation Tab . . . . .	158

## **Chapter 5. Customizing the Siebel Data Warehouse**

Siebel Data Warehouse Customization Scenarios . . . . .	161
Siebel Data Warehouse ETL Process Concepts . . . . .	163
About Siebel Data Warehouse Extension Tables . . . . .	164
About the Siebel Data Warehouse DATASRC_NUM_ID Column . . . . .	166
Working with the Custom Folder of the Siebel Data Warehouse . . . . .	166
Creating Custom Informatica Workflows . . . . .	167
Important Notes About Siebel Data Warehouse Customizations . . . . .	169
Standard Templates for Siebel Data Warehouse Customizations . . . . .	172
Fact Templates for Siebel Data Warehouse Customizations . . . . .	173
Customization Steps for Siebel Data . . . . .	174
Adding a Column From a Main Source Table in the Siebel OLTP . . . . .	175
Adding a Column From a Related Source Table in the Siebel OLTP . . . . .	175
Modifying an Existing Column in the Siebel Data Warehouse . . . . .	177
Adding a New Dimension in the Siebel Data Warehouse and Using It with an Existing Fact . . . . .	177

Adding a New Fact Table in the Siebel Data Warehouse . . . . . 179

Adding a New Dimension Table for a New Fact Table in the Siebel Data Warehouse . . . . . 180

Siebel Data Warehouse Customization Steps for Non-Siebel Data . . . . . 180

    Adding New Data as a Whole Row into the Standard Dimension Table in Siebel Data Warehouse . . . . . 181

    Adding New Data as a Whole Row into the Standard Fact Table in Siebel Data Warehouse . . . . . 184

    Adding a Column in a Standard Dimension Table in Siebel Data Warehouse . . . . . 189

    Adding a Column for a Standard Fact Table in Siebel Data Warehouse . . . . . 191

    Building a Custom Dimension Table from External Sources in Siebel Data Warehouse . . . . . 192

    Linking the Custom Dimension to a Standard Fact Table in Siebel Data Warehouse . . . . . 193

    Building Custom Fact Tables from External Sources in Siebel Data Warehouse . . . . . 194

**Chapter 6. Using Siebel Data Warehouse Exception Reports**

Understanding Siebel Data Warehouse Exceptions . . . . . 197

Executing Siebel Data Warehouse Exception Reports . . . . . 201

**Chapter 7. Upgrading Siebel Data Warehouse**

Upgrade Requirements for Siebel Data Warehouse 7.7 . . . . . 208

Siebel Data Warehouse 7.7 Product Changes and Customization Impacts . 208

Siebel Data Warehouse Upgrade Paths . . . . . 209

Approaches to Upgrading the Siebel Data Warehouse . . . . . 210

Process of Siebel Data Warehouse Upgrade . . . . . 211

Data Migration Workflows for Siebel Data Warehouse Data Upgrades . . . 213

Running the Data Migration Workflow for Preconfigured Content in Siebel Data Warehouse . . . . . 215



Siebel Data Warehouse Upgrade Customizations . . . . .	216
Data Migration Workflows for Siebel Industry Applications Data Warehouse Upgrades . . . . .	221
Running the Data Migration Workflow for Preconfigured Siebel Industry Applications Content . . . . .	222

## **Chapter 8. Troubleshooting the Siebel Data Warehouse**

Siebel Data Warehouse Installation and Initial Configuration Issues . . . . .	225
Informatica and Loading Issues . . . . .	226
Unicode Issues with Informatica and the Siebel Data Warehouse . . . . .	230
UNIX Installation Issues with the Siebel Data Warehouse . . . . .	232
Siebel Data Warehouse Application Console (DAC) Issues . . . . .	235

## **Appendix A. Siebel Data Warehouse for Life Sciences Data Considerations**

Importing Syndicated Data into Siebel Data Warehouse . . . . .	239
Siebel Analytics Data Loading Matrix for Syndicated Data . . . . .	242
Data Source Values for Syndicated Data Attributes . . . . .	246
Data Loading Issues with Siebel Data Warehouse . . . . .	247
Supported Incremental Updates in the Siebel Data Warehouse LS Dimension Tables . . . . .	249

## **Index**



# Introduction to Siebel Data Warehouse Installation and Administration Guide

*Siebel Data Warehouse Installation and Administration Guide* explains how to install, configure, and administer the Siebel Data Warehouse (the physical data warehouse database) and its related components. Before it can be installed, the Siebel Data Warehouse database must be created manually by a database administrator.

Although job titles and duties at your company may differ from those listed in the following table, the audience for this guide consists primarily of employees in these categories:

<b>Database Administrators</b>	Administrators who administer the database system, including data loading, system monitoring, backup and recovery, space allocation and sizing, and user account management.
<b>Siebel Application Administrators</b>	Administrators who plan, set up, and maintain Siebel applications.
<b>Siebel Application Developers</b>	Developers who plan, implement, and configure Siebel applications, possibly adding new functionality.
<b>Siebel System Administrators</b>	Administrators responsible for the whole system, including installing, maintaining, and upgrading Siebel applications.

This guide assumes that you are knowledgeable in the areas of relational databases, decision support systems, dimensional design, and the operating system under which you are running the Siebel Analytics components.

## Product Modules and Options

This guide contains descriptions of modules that are optional and for which you may not have purchased a license. Siebel's Sample Database also includes data related to these optional modules. As a result, your software implementation may differ from descriptions in this guide. To find out more about the modules your organization has purchased, see your corporate purchasing agent or your Siebel sales representative.

# What's New in Siebel Data Warehouse Installation and Administration Guide, Version 7.7, Rev. A

**NOTE:** Siebel Analytics version 7.7 is compatible with Siebel eBusiness Applications version 7.5.3.

**Table 1. Changes Made in Version 7.7, Rev. A**

Topic	Revision
<a href="#">“IBM DB2 UDB zOS and OS/390-Specific Database Requirements for Siebel Data Warehouse” on page 31</a>	Added information about requirements specific to Siebel Data Warehouse for IBM DB2 UDB zOS and OS/390.
<a href="#">“Menu Bar of the DAC” on page 105</a>	Enhanced the descriptions for the DAC Repository Management tool.
<a href="#">“DAC Navigation Tree” on page 110</a>	Added information about using the navigation tree in the DAC.
<a href="#">“DAC Editable Lists” on page 111</a>	Added information about the editable lists in the DAC.
<a href="#">“Tailoring Execution Plans for Your Needs” on page 112</a>	Clarified information about pruning execution plans and added information about the chaining process related to table groups.
<a href="#">“Using the DAC Preview Functionality to View Tasks and Dependencies” on page 117</a>	Clarified the description of the information available in the preview.
<a href="#">“Distributing DAC Metadata” on page 120</a>	Added procedure for distributing DAC metadata to different environments.
<a href="#">“About the DAC Design View” on page 129</a>	Added descriptions of the actions available in each tab of the DAC Design view.

**Table 1. Changes Made in Version 7.7, Rev. A**

Topic	Revision
<a href="#">“About the DAC Setup View” on page 150</a>	Added descriptions of the actions available in each tab of the DAC Setup view.
<a href="#">“About the DAC Execute View” on page 152</a>	Added descriptions of the actions available in each tab of the DAC Execute view.
<a href="#">“Using SQL Files as an Execution Type in the DAC” on page 148</a>	Added procedure for using SQL files as an execution type in the DAC. Includes examples of SQL and XML files.
<a href="#">“Using Siebel Data Warehouse Exception Reports” on page 197</a>	Added information about executing exception reports in the DAC.

Table 2 lists changes described in version 7.7 of the documentation to support version 7.7 of the Siebel Analytics software.

**Table 2. New Product Features in Siebel Data Warehouse Installation and Administration Guide, Version 7.7**

Topic	Description
<a href="#">“DAC Quick Start” on page 99</a>	Describes how to use the Data Warehouse Application Console (DAC) to design, execute, and monitor ETL processes.
<a href="#">“DAC Functional Reference” on page 129</a>	Provides detailed information about designing, executing, and monitoring ETL execution plans.
<a href="#">“Customizing the Siebel Data Warehouse” on page 161</a>	Describes how to customize the preconfigured execution plans.
<a href="#">“Unicode Issues with Informatica and the Siebel Data Warehouse” on page 230</a>	Provides troubleshooting information about Unicode.
<a href="#">“UNIX Installation Issues with the Siebel Data Warehouse” on page 232</a>	Provides troubleshooting information about UNIX.

**Table 2. New Product Features in Siebel Data Warehouse Installation and Administration Guide, Version 7.7**

Topic	Description
<a href="#">“Siebel Data Warehouse Application Console (DAC) Issues” on page 235</a>	Provides troubleshooting information about the DAC.
<a href="#">“Upgrading Siebel Data Warehouse” on page 207</a>	Provides information about upgrading Siebel Data Warehouse to version 7.7.

# Overview of Siebel Data Warehouse

# 1

The Siebel Data Warehouse (also referred to as the Siebel Relationship Management Warehouse) is a unified data repository for all customer-centric data. The purpose of the Siebel Data Warehouse is to support the analytical requirements of Siebel eBusiness applications.

The Siebel Data Warehouse includes the following:

- A data integration engine that combines data from the Siebel transactional database and other data sources to build a data warehouse.
- An open architecture to allow organizations to use third-party analytical tools in conjunction with the Siebel Data Warehouse using the Siebel Analytics Server.
- Optional prebuilt data extractors to incorporate data from external applications into the Siebel Data Warehouse (licensed separately).
- A set of ETL (extract-transform-load) processes that takes data from the Siebel eBusiness application transactional data (OLTP) and creates the Siebel Data Warehouse tables.
- The Data Warehouse Application Console (DAC), a centralized console for the set up, configuration, administration, loading, and monitoring of the Siebel Data Warehouse.

## Siebel Data Warehouse Architecture

High-level analytical queries, like those commonly used in Siebel Analytics, 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 Siebel Data Warehouse was constructed using dimensional modeling techniques to allow for fast access to information required for decision making. The Siebel Data Warehouse derives its data from Siebel operational applications, and uses Informatica's data integration technology to extract, transform, and load data from the Siebel transactional database into the Siebel Data Warehouse.

Figure 1 illustrates the Siebel Data Warehouse architecture.

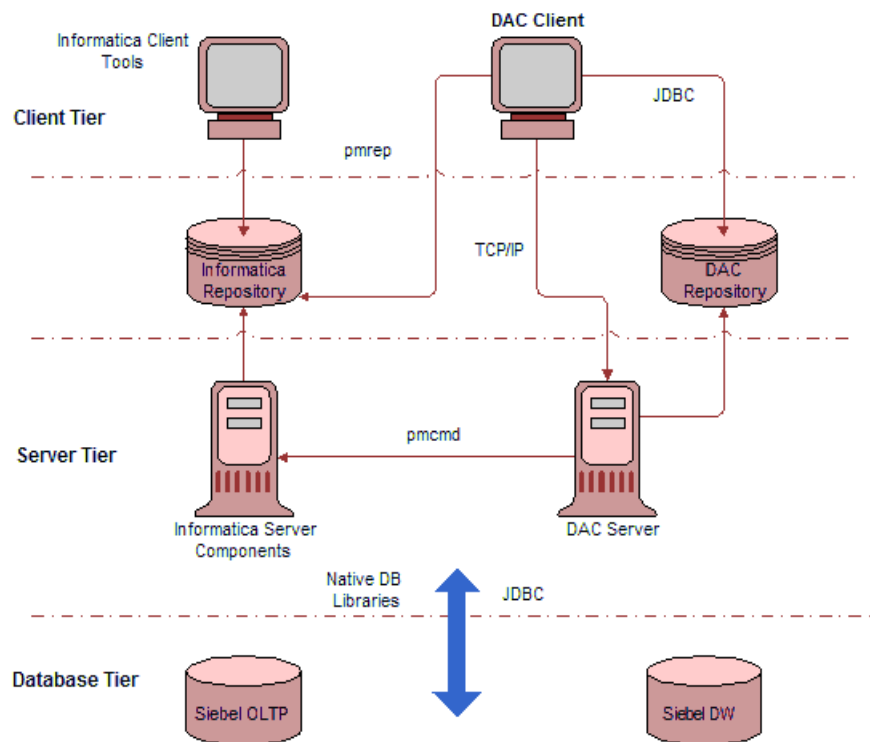


Figure 1. Siebel Data Warehouse Architecture

## Siebel Data Warehouse Architecture Components

The Siebel Data Warehouse architecture comprises the following components:



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

## About the Data Warehouse Application Console

The Data Warehouse Application Console (DAC) is a centralized console, providing access to the entire Siebel Data Warehouse application. It allows you to create, configure, and execute modular data warehouse applications in a parallel, high-performing environment.

Important features of the DAC include the following:

- Single command and control point for the Siebel Data Warehouse
- Dynamic load balancing of Informatica workflows across servers
- Dynamic load balancing of database tasks (non-Informatica) across database connections and databases

- Metadata-driven ETL execution plans and drop and create indices
- Registration of Informatica servers, Informatica databases, and source and target databases
- Automated configuration of ETL for full and incremental load of selected data warehouse subject areas
- Intelligent prioritization of sessions in queue during execution based on estimated run time and other criteria
- Structured history tracking and categorization of diagnostic logs by execution plans rather than by sessions
- Customization and extension of data warehouse subject areas
- Scheduling and optimal execution of the ETL process based on database and hardware resources
- Restarting of the ETL process from point of failure
- Email notification about the status of ETL processes

## About DAC Repository Objects

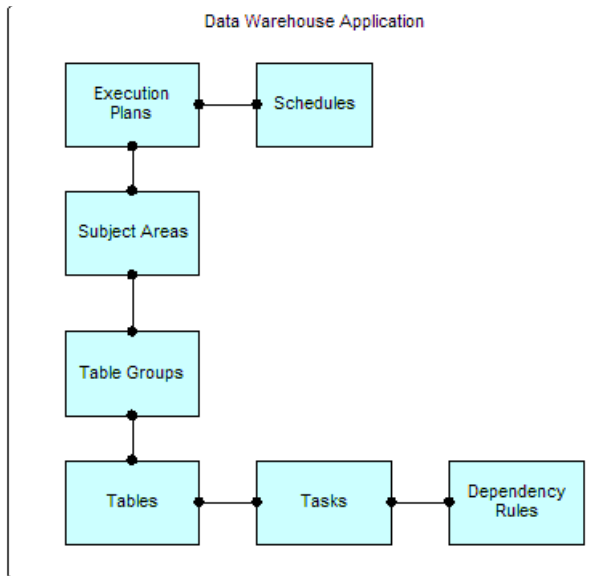
The DAC repository stores application objects in a hierarchical framework that defines a data warehouse application.

A data warehouse application comprises the following repository objects:

- **Tables.** Physical database tables defined in the database schema. Can be OLTP or data warehouse tables. Table types can be fact, dimension, hierarchy, aggregate, and so on.
- **Table group.** A logical grouping of tables that defines a logical relationship between its component tables. The table group definition is based on a central or a main table that drives the relationship between itself and the other tables in the group. For example, table groups can be star schemas based on a central fact and its dimensions and aggregates.

- **Subject area.** A logical grouping of table groups that is related to a particular subject or application context. The subject area derives its component tables indirectly from the table group. Subject Area definitions are used for scheduling. A subject area or set of subject areas can be scheduled for full or incremental loads on a single time schedule.
- **Execution plan.** A data transformation plan defined on subject areas that need to be transformed at certain frequencies of time. An execution plan is defined based on business requirements for when the data warehouse needs to be loaded. Execution plans are scheduled in sequence, not in parallel.
- **Task.** A Task can be related to data transformation or database objects. Tasks typically operate on a set of database tables, but they can be independent of the database, for example, creating a file or executing an OS command. Tasks can be of different types, such as extract, load, or batch script, and so on, and can be classified as pre-ETL or post-ETL.
- **Dependency Rule.** A rule that determines the order in which tasks are executed. Dependency rules are defined to determine the sequence of when tasks can run, ensuring that data consistency is maintained for the warehouse. Dependency rules are defined at the task level.
- **Schedule.** A schedule specifies when and how often an execution plan runs. An execution plan can be scheduled for different frequencies or recurrences by defining multiple schedules.

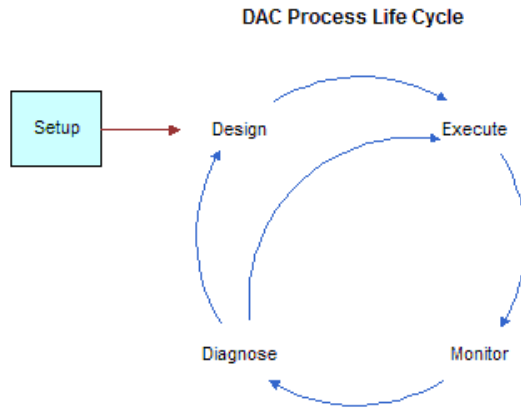
Figure 2 shows the hierarchical relationship among the repository objects.



**Figure 2. DAC Object Hierarchy**

## About the DAC Process Life Cycle

The DAC is used by different user groups to design, execute, monitor, and diagnose data warehouse applications. These phases together make up the DAC process life cycle, as shown in [Figure 3](#).



**Figure 3. DAC Process Life Cycle**

The phases of the process and the actions associated with them are as follows:

- Setup (Data warehouse developer)
  - Set up database connections
  - Set up application and database servers
  - Set up email recipients
- Design (Data warehouse developer)
  - Design data warehouse applications
  - Define application objects
- Execute (Data warehouse/ETL administrator)
  - Define scheduling parameters to execute data warehouse applications

- Deploy data warehouse applications
- Access run-time controls to restart or stop currently running schedules
- Monitor (Data warehouse/ETL administrator)
  - Monitor run-time execution of data warehouse applications
  - Monitor users, DAC repository, and application maintenance jobs
- Diagnose (Data warehouse/ETL administrator)
  - View the schedule diagnostics at run time or offline through log files
  - Run Exception Report execution plans

# Installing and Configuring Siebel Data Warehouse

# 2

This section describes the processes involved in installing the various clients and servers used for the Siebel Data Warehouse.

---

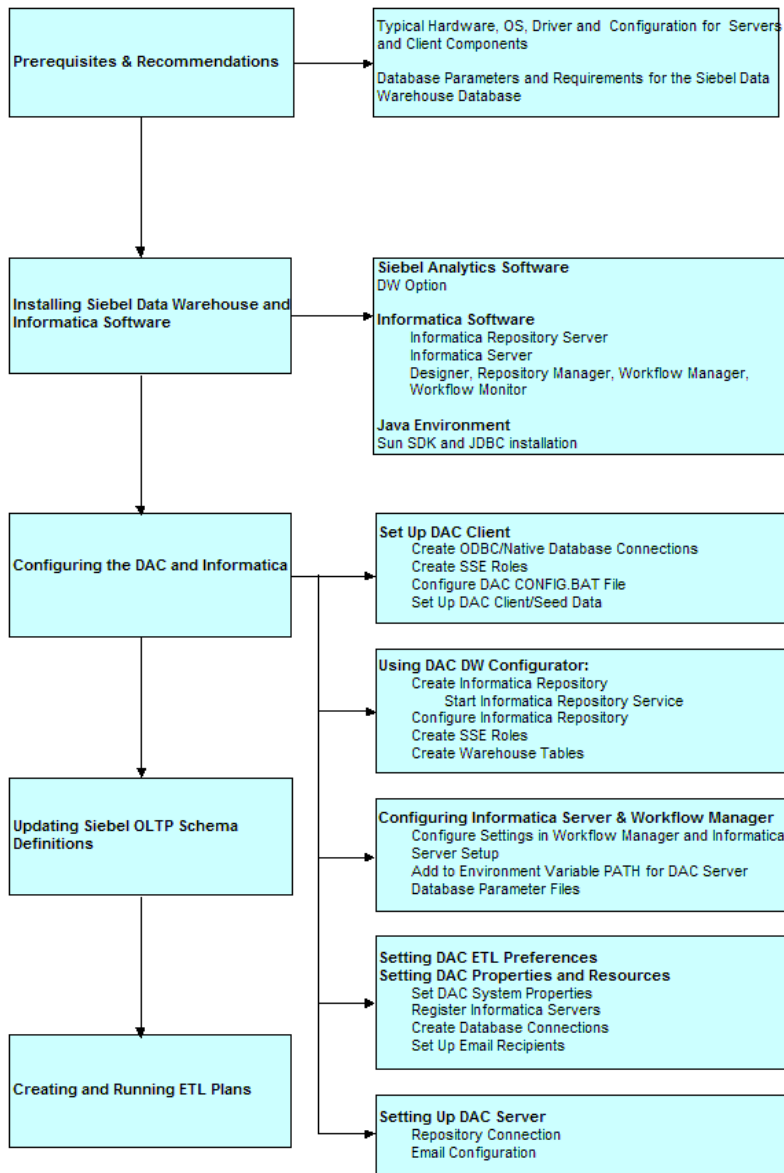
**NOTE:** To install the server components, the computers need to meet the conditions specified in *Siebel System Requirements and Supported Platforms*.

---

# Siebel Data Warehouse Installation and Configuration Process

[Figure 4](#) shows the sequence of tasks required to complete the Siebel Data Warehouse installation and configuration process.

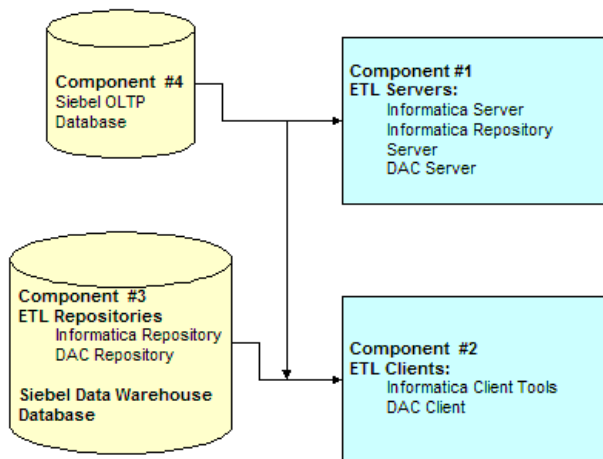




**Figure 4. Siebel Data Warehouse Installation and Configuration Process**

# Siebel Data Warehouse Deployment Configuration

Figure 5 shows the recommended Siebel Data Warehouse deployment configuration.



**Figure 5. Siebel Data Warehouse Deployment Configuration**

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

---

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

---

- Component 2 hosts all the ETL clients, that is, the Informatica client tools and the DAC client.

- Components 3 and 4 are database instances that can be hosted on one or more machines. The hardware requirements are entirely dependent on your usage and performance requirements. It is highly recommended that each of these components be defined in their own database instance to allow for instantiating parameters that are optimized for each component’s usage.

---

**NOTE:** For more detailed information about system requirements, see *Siebel System Requirements and Supported Platforms*.

---

## Operating System, Driver, and Connectivity Requirements for Siebel Data Warehouse

Table 3 provides the operating system, driver, and connectivity software requirements for the Siebel Data Warehouse components.

**Table 3. OS, Driver and Connectivity Requirements for Siebel Data Warehouse**

Component	Operating System	Software	Connectivity and Driver
1	<ul style="list-style-type: none"> <li>■ Solaris</li> <li>■ AIX</li> <li>■ HP (Informatica only)</li> <li>■ Windows 2000</li> </ul>	<ul style="list-style-type: none"> <li>■ Informatica Server 6.2.1</li> <li>■ Informatica Repository Server 6.2.1</li> <li>■ DAC Server 7.7</li> </ul>	<ul style="list-style-type: none"> <li>■ Sun SDK 1.4.1</li> <li>■ JDBC drivers</li> <li>■ Javamail</li> </ul>
2	<ul style="list-style-type: none"> <li>■ Windows 2000</li> </ul>	<ul style="list-style-type: none"> <li>■ Informatica client tools</li> <li>■ DAC client</li> </ul>	<ul style="list-style-type: none"> <li>■ Sun SDK 1.4.1</li> <li>■ JDBC drivers</li> <li>■ ODBC drivers</li> </ul>

**Table 3. OS, Driver and Connectivity Requirements for Siebel Data Warehouse**

Component	Operating System	Software	Connectivity and Driver
3 (A) Siebel DW	<ul style="list-style-type: none"><li>■ Solaris</li><li>■ AIX</li><li>■ HP</li><li>■ Windows 2000</li><li>■ OS/390</li></ul>	Database software with Siebel Data Warehouse	Not applicable
3 (B) ETL Repositories	<ul style="list-style-type: none"><li>■ Solaris</li><li>■ AIX</li><li>■ HP</li><li>■ Windows 2000</li></ul>	Database software with ETL repositories	Not applicable

## Database Parameters and Requirements for Siebel Data Warehouse

The Siebel Data Warehouse is a database that contains the dimensional schemas created by Informatica.

Although it is technically possible to put the Siebel Data Warehouse in the same database as the Siebel transactional database, it is not recommended for performance reasons. The Siebel transactional database is structured as an online transaction processing (OLTP) database, whereas the Siebel Data Warehouse is structured as an online analytical processing database. Each is optimized for its own purpose. The reasons for not combining the two databases are:

- The analytical queries interfere with normal use of the transactional database, which is entering and managing individual transactions.
- The data in a transactional database is normalized for update efficiency. Analytical queries join several normalized queries and will be slow (as opposed to prejoined, denormalized analytical tables).

- Historical data cannot be purged from a transactional database, even if not required for current transaction processing, because you need it for analysis. (By contrast, the analytical database is the warehouse for historical as well as current data.) This causes the transactional database to further slow down.
- 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.

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

The Siebel Data 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.

## **IBM DB2 UDB Windows 2000-Specific Database Requirements for Siebel Data Warehouse**

The following requirements apply to DB2 relational database management system (RDBMS) usage:

- ODBC driver for all connections must use the IBM DB2 ODBC Driver.
- Make the appropriate connections using DB2 Client Configuration Assistant.
- Create the user accounts that access the database. They must have SSE\_ROLE and be set so that the .dll commands can be run (permissions to create tables).
- One login user needs to be created in an empty database. The database objects will be created using this user login.
- The recommended baseline parameters are as follows:

## Installing and Configuring Siebel Data Warehouse

### *Database Parameters and Requirements for Siebel Data Warehouse*

```
MON_HEAP_SZ = 90
JAVA_HEAP_SZ = 2048
SHEAPTHRES = 400000
ASLHEAPSZ = 15
RQRIOBLK = 32767
QUERY_HEAP_SZ = 1000
MAXAGENTS = 400
NUM_POOLAGENTS = 200
FCM_NUM_BUFFERS = 12288
INTRA_PARALLEL = YES
DBHEAP = 16384
DBHEAP = 16384
DATABASE_MEMORY = AUTOMATIC
CATALOGCACHE_SZ = 5558
LOGBUFSZ = 256
UTIL_HEAP_SZ = 10000
BUFFPAGE = 30000
ESTORE_SEG_SZ = 65536
NUM_ESTORE_SEGS = 16
LOCKLIST = 50000
APPGROUP_MEM_SZ = 30000
GROUPHEAP_RATIO = 70
APP_CTL_HEAP_SZ = 5000
SHEAPTHRES_SHR = SHEAPTHRES)
SORTHEAP = 40000
STMTHEAP = 40960
APPLHEAPSZ = 2560
PCKCACHESZ = 800
STAT_HEAP_SZ = 9000
DLCHKTIME = 10000
MAXLOCKS = 50
LOCKTIMEOUT = 1200
MAXAPPLS = 500
AVG_APPLS = 10
MAXFILOP = 500
LOGFILSIZ = 63753
LOGPRIMARY = 200
LOGSECOND = 2
SOFTMAX = 100
LOGRETAIN = OFF
```

- All tables with a prefix of S\_ETL need to be in a tablespace separate from the other OLTP tables. These ETL tables are used by the Siebel Data Warehouse and should not be part of the routine backup processes.

A complete listing of these tables can be found in *Siebel Data Warehouse Data Model Reference*.

## IBM DB2 UDB zOS and OS/390-Specific Database Requirements for Siebel Data Warehouse

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

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

The following editions of DB2 Connect are supported:

- **DB2 Connect Enterprise Edition (EE)**. This edition is installed on a midtier server such as an Informatica Server/Client, DAC, and Siebel Analytics.
  - **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 4](#).

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

Parameter	Recommended Setting	Notes
IDTHTOIN	1800	

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

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

## Oracle-Specific Database Requirements for Siebel Data Warehouse

The following requirements apply to Oracle RDBMS usage:

- Make the appropriate connections using Oracle Net8 Assistant (for Oracle 8 databases) or Oracle Net Service (for Oracle 9 databases).
- Create the user accounts that access the database. They must have SSE\_ROLE and be set so that the .dll commands can be run (permissions to create tables).
- Set the tablespace for the Siebel Data Warehouse. Recommended values are shown in [Table 5](#).
- Create a login user with the tablespace as the default tablespace. The database objects will be created using this user login.
- Edit the Init.ora file. Use the recommended baseline parameters shown in [Table 5](#).

**Table 5. Recommended Variable Settings for Oracle Databases**

Parameter	Recommended Setting	Notes
Always_anti_join	HASH	
Cursor_space_for_time	TRUE	Use only if you do not have many concurrent connections.



**Table 5. Recommended Variable Settings for Oracle Databases**

Parameter	Recommended Setting	Notes
Db_block_lru_latches	Set to number of CPUs	
Db_block_size	32 k	
Db_file_direct_io_count	64	
Db_file_multiblock_read_count	STMAXIO / DB_BLOCK_SIZE	
Db_file_multiblock_read_count	32	
Default Pctincrease	0	
Hash_area_size	20 MB	
Hash_multiblock_io_count	0.5 * (db_file_multiblock_Read_count)	Make sure that you are <i>not</i> using MTS.
Initial Extent	20 MB	
Log_buffer	16 MB	
Next Extent	20 MB	
Optimizer_index_caching	Unset	Unset this to avoid nested loops index joins (as favored by the optimizer).
Optimizer_index_cost_adjust	100	See the note following these parameters.
Optimizer_max_permutations	2000	May be applicable to other schemas in the database. This means longer parsing times for the access paths, but once these are parsed, they should remain in the shared_pool, provided they do not get aged out. See notes for Cursor_space_for_time.

**Table 5. Recommended Variable Settings for Oracle Databases**

Parameter	Recommended Setting	Notes
Optimizer_mode	ALL_ROWS	Cost-based.
Query_rewrite_integrity	USE_STALE	
Rollback_segments		Initial, Next, and Optimum extent size of 50 MB.
Shared_pool_size	150 MB	
Sort_area_retained_size	10 MB	
Sort_area_size	10 MB	
Sort_multiblock_read_count	4	
Star_transformation_enabled	TRUE	
Tablespace	At least 50% of Siebel transactional database size	Make sure the temporary tablespace has adequate space.
Number of log file groups	4	
Size of each Log file	10 MB	
Sga_max_size	700 MB	

### Additional Suggestions for Optimizing Oracle Performance in Siebel Data Warehouse

- Siebel eBusiness Applications under Oracle support only binary sorting. If you are running an Oracle client, do one of the following:
  - Set the NLS\_SORT parameter to BINARY.
  - Choose a NLS\_LANG setting that includes binary.

These settings are required for adequate performance from the Dedicated Web Client.

- Make sure that cost-based optimization is enabled in the Oracle development, test, and production databases and that statistics are kept up to date. Otherwise, the rule-based optimizer may be used.
- Create foreign keys in the Oracle database, but configure Oracle to not enforce the foreign key relationship. The existence of foreign keys will allow Oracle to better optimize certain queries. By turning off enforcement, the database load should not be negatively affected.
  - In addition, you may also configure the Oracle star-join transformation. This requires non-enforced foreign keys in Oracle and the creation of necessary bitmap indices. This task is optional. It may not be necessary, as ongoing tuning may reach the desired performance goals.
- Analyze application for occurrences of highly skewed data that is indexed. Create histogram statistics for these indices to enable the optimizer to better
- To increase data throughput between Siebel Analytics 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=8192)(TDU=8192)  
        ORACLE_HOME = /.....)  
        SID_NAME = SOLAP)  
    )  
)
```

- 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=8192)(TDU=8192)
```

```
ADDRESS = ( PROTOCOL = TCP ) ( HOST=myhost ) ( PORT=1521 )  
CONNECT_DATA= ( SID=ORCL )  
)
```

## SQL Server-Specific Database Requirements for Siebel Data Warehouse

This section lists requirements for SQL Server database usage.

---

**NOTE:** SQL Server users must make sure the SQL Server client software is installed.

---

---

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

---

SQL Server databases should be created with ANSI NULL option selected.

### **To set the ANSI NULL option**

- 1 In Enterprise Manager, select Database.
- 2 Right-click and choose Database properties.
- 3 Click the Options tab and select the box for ANSI NULL default.

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

### **To modify the DB Library Options setting**

- 1 In the program menu Microsoft SQL Server, launch the Client Network utilities.
- 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.

---

**To create the user accounts that access the database**

- 1 Assign these user accounts SSE\_ROLE.
- 2 Set these accounts to run the .dll commands (that is, set permissions to create tables).

**Recommended SQL Server Database Parameters**

If necessary, SQL Server components can be configured to optimize performance, as shown in [Table 6](#).

**Table 6. 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	

**Table 6. Recommended Variable Settings for SQL Server Databases**

Parameter	Recommended Setting	Notes
Fill factor	95 %	For insert-intensive transactions, set Fill Factor between 90 and 95 %. For better query performance, set Fill factor to 95 or even 100 %.
Index create memory	1024 KB	Default is 0
Lightweight pooling	0	
Locks	0	
Max degree of parallelism	0	Default is 0. This turns off parallelism. Max degree of parallelism should be left at 0, which means use parallel plan generation. It should be set to 1 (use only 1 process) if you run multithreaded components (for example, several EIM threads).
Max server memory	2000 MB	Default is 2147483647
Max text repl size	65536 B	
Max worker threads	100	Default is 255
Media retention	0	
Min memory per query	1024 KB	
Min server memory	500 MB	Default is 0
Nested triggers	1	
Network packet size	8192 B	Default is 4096
Open objects	0	

**Table 6. Recommended Variable Settings for SQL Server Databases**

Parameter	Recommended Setting	Notes
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.

## Informatica Server Requirements for Windows 2000

If the Informatica Server runs under Windows 2000, the following requirements must be in place to use the *post-session* email feature of Informatica. This feature sends a confirmation email when the data warehouse is populated with new data.

- A Microsoft Outlook mail client must be configured on the Informatica Server. This works with most versions of Outlook except Outlook 2000.
- Microsoft Outlook must be running on the Microsoft Exchange server.
- The account that sends post-session emails must have rights to start the Informatica service.
- The account must have a Microsoft Outlook account.

For more information, see Informatica's user guide on the *Siebel eBusiness Third-Party Bookshelf*.

## Deploying Informatica and Siebel Data Warehouse in Unicode and Non-English Environments

This section describes the different settings for Informatica servers and databases when deploying the Siebel Data Warehouse in Unicode and non-English environments. When you configure Informatica, the Informatica repository, and the databases, you will need to refer to this section.

The Siebel Data Warehouse can be deployed in various code page environments to support global deployments. The following (source and target) configurations 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)

### Unicode to Unicode Parameter Settings for Siebel Data Warehouse

Unicode requires the code page configuration described in this section.

---

**NOTE:** Run the Informatica Server in Unicode mode when the source data supports multibyte or ISO 8859-1 (8 bit ASCII) code pages or when the source and target database is Unicode. If not using a Unicode database, select the ASCII option.

---



Table 7 shows the DBMS settings for Unicode to Unicode environments.

**Table 7. Unicode to Unicode Settings**

<b>Parameters</b>	<b>Oracle</b>	<b>IBM DB2 UDB</b>
Source DB code page.	UTF8 or AL32UTF8.	UCS-2.
Source DB connection.	In Workflow Manager > Connections > Relational, set to UTF-8.	In Workflow Manager > Connections > Relational, set to UTF-8.
Target DB (Informatica repository and DAC repository code page.	UTF8 or AL32UTF8.	UCS-2.
Target DB Connection code page.	In Workflow Manager > Connections > Relational, set to UTF-8.	In Workflow Manager > Connections > Relational, set to UTF-8.

**Table 7. Unicode to Unicode Settings**

Parameters	Oracle	IBM DB2 UDB
Informatica Server, Informatica Repository Server, Informatica client, DAC server, and DAC client code page.	<p>Any OS.</p> <p>Set the NLS_LANG to x_x.AL32UTF8. (For information about determining the NLS_LANG, see <a href="#">“Siebel Data Warehouse Unicode and Code Page Procedures”</a> on page 45.)</p> <p>If the Informatica Server is on UNIX, set PM_CODEPAGENAME appropriately (for example, MS932 if server code page is Japanese) and set the Informatica Server code page in Workflow Manager appropriately (for example, MS Windows Japanese, superset of Shift- JIS).</p>	<p>Any OS.</p> <p>Set DBCODEPAGE to 1208.</p> <p>If the Informatica Server is on UNIX, set PM_CODEPAGENAME appropriately (for example, MS932 if server code page is Japanese) and set the Informatica Server code page in Workflow Manager appropriately (for example, MS Windows Japanese, superset of Shift- JIS).</p>
Informatica Server (Unicode settings).	<p>Set SIEBELUNICODEDB appropriately. (For more information, see <a href="#">“Siebel Data Warehouse Unicode and Code Page Procedures”</a> on page 45.)</p> <p>Set the Data Movement mode to Unicode.</p>	<p>Set SIEBELUNICODEDB appropriately. (For more information, see <a href="#">“Siebel Data Warehouse Unicode and Code Page Procedures”</a> on page 45.)</p> <p>Set the Data Movement mode to Unicode.</p>

### Code Page (Multibyte/Single-Byte) to Unicode Parameter Settings for Siebel Data Warehouse

For installation on multibyte code page environments, the data warehouse setup requires two operating system environments: one for the Informatica Repository Server and one for the Informatica Server.

Table 8 shows the DBMS settings for an environment with a Japanese source database and a Unicode target Siebel Data Warehouse.

**Table 8. Japanese Code Page to Unicode Settings (Multibyte)**

Parameters	Oracle	IBM DB2 UDB
Source DB code page.	JA16SJIS.	943.
Source DB connection code page.	In Workflow Manager > Connections > Relational, set to MS Windows Japanese, super set of shift JIS.	In Workflow Manager > Connections > Relational, set to MS Windows Japanese, super set of shift JIS.
Target DB (Informatica repository and DAC repository) code page.	UTF8 or AL32UTF8.	UCS-2.
Target DB connection code page.	In Workflow Manager > Connections > Relational, set to MS Windows Japanese, superset of Shift-JIS.	In Workflow Manager > Connections > Relational, set to MS Windows Japanese, superset of Shift-JIS.
Informatica Server code page and environment variable settings.	<p>Japanese OS.</p> <p>It is recommended that you explicitly set the NLS_LANG to JAPANESE_JAPAN.JA16SJIS. (For information about determining the NLS_LANG, see <a href="#">“Siebel Data Warehouse Unicode and Code Page Procedures”</a> on page 45.)</p> <p>If the Informatica Server is on UNIX, set PM_CODEPAGENAME to MS932 and set the Informatica Server code page in Workflow Manager to MS Windows Japanese, superset of Shift-JIS.</p>	<p>Japanese OS.</p> <p>It is recommended that you explicitly set the DB2CODEPAGE to 943.</p> <p>If the Informatica Server is on UNIX, set PM_CODEPAGENAME to MS932 and set the Informatica Server code page in Workflow Manager to MS Windows Japanese, superset of Shift- JIS.</p>

**Table 8. Japanese Code Page to Unicode Settings (Multibyte)**

Parameters	Oracle	IBM DB2 UDB
Informatica Server (Unicode settings).	Do not set the parameter SIEBELUNICODEDB. Set the Data Movement mode to Unicode.	Do not set the parameter SIEBELUNICODEDB. Set the Data Movement mode to Unicode.
Informatica Repository Server, Informatica client, DAC server, and DAC client code page and environment variable settings.	Japanese OS. Set the NLS_LANG to JAPANESE_JAPAN.AL32UTF. (For information about determining the NLS_LANG, see <a href="#">“Siebel Data Warehouse Unicode and Code Page Procedures”</a> on page 45.)  If the Informatica Repository Server is on UNIX, set PM_CODEPAGENAME to MS932.	Japanese OS. Set DBCODEPAGE to 1208.  If the Informatica Server is on UNIX, set PM_CODEPAGENAME to MS932.

## Code Page (Multibyte/Single-Byte) to Code Page (Multibyte/Single-Byte) Parameter Settings for Siebel Data Warehouse

Table 9 shows the DBMS settings for an environment with a Japanese source database and a Japanese target Siebel Data Warehouse.

**Table 9. Japanese Code Page to Code Page Settings (Multibyte)**

Parameters	Oracle	IBM DB2 UDB
Source DB code page.	JA16SJIS.	943.
Source DB connection code page.	In Workflow Manager > Connections > Relational, set to MS Windows Japanese, super set of shift JIS.	In Workflow Manager > Connections > Relational, set to MS Windows Japanese, super set of shift JIS.
Target DB (Informatica repository and DAC repository) code page.	JA16SJIS.	943.

**Table 9. Japanese Code Page to Code Page Settings (Multibyte)**

Parameters	Oracle	IBM DB2 UDB
Target DB connection code page.	In Workflow Manager > Connections > Relational, set to MS Windows Japanese, superset of Shift-JIS.	In Workflow Manager > Connections > Relational, set to MS Windows Japanese, superset of Shift-JIS.
Informatica Server, Informatica Repository Server, Informatica client, DAC server, DAC client code page and environment variable settings.	Japanese OS. It is recommended that you explicitly set the NLS_LANG to JAPANESE_JAPAN.JA16SJIS. (For information about determining the NLS_LANG, see <a href="#">“Siebel Data Warehouse Unicode and Code Page Procedures”</a> on page 45.) If the Informatica Server is on UNIX, set PM_CODEPAGENAME to MS932 and set the Informatica Server code page in Workflow Manager to MS Windows Japanese (superset of Shift-JIS).	Japanese OS. It is recommended that you explicitly set the DB2CODEPAGE to 943. If the Informatica Server is on UNIX, set PM_CODEPAGENAME to MS932 and set the Informatica Server code page in Workflow Manager to MS Windows Japanese (superset of Shift- JIS).
Informatica Server (Unicode settings).	Do not set the parameter SIEBELUNICODEDB. Set the Data Movement mode to Unicode.	Do not set the parameter SIEBELUNICODEDB. Set the Data Movement mode to Unicode.

## Siebel Data Warehouse Unicode and Code Page Procedures

This section provides the following procedures related to configuring Unicode and code pages.

- [“To determine the NLS\\_LANG value”](#) on page 46
- [“To determine the DB2CODEPAGE”](#) on page 46
- [“To set SiebelUnicodeDB for Windows”](#) on page 46

- “To set NLS\_LANG value on Windows” on page 47
- “To set the NLS\_LANG value on UNIX” on page 47
- “To set SiebelUnicodeDB on Informatica Server if source and target databases are Unicode on UNIX” on page 48

### To determine the NLS\_LANG value

- For Oracle databases:
  - In the data warehouse database, run the command  

```
SELECT * FROM V$NLS_PARAMETERS
```
  - Mark the NLS\_LANG  
[NLS\_LANGUAGE]\_[NLS\_TERRITORY].[NLS\_CHARACTERSET].  
For example:  

```
American_America.UTF8
```

### To determine the DB2CODEPAGE

- In the data warehouse database, run the command  

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

Example result: 1208

### To set SiebelUnicodeDB for Windows

- 1 Navigate to HKEY\_LOCAL\_MACHINE > SYSTEM > CurrentControlSet > Services > PowerMart > Parameters > MiscInfo.
- 2 On the right window panel, right-click and select New > String Value.
- 3 Rename the new string value SiebelUnicodeDB.
- 4 Double-click SiebelUnicodeDB.
- 5 In the Value data field, enter your [user\_OLTP]@[connectString\_OLTP]  
[user\_OLAP]@[ConnectString\_OLAP].  
For example, siebel@db204007 siebel@db204008.
  - For Oracle and DB2, leave a space between siebel@db204007 and siebel@db204008.

- For MSSQL, use system DNS name for [connectString\_OLTP] and [connectString\_OLAP].

---

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

---

### **To set NLS\_LANG value on Windows**

- 1** Navigate to Control Panel > System and click the Advanced tab. Click Environment Variables.
- 2** In System variables section, click New.
- 3** Create the system variable name.
  - For Oracle databases:
    - In the Variable Name field, enter NLS\_LANG.
    - In the Variable Value field, enter `x_x.UTF8`.  
where `x_x` is the character set.  
For example: `American_America.UTF8`.
  - For DB2 databases:
    - In Variable Name field, enter `DB2CODEPAGE`.
    - In Variable Value field, enter `1208`.
- 4** Click OK to close.
- 5** Reboot the machine after creating the variables.

### **To set the NLS\_LANG value on UNIX**

- 1** Determine the NLS\_LANG value.
- 2** Enter the following at the prompt:

```
setenv NLS_LANG <NLS_LANG value>
```

For example: `setenv NLS_LANG American_America.UTF8.`

---

**CAUTION:** Do not put these `setenv` commands in the `.cshrc` file. Other machines and usernames also use this file, and this command will corrupt their non-Unicode environments.

---

#### **To set SiebelUnicodeDB on Informatica Server if source and target databases are Unicode on UNIX**

- For all UNIX platforms, if both the transactional and data warehouse data sources are Unicode, add the following line in `/export/home/informatica/pm/pmserver.cfg` file.

```
SiebelUnicodeDB=<OLTP_TBO>@<OLTP_connectionString>  
<OLAP_TBO>@<OLAP_ConnectionString>
```

For example:

```
SiebelUnicodeDB=ora16221@sdcd1580i016_qa16b  
ora16222@sdcd1580i016_qa16b (Refer to instructions)
```

## Installing Siebel Data Warehouse-Related Software on Windows

This section includes the following topics:

- [“Installing Siebel Analytics \(Data Warehouse Option\) on Windows”](#) on page 49
- [“Installing the Informatica Server and Repository Server on Windows”](#) on page 50
- [“Installing Informatica Client Software on Windows”](#) on page 52
- [“Siebel Data Warehouse Data Code Pages Validation”](#) on page 53
- [“Installing Java SDK and Drivers”](#) on page 53



---

## Installing Siebel Analytics (Data Warehouse Option) on Windows

---

**NOTE:** Only the Siebel Data Warehouse installation-specific screens are described in this book. For a description of the other Siebel Analytics installation screens, see *Siebel Analytics Installation and Configuration Guide*.

---

### **To install the Siebel Data Warehouse on Windows**

- 1** Access the installation files on the installation CD-ROM, and then run the program setup.exe.
- 2** In the Siebel Analytics Installer screen, accept the terms of the license agreement, and then click Next.
- 3** Browse for and select the license file.
- 4** Accept the default installation directory, or browse for and select an alternative directory.
- 5** Select the Data Warehouse installation option.
- 6** Select the language in which error messages are presented.
- 7** Select the following Data Warehouse components to be installed:
  - Console Server
  - Console Client
  - Console Client Utilities

---

**NOTE:** You only need to install the utilities on machines where the Informatica Server is installed. If you are using a machine only to run the DAC client or server, you do not need to install the utilities on it.

---

- 8** An alert dialog box informs you that the DAC requires Java SDK 1.4.1 or greater. Click OK.
- 9** Read the summary information about which features will be installed, and then click Next.

- 10 When the installation is completed, click Next.
- 11 Choose to restart your computer now or at a later time, and click Finish.

## Installing the Informatica Server and Repository Server on Windows

This section describes installation of the Informatica Server and Repository Server.

### **To install the Informatica Server on Windows**

- 1 In the Informatica Platform Products Setup—User Information page, enter your registration information and click Next.
- 2 In the Informatica Platform Products Setup—Select Components page check boxes, select PowerMart 6 OEM for Siebel and Server.
- 3 Click through the Welcome screen.
- 4 In the Edit Service Account page, enter the Domain, User, and Password for the Informatica Server service.
- 5 In the Choose Destination Location page, accept the default location or specify the location.
- 6 In the Informatica Server Setup Application - Options page, select Configure Informatica Service, and then click Continue.
- 7 In the Keys tab of the Configure Informatica Service dialog box, enter the Platform Key and the appropriate database server key.
- 8 In the Server tab, enter the following:

<b>Field</b>	<b>Enter</b>
Server Name	A logical name for the Informatica Server. It is recommended that you enter SIEBEL_DW_SERVER.
TCP/IP Host Address	The IP address or name of the machine on which you plan to run the Informatica Server.

The values in the remaining fields should be the default values.

- 9** In the Repository tab, enter the following:

<b>Field</b>	<b>Enter</b>
Repository Name	SIEBEL_DW_REP
Repository User	Administrator
Repository Password	Administrator
Repository Server Host Name	The host machine name where the Informatica Repository Server is installed.

- 10** In the Configuration tab, select the ASCII or UNICODE option and leave the default values in the remaining fields.
- If you select the UNICODE option, remove the check from the Validate Data Codepages check box, and check the Output Session Log in UTF8 check box.
- 11** In the Compatibility and Database tab and the JavaSdk tab, leave the default values in all fields.
- 12** Click OK in the Configure Informatica Service dialog box to complete the configuration of the Informatica Server service.
- 13** In the Informatica Server Setup Application - Options dialog box, click Exit.

**To install the Informatica Repository Server on Windows**

- 1** In the Informatica Platform Products Setup—User Information page, enter your registration information.
- 2** In the Informatica Platform Products Setup—Select Components page check boxes, select Informatica Repository Server.
- 3** Click through the Welcome screen.

- 4 In the Edit Service Account screen, enter the machine login information:

Field	Enter
Domain	Local machine or domain name of the user who has Administrator privileges
User	The Windows login user
Password	Password for this user

- 5 In the Choose Destination Location window, browse for and select the Repository Server destination folder. Click Next.
- 6 Click Next to proceed with the installation.
- 7 When the installation is complete, click Next again.
- 8 In the Configure Repository Server screen, enter the Server Port Number and Administrator Password. Click OK.

---

**NOTE:** The Server Port Number default is 5001.

---

- 9 In the Setup Complete window, click Finish.

## Installing Informatica Client Software on Windows

This section describes installation of the Informatica client, which is used to build and administer the Siebel Data Warehouse.

- Informatica Designer configures and updates the Siebel Data Warehouse mappings, transformations, and metadata.
- Informatica Repository Manager restores and manages the Siebel Data Warehouse repository.
- Informatica Workflow Manager executes the workflows.
- Workflow Monitor monitors the status of workflows and sessions.

**To install PowerMart Client**

- 1** In the Informatica Platform Products Setup—User Information page, enter your registration information and click Next.
- 2** In the Informatica Platform Products Setup—Select Components page check boxes, select PowerMart 6 OEM for Siebel and Client.
- 3** Click through the Welcome and License Agreement screens.
- 4** In the Destination Location screen, accept the default location or specify another.
- 5** Click through the Select Program Folder screen.
- 6** In the Select Components screen, select Program and Configuration Files.
- 7** In the Setup Complete screen, click Finish.

**Siebel Data Warehouse Data Code Pages Validation**

After installing the Informatica Server, Repository Server, and client tools, add the following section to the powermart.ini file located in the Informatica Client directory:

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

**Installing Java SDK and Drivers**

This section provides instructions for installing the Java SDK and the JDBC drivers.

---

**NOTE:** The DAC client requires the SDK for the Java 2 Platform, Standard Edition, Version 1.4.1.

---

**To install Java SDK 1.4.1 or above**

- Download the Java SDK from the Sun Web site, and install it using a directory path with no spaces, such as D:\j2sdk141

### To install JDBC drivers

- **Oracle.** If you are using an Oracle database (other than 8.x), find the directory where Oracle is installed. Copy the file named ojdbc14.jar and paste it in the DAC\lib directory.
  - If you are using Oracle 8.x, copy the file named classes12.zip and paste it in the DAC\lib directory. Also, edit the ORACLELIB setting in the config.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 Siebel stored procedures to be used during the ETL process.

---

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

## Setting Up the DAC Client

This section includes the following topics:

- [“Creating ODBC Database Connections” on page 55](#)
- [“Creating the SSE Role” on page 56](#)
- [“Configuring the DAC Config.bat File” on page 56](#)
- [“Starting the DAC Client and Importing Seed Data” on page 56](#)

## **Creating ODBC Database Connections**

The Informatica Server and Repository Server require native connections to the Siebel Data Warehouse and OLTP database. The Informatica client needs both an ODBC connection and a native connection to the Siebel Data Warehouse and OLTP database.

---

**NOTE:** If the Informatica Server is installed in MSSQL in non-ASCII mode, it will need an ODBC connection to the Siebel Data Warehouse and OLTP database.

---

### ***To create database connections for DB2 installations***

- 1** Using DB2 Client Configuration Assistant, create a database connection to the Siebel Data Warehouse, the Siebel OLTP, and the Informatica repository.
- 2** In the ODBC control panel, System DSN tab, create an ODBC connection to the Siebel Data Warehouse and OLTP using an ODBC driver.
- 3** Test the connections to make sure they work.

### ***To create database connections for Oracle installations***

- 1** Using Net8 Assistant or Oracle Net Service, create a native connect string (net service name) to the Siebel Data Warehouse, the OLTP, and the Informatica repository.
- 2** In the ODBC control panel, System DSN tab, create an ODBC connection to the Siebel Data Warehouse using the Siebel Merant ODBC driver that is supplied with Siebel Enterprise Server.
- 3** In the ODBC control panel, System DSN tab, create an ODBC connection to the Siebel Data Warehouse and OLTP using the Informatica-supplied Merant Closed driver. This ODBC driver was installed during the Informatica client installation (the exact name depends on the specific release of Informatica being used).
- 4** Test the connections to make sure they work.

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

- 1** Create a database connection to the Siebel Data Warehouse, the Siebel transactional database, and the Siebel Data Warehouse repository.

- 2 In the ODBC control panel, System DSN tab, create an ODBC connection to the Siebel Data Warehouse and OLTP using a SQL Server ODBC driver.
- 3 Test the connections to make sure they work.

## Creating the SSE Role

Use this procedure to create the SSE Role.

### To create the SSE Role

- Create a database role named SSE\_ROLE and make sure this role is granted the same privileges as the SSE\_ROLE on the Siebel OLTP.

## Configuring the DAC Config.bat File

### To configure the DAC

- 1 In the DAC directory, find and open the config.bat file.
- 2 Edit the JAVA\_HOME variable to point to the directory where you installed the Java SDK.  
  
Make sure there are no spaces in the path reference.
- 3 Edit the DAC\_HOME variable to point to the directory where you installed the DAC.

For example, your entries should look similar to the following:

```
set JAVA_HOME=d:\j2sdk1.4.1
set DAC_HOME=d:\SiebelAnalytics\DAC
```

## Starting the DAC Client and Importing Seed Data

The DAC client can connect to multiple DAC repositories but only one at a time. This procedure provides instructions for connecting to a repository for the first time. It guides you through creating a repository connection and importing seed data.



**To start the DAC Client**

- 1** Launch the DAC client by double-clicking the startclient.bat file in the DAC directory or by running startclient from a command prompt.
- 2** In the Login... dialog box, select Configure.
- 3** In the Configuring... dialog box, select Create Connection, and then click Next.
- 4** Enter the appropriate connection details.
- 5** Test the connection to make sure it works.

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

- 6** (Optional for Oracle or DB2 databases) Specify a tablespace.
- 7** Click Yes to have the repository created automatically.

---

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

---

- 8** Import seed data.
  - a** From the DAC menu bar, choose Tools > DAC Repository Management > Import.
  - b** Select the appropriate import option, and click OK.

---

**NOTE:** 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.

---

# Using the DAC's Data Warehouse Configurator Wizard

Complete the procedures in this section in the order in which they appear.

Use the Data Warehouse Configurator Wizard to complete the following procedures:

- [“Creating the Informatica Repository \(in the DAC\)” on page 58](#)
- [“Dropping the Informatica Repository” on page 60](#)
- [“Configuring the Informatica Repository \(in DAC's Data Warehouse Configurator\)” on page 61](#)
- [“Creating Data Warehouse Tables” on page 63](#)
- [“Dropping Data Warehouse Tables” on page 64](#)
- [“Creating Delete Triggers” on page 65](#)

Use the Informatica Repository Server Administration Console to complete the following procedure:

[“Configuring the Informatica Repository \(in Repository Server Administration Console\)” on page 59](#)

## Creating the Informatica Repository (in the DAC)

### *To create the Informatica repository*

- 1 From the DAC menu bar, click Tools > ETL Management > Configure.
- 2 Select the Create Informatica Repository check box.

The Informatica Repository tab is active.

- 3 Enter the following information:

Field	Description
Database Type	Type of database.
Table Owner	Valid database user.

<b>Field</b>	<b>Description</b>
Database Connection String	If you are using... <ul style="list-style-type: none"><li>■ Oracle (OCI8) use the tnsnames entry.</li><li>■ Oracle (Thin) use instance name.</li><li>■ SQL server use the database name.</li><li>■ DB2-UDB/DB2-390 use the connect string as defined in the DB2 configuration.</li></ul>
Host	Applicable only to SQL Server. Machine host name where the server resides.
Database Table Owner	Name of table owner.
Database Password	Password for table owner.
MSSQL Database Name	Applicable to SQL Server only. The database name where you want to restore the repository.
Informatica Administrator Name/Password	Not applicable while restoring the repository. Used only to configure the repository.
Repository Server	Name of machine hosting the Informatica Repository Server.
Repository Server Port	Port number used by Informatica Repository Server to listen to requests.
Email Address	Not applicable while restoring the repository. Used only to configure the repository.

## **Configuring the Informatica Repository (in Repository Server Administration Console)**

Before configuring the Informatica Repository, do the following:

- Create the ODBC connection for the Siebel Data Warehouse.
  - If you are using a SQL Server database, create an ODBC connection for the Siebel transactional database (OLTP). This is required if the SQL Server is Unicode.

- Make sure the Informatica software has been installed.

### **To configure the Informatica Repository**

- 1** Start the Informatica Repository Server service.
- 2** Start the Informatica Repository Server Administration Console.
- 3** Add a new server. It is recommended that you use the machine name where the Informatica Repository Server is installed.
- 4** Connect to the repository. The default password is Administrator.
- 5** Click Add Repository and add Siebel\_DW\_Rep to the Cache.
- 6** Enter the Database Type.
- 7** Enter the Repository Name as Siebel\_DW\_Rep.
- 8** Enter the Connect String Name. If you are using an MSSQL Server database, use `SERVERNAME@DATABASENAME`.
- 9** Enter the Code Page based on the operating system platform.
- 10** Enter a User and Password, and then click OK to save the information.
- 11** Select the repository, right-click and start the Informatica Repository Agent.

---

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

---

## **Dropping the Informatica Repository**

Use this procedure to drop the Informatica repository.

### **To drop the Informatica repository**

- 1** From the DAC menu bar, click Tools > ETL Management > Configure.
- 2** Select the Drop Informatica Repository check box.

- 3 In the Informatica Repository tab, enter the appropriate information, and then click Start.

## **Configuring the Informatica Repository (in DAC's Data Warehouse Configurator)**

Use this procedure to configure the Informatica repository in the Data Warehouse Configurator.

### ***To configure the Informatica Repository in the Data Warehouse Configurator***

- 1 From the DAC menu bar, click Tools > ETL Management > Configure.
- 2 Select the Configure Informatica Repository check box.
- 3 Select the Informatica Repository tab.
- 4 Enter the appropriate information, and then click Start.

---

**NOTE:** The Informatica repository cannot be configured if the source or target database is Unicode. Therefore, make sure the Is Unicode check box is not checked in the Data Warehouse tab and in the Transactional Database tab of the Data Warehouse Configurator. You then need to configure the Informatica repository in the Workflow Manager to specify the source and target databases as Unicode. See [“To set the database connection code page in Workflow Manager” on page 62.](#)

---

If your source or target database is on SQL Server and you have non-ASCII data, you must manually create database connections in the Informatica repository and replace the connections.

### ***To replace database connections for SQL Server databases that have non-ASCII data***

- 1 Start the Informatica Workflow Manager.
- 2 Connect to the repository.
- 3 Select Connections > Relational.

A new window appears.

- 4 Click Add, and then select ODBC.
- 5 If your source database is on SQL Server, create a connection using the following values:

Field	Value
Name	ODBC_OLTP
Code Page	UTF-8 encoding of Unicode

- 6 If your target database is on SQL Server, create a connection using the following values:

Field	Value
Name	ODBC_OLAP
Code Page	UTF-8 encoding of Unicode

The connections you created appear in the Replace Connections list.

- 7 Select Connections > Replace.
- 8 Click Add a New Connection Pair.
- 9 If your source database is SQL Server, enter MSSQL\_OLTP in the From field, and ODBC\_OLTP in the To field.
- 10 If your target database is SQL Server, click Add a New Connection Pair, and enter MSSQL\_OLAP in the From field, and ODBC\_OLAP in the To field.
- 11 Click Replace.

The database connection code page also has to be set manually through the Informatica Workflow Manager for Unicode/Non-Latin code pages.

### **To set the database connection code page in Workflow Manager**

- 1 Start the Informatica WorkFlow Manger.
- 2 Connect to the repository.

- 3 Select Connections > Relational.

A new window appears.

- 4 Select the appropriate OLTP connection (DB2\_OLTP for DB2, ORA\_OLTP for Oracle, ODBC\_OLTP for MSSQL).
- 5 For the Code Page, select UTF-8 encoding of Unicode or the appropriate code page, and click OK.
- 6 Select the OLAP connection (DB2\_OLAP for DB2, ORA\_OLAP for Oracle, ODBC\_OLAP for MSSQL).
- 7 For the Code Page select UTF-8 encoding of Unicode or the appropriate code page, and click OK.
- 8 Select the Informatica Repository tab.
- 9 Enter the appropriate information, and then click Start.

## Creating Data Warehouse Tables

Use this procedure to create data warehouse tables. Before you perform this procedure make sure the SSE role has been created for the Siebel Data Warehouse and OLTP and that the database user has been associated with the role.

---

**NOTE:** Additional work by the database administrator is required if the data warehouse tables need to be moved to different table spaces (for performance or manageability).

---

### **To create data warehouse tables**

- 1 From the DAC menu bar, click Tools > ETL Management Configure.
- 2 Select the Create Siebel Data Warehouse Tables check box.

The Data Warehouse tab is active.

- 3 Enter the following information:

Field	Description
Database Type	Type of database.
Table Owner	Valid database user.
Password	Valid database user password.
ODBC Data Source	Data Source Name (DSN) for the Siebel Data Warehouse.
Data Area	Tablespace where Data Warehouse tables are created.
Index Area	Tablespace where Data Warehouse indices are created (applicable only to Oracle and DB2 databases).
Is Unicode	Specifies whether the Data Warehouse database is Unicode.

### Dropping Data Warehouse Tables

Use this procedure to drop data warehouse tables.

#### **To drop data warehouse tables**

- 1 From the DAC menu bar, click Tools > ETL Management Configure.
- 2 Select the Drop Existing Siebel Data Warehouse Tables check box.  
The Data Warehouse tab is active.
- 3 Enter the following information:

Field	Description
Table Owner	Valid database user.
Password	Valid database user password.
ODBC Data Source	Data Source Name (DSN) for the Siebel Data Warehouse. <b>Note:</b> You must use the Oracle ODBC driver to drop the data warehouse tables.



## Creating Delete Triggers

The change capture process in Siebel Data Warehouse uses delete triggers to capture deleted records. The preconfigured ETL process captures deleted records for tables W\_ORG\_D and W\_PERSON\_D. The source tables for W\_ORG\_D and W\_PERSON\_D are S\_ORG\_EXT, S\_CONTACT, and S\_PRSP\_CONTACT; these tables need to have delete triggers created in the Siebel OLTP database in order to track deleted records.

The change capture mechanism from this release onwards will use delete triggers to be able to capture deleted rows information. Out-of-the box ETL captures deletes for W\_ORG\_D and W\_PERSON\_D. The source tables for these S\_ORG\_EXT, S\_CONTACT and S\_PRSP\_CONTACT need to have delete triggers created to be able to track deleted records. Such records are identified by setting the ACTIVE\_FLG column to D.

For vertical applications, there are additional source tables that need to have delete triggers created in the Siebel OLTP database to be able to track deleted records. These are S\_MDF\_TXN, S\_ASGN\_GRP\_POSTN, S\_ASGN\_RULE\_ITEM. The preconfigured ETL process captures deleted records for W\_FUND\_F and W\_ALIGNMT\_DH.

In Siebel Data Warehouse, preconfigured visibility tables are inactivated. If your organization is going to use any of the visibility-related data warehouse tables, you need to activate them in the Tables tab of the Design view of the DAC client. If you activate visibility tables, you should also create delete triggers on the optional tables.

When creating delete triggers using the Data Warehouse Configurator, you have the option of including the optional tables for which you want to enable the delete triggers. You also have the option of having the trigger statements be directly executed or being written to a file, which can then be executed by database administrator.

Before creating delete triggers, you need to create the appropriate database connections. For instructions, see [“Creating Database Connections” on page 80](#).

### **To create a delete trigger**

- 1 From the DAC menu bar, click Tools > ETL Management Configure.

- 2** Select the Create Delete Triggers in Transaction Database check box.  
The Delete Triggers tab is active.
- 3** Select Create Triggers.
- 4** Select the database where the triggers will be created.
- 5** (Optional) Select the Include optional triggers check box to include optional tables.
- 6** Click Start.

## Configuring Informatica Server and Workflow Manager

This section covers the following topics:

- [“Configuring the Informatica Service” on page 66](#)
- [“Configuring the Informatica Repository Server \(in the Repository Administration Console\)” on page 67](#)
- [“Verifying the Informatica Server Host Name” on page 69](#)
- [“Configuring the Informatica Repository Using Workflow Manager” on page 70](#)
- [“Starting the Informatica Service on Windows” on page 71](#)
- [“Adding the PATH Environment Variable for the DAC Server” on page 72](#)
- [“Configuring Database Parameter Files” on page 72](#)

### Configuring the Informatica Service

This configuration follows the installation of the Informatica servers.

**To configure the Informatica Service**

- 1 Navigate to the Configure Informatica Service screen. Make sure the settings shown in the following table are selected for each tab.

<b>Tab</b>	<b>Information</b>
Keys	Enter the applicable license keys for your installation. When finished, click OK. (Obtain the license key information from the license key letter.)
Server	Configure SIEBEL_DW_SERVER. In the Load Manger section, make sure that the maximum number of concurrent sessions is not greater than 15 and shared memory is not greater than 3000000.
Configuration	Specify whether to use ASCII or UNICODE. If you select UNICODE, clear the Validate Data Codepages check box.
Compatibility and Database	Make sure that the Database settings section Treat CHAR as CHAR on read is selected and that the maximum number of connections is 100.
Repository	Enter the repository name (Siebel_DW_Rep), the repository user and password, the repository server host name, and the repository server port number (5001). Click OK.

- 2 In the Informatica Server Setup Application—Options screen, click Exit.
- 3 In the Setup Complete screen, click Finish.

**Configuring the Informatica Repository Server (in the Repository Administration Console)**

After you restore the Informatica repository on the Administrator Workstation, you need to configure it. Set up and configure the repository connectivity, and then set up the client to point to the correct Informatica Server machine.

Before you configure the Informatica Repository Server, make sure that the Informatica Server is not running and the Informatica Repository Server is running.

**To start the Informatica Repository Server**

- In Windows Services, start the Informatica Repository Server.

Check the log file in the Informatica directory to make sure the server is running.

After you have started the Informatica Repository Server, use the PowerMart Repository Server Administration Console for this configuration procedure.

### **To configure the Informatica Repository Server**

- 1** Navigate to Start > Programs > Informatica PowerMart 6.1 OEM for Siebel—Client. Click the Repository Server Administration Console to open it.
- 2** In the main menu, click Action > Add a new Server Registration.
  - a** Enter the host name (the machine that has the repository).
  - b** Accept the default port number 5001.
  - c** Right-click the server name and choose Connect.
  - d** In the Connect to Repository section, Username and Password fields, enter Administrator.
- 3** Click on the Repository menu and select Add repository > Siebel\_DW\_Rep.
- 4** Right-click Siebel\_DW\_Rep and select Edit Connection.
  - a** If your repository server is on an English UNIX machine, set the Code Page to ISO 8859-1 Western European.

- b** To connect to the Siebel Data Warehouse repository, fill in the fields with this information:

<b>Field</b>	<b>Enter</b>
Database Type	DB2, Oracle, or MS SQL
Database user	OLAP user name
Database password	OLAP table password
Connect String	OLAP database alias name. The Informatica repository is usually stored in the data warehouse database folder.

Click OK.

- 5** Make sure that the Server is running.
- a** Under Windows, check the Windows Services.
  - b** Under UNIX, check the log file.

## **Verifying the Informatica Server Host Name**

After changing the server host name, verify the name change.

### **To verify that Informatica Server host name has been changed**

- 1** Right-click SIEBEL\_DW\_SERVER and select Monitor.

A red triangle appears next to the SIEBEL\_DW\_SERVER icon.

- 2** In Workflow Manager, highlight Siebel\_DW\_Rep. Click Server Configuration from menu bar and select Database Connections.

The Database Connections window opens. If you selected Configure Informatica Repository (enter source and target information) from the Siebel Data Warehouse installation, the Siebel Data Warehouse and Siebel transactional database should be configured already.

- 3 Highlight OLAP and click Edit. Verify that all information for the target database is correct. Verify that the Data Source field value is OLAP. Click OK or Cancel.

The target information should be the same as during the Siebel Data Warehouse installation.

- 4 Highlight OLTP and click Edit. Verify that all information for the source database is correct. Verify that the Data Source field value is OLTP. Click OK or Cancel.

The target information should be the same as during the Siebel Data Warehouse installation.

## Configuring the Informatica Repository Using Workflow Manager

Configure the Informatica repository using the following procedure.

### **To configure the Informatica repository**

- 1 Start the Informatica Workflow Manager.
- 2 In the Repositories list, select the SIEBEL\_DW\_REP repository.
- 3 Pull down the Repository menu and choose Connect.  
The Connect to Repository window opens.
- 4 Log in using the Administrator account (the password is also Administrator).
- 5 Click Connect.
- 6 Once connected, select the SIEBEL\_DW\_Server under the SIEBEL\_DW\_REP icon.
- 7 Pull down the Server Configuration menu and choose Edit Server.
- 8 In the Host Name field, enter the Computer Name (as shown in the Network control panel) of the Informatica Server.
- 9 Optionally, configure the Code Page setting:
  - If the Informatica Server is on ENU UNIX, pull down the Code Page menu and choose Western European. For localized environments, as an example, if the Informatica Server is running on a Japanese machine, choose MS Windows Japanese, Super Set of Shift JIS.

- If the Informatica Server uses ENU Windows, leave the setting unchanged. For localized environments, choose the appropriate setting.
- 10** Click Advanced. In the Advanced window, check to see if \$PMRootDir points to the directory where the Informatica server is to be installed, as in the following examples.

- For Windows platforms:

```
C:\Program Files\Informatica\PowerMart Server
```

- For UNIX platforms:

```
/export/home/informatica/pm
```

---

**NOTE:** If you follow the recommended installation, the Informatica server installs on the Informatica Server. If you install it on a different directory, make sure to update the \$PMRootDir variable with the new location.

---

- 11** Close the Server window.

Some error messages appear in the Messages pane. These are normal because the Informatica Server is not yet installed.

- 12** Select Siebel\_DW\_Server, right-click it, and choose Monitor a process.

A process monitoring window opens. Use this window monitor the Siebel Data Warehouse loading process.

- 13** Exit from Informatica PowerMart Workflow Manager.

## Starting the Informatica Service on Windows

The following procedure shows how to set up the Informatica service on Windows.

### **To start the Informatica Service on Windows**

- 1** In the Control Panel, open the Services panel.
- 2** Select the Informatica service, then double-click it.

- 3 In the Startup Type frame, select Automatic. In the Log On As frame, select System Account, then click OK.

---

**NOTE:** Do not use System Account if you plan on using email notification. Use *This account* with the appropriate user name and domain.

---

- 4 In the Services control panel, click Start to start the Informatica Service.

---

**NOTE:** If the Informatica Repository Manager and Workflow Manager are not pointing to the Siebel Data Warehouse, the Informatica service cannot start.

---

- 5 Close the Services control panel.

## Adding the PATH Environment Variable for the DAC Server

### *To add the PATH environment variable for the DAC server*

- Add to the PATH environment variable the path for the Informatica Server bin directory.

## Configuring Database Parameter Files

Use these procedures to configure the database parameter files.

- Go to the appropriate source database directory in `dwrep\Informatica\ParameterFiles` and copy the `parameterfileOLTP.txt` to the `SrcFiles` directory on the Informatica Server machine.
- Go to the appropriate target database directory in `dwdb\dwrep\Informatica\ParameterFiles` and copy the `parameterfileDW.txt` to the `SrcFiles` directory on the Informatica Server machine.
- For Siebel Industry Applications, go to the appropriate database directory in `dwrep\Informatica\ParameterFiles` and copy both `parameterfileOLTP.txt` and `sia_parameterfileOLTP.txt` to the `SrcFiles` directory on the Informatica Server machine.



- For Siebel Industry Applications, go the appropriate target database directory in `dwdb\dwrep\Informatica\ParameterFiles` and copy both `parameterfileDW.txt` and `sia_parameterfileDW.txt` to the `SrcFiles` directory on the Informatica Server machine.

## Setting DAC ETL Preferences

There are a number of ETL parameters that you can set using the DAC's ETL Preferences dialog box.

### **To access the ETL Preferences dialog box**

- 1 Launch the DAC client by double-clicking the `startclient.bat` file in the DAC directory or by running `startclient` from a command prompt.
- 2 From the DAC menu bar, choose `Tools > ETL Management Preferences`.

The available parameters are described in [Table 10](#).

**Table 10. ETL Preferences**

Parameter	Description
Analysis Start Date	The start date used to build the day dimension and to flatten exchange rates and costs lists.
Analysis End Date	The end date used to build the day dimension and to flatten exchange rates and costs lists.

**Table 10. ETL Preferences**

Parameter	Description
Prune Days	<p>The LAST_UPD column in Siebel OLTP tables is used for incremental change capture. This timestamp reflects the actual event time. It is therefore possible for a data row to be committed to the OLTP database with a LAST_UPD date that is older than the date on which the last refresh was executed. This will cause the data row to be missed in the subsequent extract (if based purely on LAST_UPD date).</p> <p>However, the LAST_UPD date column still provides an opportunity to improve the change capture process by overlapping the extraction date widow by the number of days set in this parameter. The records extracted in the overlapped window are filtered by comparing this information with information in the Image table.</p> <p>The Prune Days setting ensures that the rows that had values in LAST_UPD older than values in LAST_REFRESH_DATE are not missed. This is a parameter that can be set based on experience with processes, such as remote synch, that potentially can cause records to be missed. This parameter cannot be less than 1.</p> <p>For example: Assume the table W_PERSON_D was refreshed on January 15th by querying the table S_CONTACT. And, the Prune Days setting was set to 5. The next time S_CONTACT is queried to load W_PERSON_D, the records that have a LAST_UPD value since January 10 are compared with the ROW_ID of the Image table to cover for any missing records between January 15 and January 10 (the overlap period).</p>
Default OLTP Currency	During the ETL process, if a record's transaction currency is null, the ETL process uses the currency specified here as the default transaction currency.
DW Currency	The currency to which all financial transaction are converted during the load process.
Default Exchange Rate	The exchange rate used during a data warehouse currency conversion if certain currency exchange rates are not available.
Max Days in a Sales Stage	Applicable to opportunities only. The maximum number of days an opportunity can spend in a sales stage.

**Table 10. ETL Preferences**

Parameter	Description
Default Language	<p>The language to which columns' display strings are translated.</p> <p><b>Note:</b> Language-independent strings are stored in the <code>_l</code> columns. If you do not know what the default language setting is, issue the following query against the OLTP database:</p> <pre>select lang_id from S_LST_OF_VAL where type='ETL_UNSPEC_STR';</pre>
Default DW Index Space	<p>Applicable to Oracle databases only. The tablespace where indices are created.</p> <p><b>Note:</b> This tablespace is used only when indices are dropped and created. Once the indices are created, changing the value does not move them over.</p>
Alignment Version	<p>Applicable to vertical applications only. Seed data needs to be updated in certain lists of values to make sure that the ETL process correctly populates the alignment type and product category relationships in the Siebel Data Warehouse.</p> <p>Remove the check from this check box if you want to make changes to the existing alignment.</p> <p>Check this check box if you want to create a new alignment version.</p> <p>This setting is used by Siebel Life Sciences to control loading of changes in assignment rules in Siebel Assignment Manager.</p>

## Setting Up DAC System Properties and Resources

This section covers the following topics:

- [“Setting Up DAC System Properties” on page 76](#)
- [“Registering Informatica Servers” on page 79](#)
- [“Creating Database Connections” on page 80](#)
- [“Configuring Email Recipients” on page 82](#)

## Setting Up DAC System Properties

The DAC system properties determine the behavior of the DAC server. This section describes the system properties available in the DAC System Properties tab in the Setup view of the DAC client.

### To access the DAC System Properties tab

- 1 Launch the DAC client by double-clicking the startclient.bat file in the DAC directory or by running startclient from a command prompt.
- 2 Click Setup on the DAC toolbar.

The DAC System Properties tab is active.

Table 11 provides a description of the DAC system properties.

**Table 11. DAC System Properties**

Property	Description
Analyze Frequency (in days)	For the DAC repository table, the frequency (in days) the DAC client automatically updates the table and index statistics for the DAC repository. The value must be numerical.
Analyze Tables	For the data warehouse schema tables, indicates whether during the ETL process the DAC server automatically issues analyze table statements when a table is truncated and loaded. Possible values are True and False.
Auto Restart ETL	When the DAC server abnormally terminates during the execution of an ETL, the status remains as Running. The status of an ETL is updated only by the DAC server.  When this property is set to True, the server automatically continues running the ETL process. When set to False, the run statuses are updated to Failed. In this case, the administrator needs to restart the failed ETL manually.
CreateQueryIndexesAtTheEnd	During the ETL process, the DAC server automatically drops and creates indices. When set to True, this property groups all query type indices and runs them after the ETL is complete. Possible values are True and False.

**Table 11. DAC System Properties**

Property	Description
DAC Server Host	<p>Host name or IP address of the machine where the DAC server resides.</p> <p><b>Note:</b> 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.</p> <p><b>Note:</b> This property also takes the value <i>localhost</i>. However, this value is provided for development and testing purposes and should not be used in a production environment.</p>
DAC Server OS	<p>Operating system of the machine where the DAC server resides. Possible values are Windows, Solaris, or AIX.</p> <p><b>Note:</b> If you move the DAC server from another operating system to AIX, you need to change the DAC server host to the appropriate value, restart the DAC client, and reenter all the password fields for the Informatica servers and database connections, and reconfigure the DAC server on the AIX machine by running the <code>serverSetupPrompt.sh</code>.</p>
DAC Server Port	<p>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.</p>
DropAndCreateIndexes	<p>Indicates whether, during the ETL process, the DAC server automatically drops and creates indices. Possible values are True and False.</p>
Dryrun	<p>Indicates whether tasks are executed without invoking Informatica workflows. The following processes are executed: change capture, truncation of tables, drop and creation of indices, and analyze statements. Possible values are True and False.</p> <p><b>Note:</b> This option should be used for debugging purposes only and not used in a production environment.</p>
Generic Task Concurrency Limit	<p>Determines how many tasks with execution types other than Informatica can be run concurrently. The value must be numerical.</p>

**Table 11. DAC System Properties**

Property	Description
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.
Output Redirect	<p>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.</p> <p>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.</p>
Repository DB Pool Size	<p>Maximum number of connections to the DAC repository the DAC server will maintain. The value must be numerical.</p> <p><b>Note:</b> As the DAC server runs it needs to perform transactions and queries on the DAC repository. In order for it to do so efficiently, it internally maintains a pool of preexisting database connections. This reduces latency and leads to better use of resources through connection sharing. However, for this to happen, you need to specify the maximum size of the connection pool. This setting should be determined in consultation with the database administrator of the database where the DAC repository resides.</p>
Scheduler.Poll.Interval	Frequency (in seconds) the DAC server polls for changes in the schedule configuration.

**Table 11. DAC System Properties**

<b>Property</b>	<b>Description</b>
Script Before Every ETL	<p>The name of the script or executable to be run before every execution plan.</p> <p>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.</p> <p>The execution plan runs only after the external process has finished. Therefore, it is important that the script or executable does not fail.</p>
Script After Every ETL	<p>The name of the script or executable to be run after every execution plan.</p> <p>For more information, see the description of the property Script Before Every ETL.</p>
Server Log Level	<p>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.</p>
SQL Trace	<p>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 hints 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.</p> <p>It is recommended that you set this property to False.</p>

## **Registering Informatica Servers**

This section describes registering the Informatica servers in the DAC and specifying the number of workflows that can be executed in parallel. The DAC server automatically load balances across the servers and does not run more than the value specified for each of them. You can perform these tasks in the Informatica Servers tab in the Setup View of the DAC client.

You need to register at least one Informatica server. You should register one Informatica server for each Informatica repository.

The DAC server uses pmcmd to run the Informatica workflows on the Informatica servers. Make sure the path to the directory in which the Informatica server is installed is added to the DAC server system properties.

---

**NOTE:** It is recommended that the Informatica Server be installed on the machine where the DAC server is installed. If a machine is running only the DAC client, it does not need an Informatica server installed.

---

#### **To access the Informatica Servers tab**

- 1 Launch the DAC client by double-clicking the startclient.bat file in the DAC directory or by running startclient from a command prompt.
- 2 Click Setup on the DAC toolbar, and then click the Informatica Servers tab.

Table 12 provides a description of the Informatica server properties.

**Table 12. Informatica Server Properties**

Property	Description
Name	Logical name for the Informatica server.
Server Host Name	Informatica server host name or IP address.
Port	Port where the Informatica server connects for requests. (The default port number is 4001).
Login	Informatica user name who has appropriate privileges to execute workflows.
Password	Informatica user password.
Maximum Sessions	Maximum number of workflows that can be executed on the server. If the number of sessions is 0 or is not specified, the DAC server assigns the default value of 5.
Inactive	Indicates whether the Informatica server is active or inactive.

## Creating Database Connections

You can set up database connections in the Database Connections tab in the Setup View of the DAC client.



**To access the Database Connections tab**

- 1** Launch the DAC client by double-clicking the startclient.bat file in the DAC directory or by running startclient from a command prompt.
- 2** Click Setup on the DAC toolbar, and then click the Database Connections tab.

Table 13 provides a description of the database connection properties.

**Table 13. Database Connection Properties**

Property	Description
Name	Logical name for the connection. (The metadata that comes with Siebel Analytics Data Warehouse has two data sources: OLTP, which points to the Siebel Transaction Database, and the Data Warehouse.)
Type	Source or Warehouse.
Connection Type	Type of database connection. Possible values are the following: <ul style="list-style-type: none"> <li>■ <b>Oracle (OCI8)</b>. Connect to Oracle using the tnsnames entry.</li> <li>■ <b>Oracle (Thin)</b>. Connect to Oracle using thin driver.</li> <li>■ <b>DB2</b>. DB@ UDB database.</li> <li>■ <b>DB2-390</b>. DB2 390 database.</li> <li>■ <b>MSSQL</b>. Microsoft SQL Server.</li> </ul>
Connection String	If you are using... <ul style="list-style-type: none"> <li>■ Oracle (OCI8) use the tnsnames entry.</li> <li>■ Oracle (Thin) use instance name.</li> <li>■ SQL server use the database name.</li> <li>■ DB2-UDB/DB2-390 use the connect string as defined in the DB2 configuration.</li> </ul>
Table Owner	Name of table owner.
Password	Password for table owner.

**Table 13. Database Connection Properties**

Property	Description
DBHost	Host machine where the database resides. This field is mandatory if you are using Oracle (Thin) or MSSQL but is not required if you are using Oracle (OCI8), DB2, or DB2-390.
Port	Port where the database receives requests. Required for Oracle (Thin) and MSSQL databases. Not required for Oracle (OCI8), DB2, or DB2-390 databases.

## Configuring Email Recipients

You can set up email recipients in the Email Recipients tab in the Setup View of the DAC client.

### **To access the Email Recipients tab**

- 1 Launch the DAC client by double-clicking the startclient.bat file in the DAC directory or by running startclient from a command prompt.
- 2 Click Setup on the DAC toolbar, and then click the Email Recipients tab.

Table 14 provides a description of the email recipient properties.

**Table 14. Email Recipient Properties**

Property	Description
Name	Logical name of the user to be notified.
Email Address	Email address where the notification is sent.
Notification Level	The notification levels are as follows: <ul style="list-style-type: none"><li>■ 10 -- Notifies recipient of success or failure of each task.</li><li>■ 5 -- Notifies recipient of success of failure of the entire ETL process.</li><li>■ 1 -- Notifies recipient that ETL completed successfully.</li></ul>
Inactive	Indicates whether the record is active or inactive.

## Setting Up the DAC Server

This section provides the procedure for configuring the connection between the DAC server and the DAC repository and for setting up email recipients. You must perform this task on the machine hosting the DAC server. There should be one installation of the DAC server for each DAC repository. The repository configuration is mandatory. Setting up email recipients is optional.

The DAC server has a built-in login-authentication based email (SMTP) client, so it talks to any SMTP login-authenticating server to send emails. The DAC server notifies email recipients about the status of various on-going activities, such as the status of an ETL execution.

---

**NOTE:** On machines in which you must use Telnet, such as UNIX servers, you can also invoke the serverSetup script. This option leads you into setting up the server bootup properties and also the email account configuration.

---

### **To configure the DAC repository connection and email recipients**

- 1** Launch the DAC client by double-clicking the startclient.bat file in the DAC directory or by running startclient from a command prompt.
- 2** From the menu bar, click Tools > DAC Server Management DAC Server Setup.  
A confirmation dialog box asks you to confirm that you want to configure the DAC repository connection.
- 3** Click Yes.  
The Server Configuration dialog box appears.
- 4** In the Repository Configuration tab, enter the appropriate information.
- 5** Click Test Connection to make sure the connection works.
- 6** In the Email Configuration, enter the appropriate information, and click Save.

Once the configuration has been completed, you can start the DAC server by double-clicking on the startserver.bat file in the DAC directory or by running the startserver.bat from a command window.

# Installing and Configuring Siebel Data Warehouse on UNIX

This section provides instructions for installing and configuring the Informatica software in the UNIX environment. For instructions on installing the Siebel Data Warehouse software on UNIX, see *Siebel Analytics Installation and Configuration Guide*.

The following topics are included:

- [“Setting Up Environment Parameters on UNIX \(Prerequisite\)” on page 84](#)
- [“Installing Informatica Software on UNIX” on page 84](#)
- [“Setting the Environment Variables for Informatica Server on UNIX” on page 90](#)
- [“Configuring Informatica Repository from Informatica Workflow Manager on UNIX” on page 91](#)
- [“Configuring the DAC Client and Server on Solaris and AIX” on page 92](#)
- [“Starting the Informatica Repository Server on UNIX” on page 93](#)
- [“Starting and Stopping the Informatica Server on UNIX” on page 93](#)

## Setting Up Environment Parameters on UNIX (Prerequisite)

If you are using a non-ENU operating system, you must set the environment parameter `PM_CODEPAGENAME` appropriately. For example, if the Informatica repository has been restored from a machine running a Japanese operating system, set the `PM_CODEPAGENAME` environment variable to `MS392`. For more information, see [“Deploying Informatica and Siebel Data Warehouse in Unicode and Non-English Environments” on page 40](#).

## Installing Informatica Software on UNIX

This section provides instructions for installing the required Informatica software on UNIX.

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

Install the Informatica software directly from the CD-ROM.

---

**NOTE:** If you cannot install the software directly from the CD-ROM, you must use FTP in binary mode; otherwise, the installation will fail.

---

After the FTP process, the permission setting for all install files must be changed. The example below shows the install prompts as well as the values that should be entered.

**To change the permission setting for the install files**

- Enter the following:

```
chmod 777 *  
  
a0repser.icf imr.idm      pc.icf      pmlocale.idm repomisc.idm  
extproc.idm instalja.msg pmdoc.idm  pmmisc.idm  reposhr.idm  
icipc      install      pmeval     pmserver.idm reposql.idm  
icipc_pd   install.msg  pmexes.idm repoexes.idm rw.idm  
icirepo    lictxt.txt  pmlibs.idm repolibs.idm x0imr.icf
```

The installation process gets kicked off by running the following from the install directory:

```
./install  
  
=====
```

Welcome to Informatica Installation Wizard.

All Informatica products Copyright 1996-2003.

```
=====
```

Please enter your Product Key for Informatica Platform Products:  
EBLO-AFIF-YAZB-YGYJ-MVSM

The Infrastructure Products may be licensed solely for use with the Siebel Marketing Transaction DataBase or DataMart as the target. Infrastructure Products are licensed for restricted use

in extracting and loading data but not transforming such data with the Siebel Marketing DataMart as the source and not necessarily the target.

If included with the Siebel applications, a single license to the Infrastructure Products shall be restricted to use on 10 or fewer CPUs per server through the end of 2006, and thereafter shall be restricted to use on 14 or fewer CPUs per server. This CPU restriction shall not apply to customers who have obtained a Full Use License or an unlimited CPU license to Infrastructure Products from Siebel or Informatica.

Do you accept the above license (y/n) : y

### Installing the Informatica Repository Server

The example below shows the install prompts as well as the values that should be entered.

```
=====
Welcome to Informatica Installation Wizard.

All Informatica products Copyright 1996-2003.

=====
1. Informatica Repository Server 6.2
2. Informatica PowerMart Server 6.2 OEM for Siebel
3. Informatica Metadata Reporter 6.2
0. Exit

Please select a product to install: 1

Installing Informatica Repository Server 6.2

Please specify a directory to install to [/home/qa-sm/qauser7s/
informatica/repositoryserver] : /export/home/informatica/
repositoryserver

/export/home/informatica/repositoryserver does not exist. create
(y/n) : y
Installing module repoexes
Module repoexes extracted successfully.
Installing module repolibs
Module repolibs extracted successfully.
Installing module repomisc
```

```
Module repomisc extracted successfully.
Installing module pmlocale
Module pmlocale extracted successfully.
Installing module rw
Module rw extracted successfully.
Installing module reposql
Module reposql extracted successfully.
Installing module reposhr
Module reposhr extracted successfully.
Installing module pmdoc
Module pmdoc extracted successfully.

Product specific installation is starting...

Informatica(tm) PMREPCONFIG 6.2 (0317)
Copyright (c) Informatica Corporation 1994 - 2003
All Rights Reserved.

Configuration file name [pmrepserver.cfg]:
ServerPort[5001]:
AdminPasswd[]: Administrator
RaMinPort[5002]:
RaMaxPort[65535]:
ConfigDir[Config]:
BackupDir[Backup]:
PluginDir[Plugin]:
ErrorSeverityLevel( E[rror] | W[arning] | I[nfo] | T[race] |
D[ebug] ) [info]:
LogFileName[pmrepserver.log]:
Installation of Informatica Repository Server 6.2 has completed
successfully.

Would you like to view the readme file right now (y/n) : n

Press enter to continue...
```

## **Installing the Informatica Server**

The example below shows the install prompts as well as the values that should be entered.

```
=====

Welcome to Informatica Installation Wizard.

All Informatica products Copyright 1996-2003.
```

## Installing and Configuring Siebel Data Warehouse

*Installing and Configuring Siebel Data Warehouse on UNIX*

```
=====
1. Informatica Repository Server 6.2
2. Informatica PowerMart Server 6.2 OEM for Siebel
3. Informatica Metadata Reporter 6.2
0. Exit

Please select a product to install: 2

Installing Informatica PowerMart Server 6.2 OEM for Siebel

Please specify a directory to install to [/home/ga-sm/qauser7s/
informatica/pc] : /export/home/informatica/pm

/export/home/informatica/pm does not exist. create (y/n) : y

Installing module pmserver
Module pmserver extracted successfully.
Installing module pmmisc
Module pmmisc extracted successfully.
Installing module pmexes
Module pmexes extracted successfully.
Installing module pmlibs
Module pmlibs extracted successfully.
Installing module pmlocale
Module pmlocale extracted successfully.
Installing module rw
Module rw extracted successfully.
Installing module extproc
Module extproc extracted successfully.
Installing module reposhr
Module reposhr extracted successfully.
Installing module pmdoc
Module pmdoc extracted successfully.

Product specific installation is starting...

+-----+
+ Informatica Server Configuration Utility      +
+ Copyright (c) Informatica Corporation 1996-2003 +
+ All Rights Reserved.                          +
+-----+

Configuration file name [pmserver.cfg]:
ServerName[None]: SIEBEL_DW_SERVER
RepositoryName[RepositoryName]: Siebel_DW_Rep
PMUser[Administrator]:
PMPassword[*****]: Administrator
```





Installation of Informatica PowerMart Server 6.2 OEM for Siebel has completed successfully.

If you would like to configure the Informatica Server again, at the prompt enter:

```
./pmconfig
```

## Setting the Environment Variables for Informatica Server on UNIX

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

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

```
setenv PATH /export/home/informatica/pm:.${PATH}

setenv LD_LIBRARY_PATH /export/home/informatica/
pm:${LD_LIBRARY_PATH}

setenv PATH /export/home/informatica/
repositoryserver:.${PATH}

setenv LD_LIBRARY_PATH /export/home/informatica/
repositoryserver:${LD_LIBRARY_PATH}
```

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

```
setenv PATH ${PATH}:/export/home/informatica/pc:.

setenv LIBPATH ${LIBPATH}:/export/home/informatica/pc

setenv PATH ${PATH}:/export/home/informatica/
repositoryserver:.

setenv LIBPATH ${LIBPATH}:/export/home/informatica/
repositoryserver
```

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

```
setenv PATH /export/home/informatica/pm:.${PATH}

setenv SHLIB_PATH /export/home/informatica/pm:${SHLIB_PATH}

setenv PATH /export/home/informatica/
repositoryserver:.${PATH}
```

```
setenv SHLIB_PATH /export/home/informatica/  
repositoryserver:${SHLIB_PATH}
```

## **Configuring Informatica Repository from Informatica Workflow Manager on UNIX**

Use this procedure to configure the Informatica Repository from the Informatica Workflow Manager.

### **To configure the Informatica Repository**

- 1** In Informatica Workflow Manager, right-click SIEBEL\_DW\_SERVER and select Edit.
- 2** In the Host Name (Name or Dotted IP address) field, enter the name of the machine on which the Informatica Server is running.
- 3** Enter the name of the UNIX machine on which the Informatica Server is running.
- 4** In Code Page field, pick the appropriate code page, and then click OK. The Code Page for the preconfigured Informatica repository is MS Windows Latin 1 (ANSI), Super Set of Latin 1.

For example, if the informatica server is running on ENU, choose Western European, or If the Informatica Server is running on a Japanese operating system, choose MS Windows Japanese, Super Set of Shift JIS.

- 5** Click Advanced.

The Server Variables section appears.

- 6** In the \$PMRootDir, change the value to that of the path of the PowerMart server directory.

For example: `/export/home/informatica/pm)`

### Configuring the DAC Client and Server on Solaris and AIX

To configure the DAC client and server on Solaris and AIX, follow the instruction for configuring the DAC client and server on Windows in [“Setting Up the DAC Client” on page 54](#) and [“Setting Up the DAC Server” on page 83](#) as well as the instructions in this section.

The appropriate files for Solaris and AIX have .sh extension.

#### To configure the DAC client and server on Solaris and AIX

- 1 Make sure that the shell from which the DAC client or server is started has the database connectivity environment correctly setup if you are using either the Oracle OCI driver or the DB2 native driver. If you are using the SQL Server driver or the Oracle Thin driver this step is not necessary.
- 2 Make sure the information environment is set up correctly.

- For csh-shell or its related shell:

If Informatica is installed in the directory /export/home/informatica621, then on Solaris, the appropriate variables to set are the following:

```
setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:/export/home/informatica621/pm
```

```
setenv PATH ${PATH}:/export/home/informatica621/pm
```

```
setenv PM_HOME /export/home/informatica621/pm
```

On AIX, the appropriate variables to set are the following:

```
setenv LIBPATH ${LIBPATH}:/export/home/informatica621/pm
```

```
setenv PATH ${PATH}:/export/home/informatica621/pm
```

```
setenv PM_HOME /export/home/informatica621/pm
```

- If you are using bash-shell or shells with a similar syntax, then change the entry as follows:

Change:

```
setenv PATH ${PATH}/export/home/informatica/pm
```

To:

```
export PM_HOME="/export/home/informatica/pm"
```

---

**NOTE:** On Solaris, the recommended JVM is the Sun J2SDK 1.4.1 and on AIX it is IBM JDK 1.4

---

- 3** If you are installing the DAC server on AIX, do the following:
  - a** Launch the DAC client, and navigate to Setup > DAC System Properties.
  - b** Change the value of DAC Server OS to AIX.
  - c** Close the DAC client, and launch it again.
  - d** Navigate to Setup > Informatica Servers and reenter all the password fields.
  - e** Navigate to Setup > Database Connections and reenter all the password fields.

**To run the DAC server**

- For AIX, use the script named `startserver_aix.sh`.
- For Solaris, use the script named `startserver_sun.sh`.

## Starting the Informatica Repository Server on UNIX

**To start the Informatica Repository Server**

- Go to `/export/home/informatica/repositoryserver` and enter `pmrepserver`.

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

## Starting and Stopping the Informatica Server on UNIX

Use this procedure to start the Informatica Server.

### **Prerequisite**

- From the UNIX machine, make sure you can connect to the OLTP and the data warehouse database.
- The Informatica repository must have been configured using the Informatica Workflow Manager.
- The environment variables for the Informatica Server on UNIX must be set.
- The appropriate parameter files should have been copied to the SrcFiles directory.
- For Unicode, the SiebelUnicodeDB must be set. For more information, see [“Deploying Informatica and Siebel Data Warehouse in Unicode and Non-English Environments”](#) on page 40.

### **To start the Informatica Server**

- Go to `/export/home/informatica/pm`, and enter `pmserver`.

Check the log file to see if the server started. If it started correctly, the log file will state: “Server Initialization Completed.”

### **To stop the Informatica Server**

- Enter

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

For example:

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

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

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

## Updating Siebel OLTP Schema Definitions

Customers upgrading to Siebel Data Warehouse version 7.7 need to update the schema definitions in the Siebel transactional database (OLTP).

---

**NOTE:** This process requires the use of Siebel Tools.

---

This process consists of

- Backing up the repository
- Importing the .sif using Siebel Tools
- Verifying the application of the .sif

### Backing Up the Siebel Repository

Whether or not the SIF file is being applied to a development or to a production environment, back up the repository before beginning the process. To back up the repository, use the Siebel utility `repimexp` to export the repository to a .dat file.

#### **To back up a repository**

- 1 Open a DOS window and navigate to the `siebsrvr\bin` directory.
- 2 Run the command `siebenv` to set the Siebel environment variables.
- 3 Export the repository to a .dat file:

```
repimexp /a e /u username /p password /c odbc /d table_owner /r  
repository /f dat_file /v y /l logfile
```

### Importing the SIF File into the Siebel OLTP

#### **To import new schema definitions into the Siebel OLTP**

- 1 Copy the file named `SRW_76.sif` from `\dwdb\dwrep` to `\tools\objects`.
- 2 Open Siebel Tools and point to the development database.
- 3 Lock the projects named Table ETL and Table Nquire.

- 4** In the menu, click Tools > Import from Archive.
- 5** Navigate to the directory containing the SIF files and open `srw_76.sif`.  
The Import Wizard-Preview screen appears.
- 6** Click Next.
- 7** Select the option Merge the object definition from the archive file with the definition in the repository, and click Next.
- 8** The Import Wizard reviews all the object conflicts between what is in the repository and what is in the SIF file. When the Import Wizard process is done, you can expand the tree under Conflicting Objects to verify the changes being applied from the SIF file to the repository. When you are done, click Next.
- 9** A warning appears: “The operation is going to modify your repository ...” Click Yes.
- 10** When the process finishes, click Finish.

After you import the new schema into Siebel Tools, you need to apply the schema to the physical database.

#### **To apply schema definitions to the physical database**

- 1** In Siebel Tools, in the Object Explorer, click on the Table tab and query for `S_ETL_*`.  
A list of tables appears.
- 2** Select all the tables in the list and click Apply.  
The Apply Schema wizard appears.
- 3** In the Tables drop-down menu, select Current Query. Enter the appropriate connection parameters, and click Apply.

## Verifying the SIF Patch Application

After the SIF import is complete, verify that the contents of the SIF file were successfully applied to the repository.



Contents of s\_contact\_s\_etl\_ref\_param.sif:

Index	Index Columns	Project
S_CONTACT_W2	PR_PER_ADDR_ID	Table Person

## Creating Siebel Stored Procedures for DB2-UDB

The DAC uses siebstat and siebtrun stored procedures when running ETL processes. Typically, these stored procedures will be available in your Siebel OLTP; they may not be available on the data warehouse database.

**To verify the existence of these procedures, follow the steps below.**

- From a DB2 command prompt or control center, issue the following SQL scripts:

```
db2 => create table test_siebproc(id int);
DB20000I The SQL command completed successfully.

db2 => insert into test_siebproc(id) values (1);
DB20000I The SQL command completed successfully.

db2 => call siebtrun('TEST_SIEBPROC');
SQL0805N Package "NULLID.SYSSH200" was not found. SQLSTATE=51002
```

---

**NOTE:** If you get an error message, you do not have the Siebel procedures.

---

Creating stored procedures must be performed by the database administrator.

**To create DB2 stored procedures**

- 1 Refer to folder SiebelAnalytics\dwdb\siebproc. It contains compiled Siebel stored procedures for all platforms.

- 2** Copy the stored procedures into the appropriate directory on the DB2 server side, for example, d:\Program Files\SQLLIB\function directory for Windows.

If your DB2 environment is 32-bit, copy siebproc. If it is 64-bit, copy siebproc64 and rename it to siebproc.

This directory also contains the files siebproc.sql and sqlproc.ksh file, which are used to create the function.

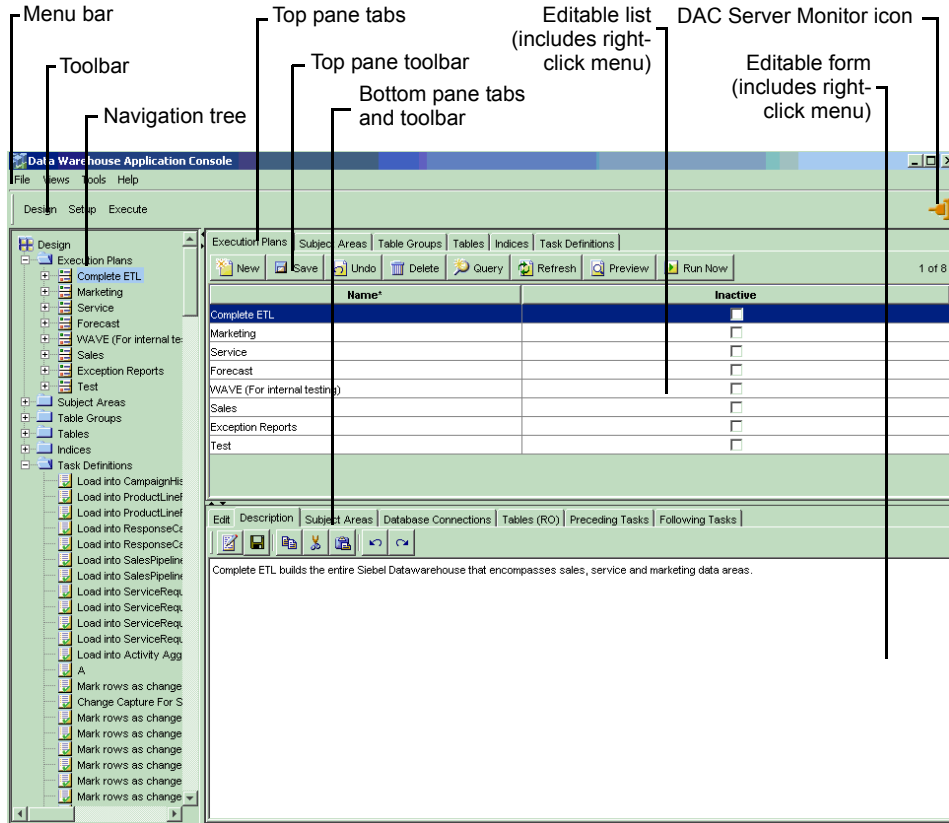
Once these procedures are created, you can verify that they exist. After the test is complete, you can drop the table TEST\_SIEBPROC.

This section provides the following topics to help you get started using the Data Warehouse Application Console (DAC):

- [“Navigating the DAC User Interface” on page 100](#)
- [“Logging In to the DAC” on page 111](#)
- [“Tailoring Execution Plans for Your Needs” on page 112](#)
- [“Using the DAC Query Functionality” on page 122](#)
- [“Executing an ETL with the DAC” on page 125](#)
- [“Handling ETL Failures with the DAC” on page 125](#)

# Navigating the DAC User Interface

Figure 6 shows the main elements of the DAC window.



**Figure 6. Example of DAC Window**

## Main Toolbar of the DAC

The main toolbar of the DAC is located directly under the menu bar. It provides access to the following views:

- **Design.** The Design view provides access to functionality related to creating and managing execution plans. For more information, see [“About the DAC Design View” on page 129.](#)
- **Setup.** The Setup View provides access to functionality related to setting up DAC system properties, Informatica servers, database connections, and email notification. For more information, see [“About the DAC Setup View” on page 150.](#)
- **Execute.** The Execute view provides access to functionality related to running, monitoring, and scheduling execution plans. For more information, see [“About the DAC Execute View” on page 152.](#)

## DAC Toolbars and Right-Click Menu Commands

Table 15 lists commands available in some of the top and bottom pane toolbars and in the right-click menus. The commands that are available depend on the tab that is active.

**Table 15. DAC Toolbar and Right-Click Menu Commands**

Command	Location in DAC	Description
New	Top and bottom pane toolbars, right-click menu	Creates a placeholder for a new record in the selected list.
Save	Top and bottom pane toolbars, right-click menu	Saves the current record.
Undo	Top and bottom pane toolbars, right-click menu	Undoes changes made to the current record after the last save.
Delete	Top and bottom pane toolbars, right-click menu	Deletes the selected record. If you delete a parent record, the child records are also deleted.
Query	Top and bottom pane toolbars, right-click menu	Opens a blank query.

**Table 15. DAC Toolbar and Right-Click Menu Commands**

<b>Command</b>	<b>Location in DAC</b>	<b>Description</b>
Refresh	Top and bottom pane toolbars, right-click menu	Retrieves the data from the repository with the last used query.
Preview	Top pane toolbar in Design view tabs: Execution Plans, Subject Areas, and Table Groups	Generates a preview of all tasks and their dependencies.
Edit	Bottom pane toolbar in Design view	Opens a dialog box in which you can add and remove objects associated with the selected repository object.
Details	Bottom pane toolbar in Execute view, Task tab	Provides a list of the task details that are executed when the selected task is run.
Run Now	Top pane toolbar in Design view, Execution Plans tab	Starts a new ETL process.
Start	Top pane toolbar and right-click menu in Execute View tabs: Current Run, Run History	Restarts the selected ETL, after the ETL has failed, stopped, or been aborted.
Stop	Top pane toolbar and right-click menu in Execute View tabs: Current Run, Run History	Stops an ETL in progress. All currently running tasks will complete, and queued tasks will stop. The status of the ETL changes to Stopped.
Abort	Top pane toolbar and right-click menu in Execute View tabs: Current Run, Run History	Causes an ETL in progress to abort. All currently running tasks will be aborted. The status of queued tasks and the ETL itself will change to Stopped.

**Table 15. DAC Toolbar and Right-Click Menu Commands**

Command	Location in DAC	Description
Auto Refresh	Top pane toolbar and right-click menu in Execute View tabs: Current Run, Run History	Allows you to turn on and off the automatic screen refresh functionality and set the refresh interval.
Copy String	Right-click menu	Copies the contents of a cell (editable and read-only) to the clipboard
Paste String	Right-click menu	Pastes a string from the clipboard into a selected cell that supports a string data type.
Copy Record	Right-click menu	Creates a copy of the selected record, with a unique record ID and (2) appended to its name. The new record is committed to the database when you click the Save button or click outside the cell. In the Design view tabs (except for the Indices tab), Copy Record copies the selected record and the record's child records. In the Design view Indices tab and Setup and Execute views, Copy Record copies only the selected record.
Output to File	Right-click menu	Outputs to a text file in the DAC root folder the contents of the current tab's record list.
Record Info	Right-click menu	Displays the record's unique ID and the timestamp for when it was last updated.
Generate Index Scripts	Right-click menu	Generates drop index, create index, and analyze table scripts for all tables that participate in the ETL process. The results are stored in the log\scripts directory.
Generate Change Capture Scripts	Right-click menu	Generates change capture scripts for tables with defined image suffixes. The scripts may include delete triggers, create and drop statements for delete triggers, and image tables and their indices.

**Table 15. DAC Toolbar and Right-Click Menu Commands**

<b>Command</b>	<b>Location in DAC</b>	<b>Description</b>
Update Rows	Right-click menu	For some columns, allows you to update the column value for each row to a single value. When you select this command, a dialog box allows you to select the column, and then a second dialog box allows you to enter a value for the column.
Import Database Tables	Right-click menu	Allows you to import table definitions from a selected database.
Import Indices	Right-click menu	Allows you to import index definitions from the database for one or more tables as listed in the result of the query.
Output Task Description	Right-click menu	Saves to an HTML file the description for a selected task definition or for all task definitions.
Run Now	Right-click menu	Starts a new ETL process.
Go To	Right-click menu	Allows you to drill down to a particular record that is associated with the selected record. Available when the Task Definitions, Indices, Tables, and Table Groups tabs are active (Design View).
Mark as Completed	Right-click menu	Changes the status of a stopped or failed ETL to Completed. In the audit trail for this ETL, the status is Marked as Completed. Use this command with caution; it can cause the data warehouse to be inconsistent.
Requeue Task	Right-click menu	Changes the status of the selected, completed task and its dependents to Queued. Available only for failed ETLs.
Get Run Information > Get Log File	Right-click menu	Fetches the log file for this run from the DAC server and saves it in the ServerLog folder.
Get Run Information > Analyze Run	Right-click menu	Saves a description of the run as an HTML file in the Log/Statistics folder.



**Table 15. DAC Toolbar and Right-Click Menu Commands**

Command	Location in DAC	Description
Get Run Information > Get Chart	Right-click menu	Displays a chart showing changes in task statuses over time in a separate window.
Get Run Information > Get Graph	Right-click menu	Displays a graph showing changes in task statuses over time in a separate window.

## Menu Bar of the DAC

The menu bar is located in the upper-left corner of the DAC window. It consists of the File, Views, Tools, and Help menus.

### File Menu

The File menu contains one command, Close, which closes the application window.

### Views Menu

The Views menu provides access to each of the views available in the DAC. The main menu options are Design, Setup, and Execute. These are the same options that are available in the toolbar.

### Tools Menu

The Tools menu provides access to various tools you can use to manage and monitor the DAC repository, DAC server, and ETL process.

Table 16 provides a description of the tools available in the Tools menu.

**Table 16. DAC Tools Menu Commands**

Tools Menu Command	Description
DAC Repository Management > Export	<p>Allows you to export all or part of the DAC metadata in order to back up the data or to reproduce the environment elsewhere. The seed data is exported in XML format and is stored in the DAC\export directory.</p> <p>In the Export dialog box, you can select the following options:</p> <ul style="list-style-type: none"> <li>■ Export All Tables.</li> <li>■ Export Categories. The Logical option exports all information contained in the Design view and database connection information. The Run Time option exports all information contained in the Execute view. The System option exports all information contained in the Setup view, except passwords for servers and database connections.</li> <li>■ Export Selected Tables. Allows you to select tables for exporting.</li> </ul>
DAC Repository Management > Import	<p>Allows you to import all or part of the DAC metadata. When you import data, any existing data in the repository is overwritten.</p> <p>If you want to import new DAC metadata while still working on the same data warehouse, you must keep the table refresh dates intact.</p> <p>In the Import dialog box, you can select the following options:</p> <ul style="list-style-type: none"> <li>■ Import All Tables.</li> <li>■ Import Categories. The Logical option imports all information contained in the Design view and database connection information. The Run Time option imports all information contained in the Execute view. The System option imports all information contained in the Setup view, except passwords for servers and database connections.</li> <li>■ Import Selected Tables. Allows you to select tables for importing.</li> </ul>

**Table 16. DAC Tools Menu Commands**

<b>Tools Menu Command</b>	<b>Description</b>
DAC Repository Management > Validate Repository	Allows you to start the repository validation process.
DAC Repository Management > Import Informatica Sessions	Allows you to import a list of Informatica workflow names and folders into a staging table in the DAC repository.
DAC Repository Management > Purge Run Details	<p>Allows you to purge completed runs from the run history. You can purge all runs (except the last run) or specify particular runs to be purged. Note: The last run cannot be purged.</p> <p>In the Purging Runs... dialog box, the following options are available:</p> <ul style="list-style-type: none"> <li>■ Entire Run History. Allows you to purge all runs and their related information. The data warehouse must be reset for this option to complete successfully.</li> <li>■ All. Purges all completed runs except for the last run.</li> <li>■ By Execution Plan. Allows you to select an execution plan whose associated runs you want to purge.</li> <li>■ By Run Name. Allows you to select an individual run for purging.</li> <li>■ Before Specified Date. Allows you to select a date before which all runs except the last run will be purged.</li> <li>■ Details Only. Purges all related information about a run but leaves the run header information.</li> </ul>
DAC Repository Management > Analyze Repository Tables	Allows you to run analyze table commands for all the DAC repository tables.
DAC Repository Management > Drop DAC Repository	Allows you to drop all the DAC repository tables. This action deletes all data in the repository.
DAC Server Management > Get Server Log	When the DAC server is running an ETL process, this command opens a text box that displays streaming data related to the process.
DAC Server Management > DAC Server Setup	Allows you to configure the DAC server connections and server email settings. This action should be performed on the machine where the DAC server is running.

**Table 16. DAC Tools Menu Commands**

<b>Tools Menu Command</b>	<b>Description</b>
ETL Management > Configure	Opens the Data Warehouse Configuration wizard, which you can use to configure an ETL process.
ETL Management > Reset Data Warehouse	Clears the refresh dates for all source and target tables. This action forces a full load to occur.
ETL Management > Preferences	Allows you to configure ETL preferences, such as start and end dates for building the day dimension, currency, language, exchange rate, prune days for disconnected users, maximum number of days an opportunity can spend in a sales stage, and the index space for Oracle databases.
Task Properties > Task Phases	Allows you to add or edit task phases.
Task Properties > Task Folders	Allows you to add or edit task folders.
UI Styles > Windows (MFC)	Changes the user interface to the Windows style.
UI Styles > UNIX (MOTIF)	Changes the user interface to the UNIX style.
UI Styles > Java (METAL)	Changes the user interface to the Java style.




## **Help Menu**

The Help Menu provides access to two dialog boxes: the Login Details dialog box, which contains login information; and the About DAC Client dialog box, which contains the DAC version number.

## DAC Server Monitor Icon

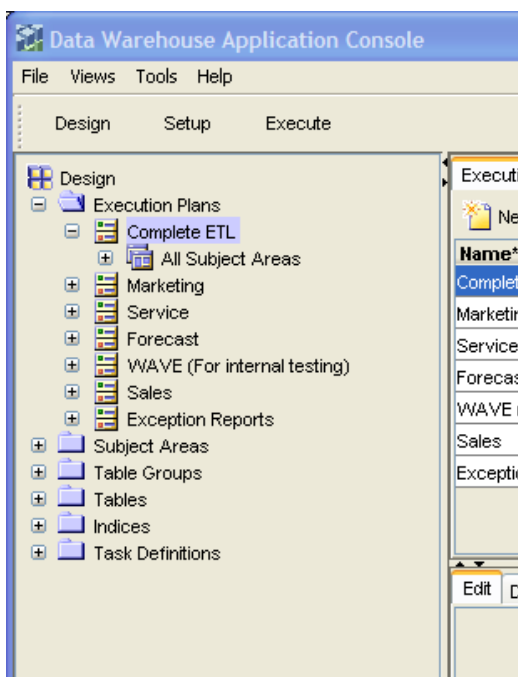
The Server Monitor is located in the upper-right corner of the DAC client. Its color and shape change based on the DAC server status. When the DAC client cannot establish a connection to the DAC server, the Server Monitor icon resembles a red electrical plug. When the client is connected to the server and the server is idle, the icon resembles an orange electrical plug in a socket. Finally, if the client is connected to a server that is running an ETL process, the icon resembles a green electrical plug with a lightning sign superimposed on it. In addition, clicking on the icon when there is a connection to the server opens a text box that displays data related to the ETL process.

**Table 17. DAC Server Monitor Icons**

Icon	Description
	DAC client cannot establish a connection to the DAC server.
	DAC client is connected to the server, and the server is idle.
	DAC client is connected to a server that is running an ETL process.

## DAC Navigation Tree

The navigation tree appears on the left side of the DAC window, as shown in [Figure 7](#). The tree root nodes correspond to the tabs in the top pane of the DAC window. When a plus sign (+) appears before a node, you can expand the node to view the records belonging to the node. You can double-click a record in the tree to have it display in the top pane in a single-record mode (New, Delete, Copy Record, and Query commands are unavailable), and double-click the root node to return to the list mode.



**Figure 7. DAC Navigation Tree**

## DAC Editable Lists

The top and bottom panes of the DAC window display records in a list format. Some of the columns in the list are editable, and others are read-only. The toolbar at the top of each pane allows you to perform various tasks associated with a selected record in the list. For a description of the toolbar commands, see [“DAC Toolbars and Right-Click Menu Commands” on page 101](#).

A right-click menu is also accessible from the lists in both the top and bottom panes. For a description of these commands, see [“Menu Bar of the DAC” on page 105](#).

The list format allows you to do the following:

- Edit the data in place and save the record by either clicking another record in the list or clicking the Save button.
- Reorder the columns.
- Sort the data in the list by clicking on the column name.
- Select predefined values from picklists.
- For fields that refer to values from other entities, use the query functionality in pop-up dialog boxes.
- Use Ctrl + C to copy an editable string to the clipboard (not available for read-only strings).
- Ctrl + V to paste a string from the clipboard into a selected cell that supports a string data type.

## Logging In to the DAC

The DAC application connects to a database called the DAC repository, which contains definitions of DAC application objects. When you log in to DAC for the first time, you need to configure the database connection to the DAC repository. You can also connect to multiple DAC repositories by configuring multiple database connections, but only one DAC repository is active at a time.

### **To log in to DAC for the first time**

- 1** Navigate to the directory where DAC is installed and double-click `startclient.bat`.
- 2** In the Login ... dialog box, select Configure.
- 3** In the Configuring ... dialog box, select Create Connection, and then click Next.
- 4** In the Connection Details area of the Configuring ... dialog box, enter the appropriate information for the database to which you want to connect.
- 5** Click Test Connection to make sure you configured the database correctly.

When you connect to a database for the first time, a DAC repository schema is created if it does not already exist.

## **Tailoring Execution Plans for Your Needs**

Execution plans are groups of subject areas that are used to execute ETL processes. For more information about the repository objects that are the components that make up execution plans, see [“About DAC Repository Objects” on page 18](#). For information about DAC functionality, see [“DAC Functional Reference” on page 129](#). For detailed information about customizing execution plans, see [“Customizing the Siebel Data Warehouse” on page 161](#).

In designing an execution plan, you should consider the following questions:

- **Tables.** Which tables need to be populated for the data warehouse. From which tables does your organization source data?
- **Indices.** Do the target tables have the correct indices defined?
- **Tasks.** Are the tasks that load this table defined?
- **Dependencies.** What does a task require to be run?
- **Table groups.** Do table groups need to be built?
- **Subject areas.** Do the subject areas cover all the relevant table groups?



- **Execution plans.** Which subject areas need to be refreshed and how often and at what time frequency?

When an execution plan is run, the DAC server uses the following logic to identify the tasks that need to be run and their order of execution:

**1** Initial selection of tables.

- a** Find all the subject areas.
- b** Find all the table groups that belong to the subject areas.
- c** Find all the tables that belong to the table groups.

**2** Recursive selection of related tables.

Recursively find all the table groups and their tables with the above list of tables. For a description of this process, see [“About the DAC Table Groups Tab” on page 135](#).

**3** Initial selection of tasks.

Find all the tasks that load into the tables selected above, that is, tasks whose target table is one of the selected tables (the tasks that are non-optional).

**4** Recursive selection of all tasks.

- a** Depending on the non-optional dependency rules, recursively figure out the prerequisite tasks.
- b** Depending on the optional dependency rules, figure out the optional tasks.

**5** Filter tasks based on the database connections that will participate in the ETL.

**6** Selection of dependencies.

Load dependencies to identify the execution order.

**7** Identify the task details.

Iterate through all the tasks selected and compute the task details based on whether the target table is loaded in full mode or incremental mode.

**8** Identify the list of source tables for change capture, create tasks for change capture, and sync for Siebel source tables.

In the DAC, the Design view allows you to view and modify the relationships among the repository objects discussed above. For more information about the Design view, see [“About the DAC Design View” on page 129](#).

### Pruning Execution Plans in the DAC

The Siebel Data Warehouse includes preconfigured execution plans. The Complete ETL execution plan contains a single subject area, named All Subject Areas, which includes all the table groups of the star type. There are also preconfigured execution plans for marketing, sales, forecast, and service. If none of the preconfigured execution plans meet the needs of your organization, you can define your own subject areas and build stars as needed. You can also modify an existing execution plan to meet your needs by inactivating its child objects.

### Pruning Subject Areas in the DAC

A subject area is a group of table groups (stars). For example, the subject area named Sales Subject Area may consist of the following stars: Order, Opportunity, and Quote. Including a subject area in an execution plan means that all the tables that belong to the table groups and the related table groups will be populated during the ETL process.

You can add and remove table groups from the preconfigured subject areas, or you can remove a table group from a subject area by *inactivating* the table group.

You can also create a new subject area by associating it with the appropriate table groups. You can create a subject area with any kind of table group, but it is recommended that you use table groups of the star type. A star usually consists of a fact table, dimensions, and aggregates (if any).

### Pruning Table Groups in the DAC

A table group is a logical grouping of tables that defines a logical relationship among all the component tables. The table group definition is based on a main table that drives the relationship between itself and the other tables in the group.

A table can be a main table for only one table group. When you design a table group, you associate it with a list of tables that need to be populated whenever the main table participates in an ETL. The tables can be main tables for other table groups; in such cases, when an ETL is run, the DAC server recursively includes all necessary tables. This process in which the tables are recursively populated is called *chaining*.

You can prune a table group by inactivating one or more of the tables associated with it. When designing an execution plan, inspect the table groups that you are interested in. If there are dimensions in a fact table that you are not interested in populating during the ETL process, inactivate them or remove the table association. This might also require you to modify the workflows and dependencies as well. Use the Preview functionality to figure out if your design is correct at the table group, subject area, and execution plan level.

---

**CAUTION:** If a table is removed from one table group but is still active in others that belong to the execution plan, you will see it appearing in the preview. If the table is not required for the entire ETL, consider inactivating the table. For more information about inactivating tables, see [“The Chaining Process” on page 115](#).

---

## The Chaining Process

When multiple target (data warehouse) tables share the same source table, they must be populated together. For example, the Person Fact table group contains the tables listed below. The main table for this table group is W\_PERSON\_F. When W\_PERSON\_F participates in an ETL, the other tables listed below also participate. W\_PERSON\_D is the main table for the Person Dimension table group; whenever it participates in an ETL, the other tables that are part of the Person Dimension table group also participate in the ETL.

Name	Main Table for Group	Relationship Type
W_GEO_D	GEO DIMENSION GORUP	Dimension
W_PERSON_F	PERSON FACT GROUP	Self
W_PERSON_D	PERSON DIMENSION GROUP	Dimension
W_ORG_D	ORGANIZATION DIMENSION GROUP	Dimension
W_DAY_D	DAY DIMENSION GROUP	Dimension

W\_INDUSTRY\_D                      INDUSTRY IDMENSION GROUP                      Dimension

A similar kind of chaining concept is used for entities that have parent-child relationships. When a parent table participates in an ETL, its child tables also get populated. For example, the Order Dimension and Order Item Fact tables have a parent-child relationship. When the Order Dimension table is part of a table group that participates in an ETL, the Order Item Fact table will also be populated.

Therefore, when pruning the data warehouse, you should start with the table groups of type star and verify that all the related tables are required. If a table is not required and it is a parent table, you need to disable the link between it and the child tables that you do not want to populate. For information about disabling entities, see [“Disabling DAC Repository Entities” on page 117](#).

The chaining process is used for the following reasons:

- To maintain data integrity

For example, when the Person Fact table group is built, the Person Dimension group is also populated.

- Two or more data warehouse tables can share the same source table

When multiple data warehouse tables (target tables) share the same source table, they must be populated together. For example, the tables W\_PERSON\_F and W\_PERSON\_D have the same main source table, S\_CONTACT. Suppose you want to populate the Revenue Fact table group, which has W\_REVN\_F as its main table and W\_PERSON\_D as one of the dimension tables. Whenever the Revenue Fact table group is chosen for an ETL execution, W\_PERSON\_F is also populated because W\_PERSON\_F is chained to W\_PERSON\_D.

- To maintain parent-child relationships

For example, when the Order Dimension table is built, the Order Item Fact table also must be populated, otherwise the data will not be in sync.

## Pruning Tasks and Dependencies in the DAC

A task is a command that is used to populate a certain table or tables. Tasks can have prerequisite tasks that need to be executed prior to its own execution. These are defined as dependency rules. There are mandatory and optional dependencies. For example, Load into Activity Fact has a mandatory dependency on Extract into Activity Fact, but has optional dependency on all the tasks that load the dimensions. This is because loading of the Activity Fact table cannot be accomplished until its extract is done. However, if the dimension tables are not absolutely necessary, that is, if part of the pruning process eliminates a certain table from the initial list of tables selected, then the load task will not wait for that optional tasks.

## Disabling DAC Repository Entities

You can disable entities at different levels for the sake of pruning.

- If certain tables are not required by a table group, you can remove the tables from the table group or inactivate them. You might also need to review and modify the Informatica workflows.
- If a task that is loading/updating a certain table is deemed not necessary, you can inactivate it.
- If a task is depending on another task, and is deemed not necessary, you can remove the dependency or inactivate the link.
- If a table is not required in the whole of the ETL process, you can disable the table, and none of the tasks that write to it will be chosen for execution.

Once the dependencies are correctly defined, the DAC server will execute them in proper order during ETL execution.

## Using the DAC Preview Functionality to View Tasks and Dependencies

You can verify the design of an execution plan using the Preview functionality, which is accessible from the top pane toolbars of the Execution Plans, Subject Areas, and Table Groups tabs, and in the right-click menu on these tabs.

The Preview functionality generates a preview of all tasks and their dependencies using the DAC server logic described in [“Tailoring Execution Plans for Your Needs” on page 112](#). This process reviews an execution plan’s associated subject areas, table groups, and tables to identify tasks and their dependencies.

The preview shows the approximate order in which tasks will be executed, and a list of source tables, target tables, and tables for which an image will be built. Only the tasks that source or write to database connections as defined in the execution plan will be included. The preview also shows a detailed analysis of the dependencies. By reviewing the preview, you can validate the design of your execution plan without having to run the ETL.

The preview has three parts. The first part is titled, “Initial Selection of Tables going through the related Table Groups.” This part shows the selection of tables to be populated by table group and is created by iterating through the table groups. The selection process starts with the initial, distinct list of all the table groups (iteration 0) and goes through the related table groups (iteration 1, 2, and so on), selecting tables that are not already on the list of chosen tables.

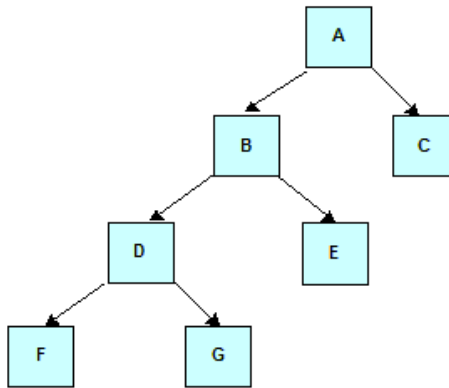
The second part of the preview is titled, “Task Selection Analysis.” This part shows the task program. It starts with the tasks that write to any of the tables selected during the table selection process. The DAC server logic goes through the dependency definitions, selecting all the mandatory (non-optional) tasks and then selecting all the optional tasks. Eventually, the dependencies define the order of execution.

The third part of the preview is titled, “Step Dependency Analysis.” This part shows, for all given tasks, the following information:

- **Depends On.** The immediate tasks that a given task depends on.
- **Dependents.** The tasks that are dependent on a given task.
- **Number of All Depends.** All the Depends On tasks.
- **Number of All Dependents.** All the Dependent tasks.

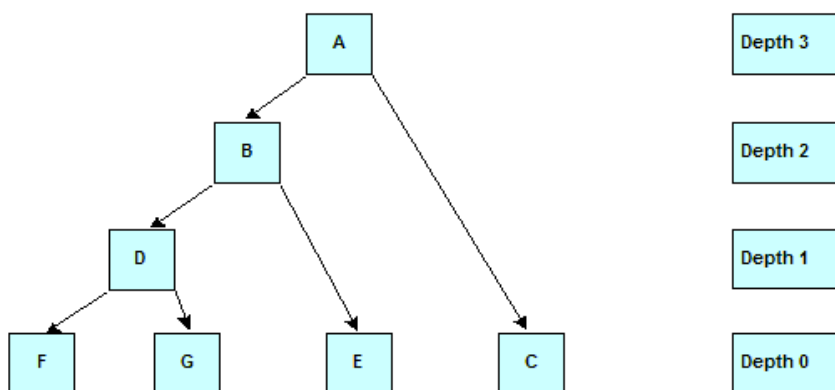
This part of the preview is useful for understanding how a task relates to all other tasks, which can be helpful for debugging.

Figure 8 shows the dependencies of various tasks: task A depends on tasks B and C, task B depends on tasks D and E, and task D depends on tasks F and G.



**Figure 8. Task Dependencies**

In [Figure 9](#) the tasks are shown as grouped by the DAC server based on the level of dependency. (The level of dependency is referred to as a *depth*.) Tasks that have no dependencies are depth 0. Tasks that depend on other tasks of depth 0 are depth 1. Tasks that depend on other tasks of depth 1 or 0 are depth 2, and so on.



**Figure 9. Task Dependencies by Depth**

Because the DAC server dynamically prioritizes the tasks that are eligible to run based on the number of resources available, it uses the information generated by the preview to decide which tasks to execute. When more than one task is eligible to be run, a series of prioritization rules is applied.

## Distributing DAC Metadata

Typically, you may have multiple environments, such as development, QA, production, and so on. When you make changes to the development environment, you test it, and then deliver it, exporting the whole environment and distributing it to the other environments. The data is exported as XML files, which are stored in the DAC\export directory on the client machine where the export is done.

To apply changes from the development environment to any other, you copy all of the XML files into the DAC\export folder and then import the data.



**To distribute DAC metadata from the development environment to another environment**

- 1 In the DAC, select Tools > DAC Repository Management > Export.

The Export dialog box appears.

- 2 Select one of the following options:

- **Export All Tables.** Select this option to duplicate the entire environment. All tables in the DAC repository and the information in the Design view associated with them are exported as XML files and stored in the DAC\export folder. Exercise caution when selecting this option, because when you import this data, it replaces all run-time information that exists on the database where the import is performed.
- **Export Categories.** Select this option to export part of the data from the development environment. The data is stored as XML files in the DAC\export folder. Select one of the following categories:
  - **Logical.** Exports all information contained in the Design view and the database connection information. This option preserves all the run data in the new environment, including refresh dates.
  - **Run Time.** Exports the run-time information but preserves all the logical definitions in the new environment. You may seldom have to select this option.
  - **System.** Exports system preferences, ETL preferences, and email recipient information. Passwords for servers and database connections are not exported.

- 3 Copy the XML files from the DAC\export folder in the development environment to the DAC\export folder on the local machine in the new environment.

- 4 In the DAC, in the new environment, select Tools > DAC Repository Management > Import.

The Import dialog box appears.

- 5 Select the import option that corresponds to the export option you selected in step 2. The following options are available:

- **Import All Tables.** For security reasons, this option does not populate the password fields. Make sure you update the passwords before attempting to run an ETL process.
- **Import Categories.** Select one of the following categories:
  - Logical
  - Run Time
  - System

## Using the DAC Query Functionality

Querying is a way to locate one or more records that meet your specified criteria. Query functionality is available in every DAC screen. When you enter query mode, the Edit and Description child tabs in the bottom pane are not available.

## DAC Query Commands and Operators

You can use the query commands and operators in [Table 18](#) to define your query criteria.

**Table 18. DAC Query Commands and Operators**

Operator	Description
=	Placed before a value, returns records containing a value equal to the query value.
<	Placed before a value, returns records containing a value less than the query value.
>	Placed before a value, returns records containing a value greater than the query value.
< >	Placed before a value, returns records containing a value that is not equal to the query value.
< =	Placed before a value, returns records containing a value less than or equal to the query value.

**Table 18. DAC Query Commands and Operators**

Operator	Description
> =	Placed before a value, returns records containing a value greater than or equal to the query value.
*	Wildcard that can be placed in the middle, or at the beginning or end of a text string.
!	Used for negation.
""	Surrounds a string that, unless modified by a wildcard, must be matched exactly.
\	Escape symbol is used when double quotes should not be processed as a special symbol. For example, <i>!( " *null text" or( \ " * \ " ) )</i> is a value expression for a text field. The query returns values that do not end with a string <i>null text</i> and that are not surrounded by double quotes.
()	Surrounds the values and operators that will be processed first.
NULL	Returns records for which the query field is blank. (Not case sensitive.)
AND	Placed between values, returns only records for which all the given conditions are true. (Not case sensitive.)
OR	Placed between values, returns records for which at least one condition is true. (Not case sensitive.)

## DAC Query Examples

The following examples show different ways you can query on the Name column of the Tasks tab.

- *Extract\** lists all tasks whose name starts with *Extract*.
- *\*Extract\** lists all tasks whose name contains the word *Extract*.
- *!Extract\** lists all tasks whose name does not start with the word *Extract*.
- *!null* lists all tasks whose name is not null.

- *Extract\** or *Aggregate\** lists all tasks whose name starts with *Extract* or *Aggregate*.
- *Load\** and *\*Aggregate\** lists all tasks whose name starts with *Load* and also contains the word *Aggregate*.
- "*Extract for Wave Dimension*" or "*Load into Wave Dimension*" lists tasks whose name is either *Extract for Wave Dimension* or *Load into Wave Dimension*.

---

**NOTE:** When using spaces within strings, you need to surround the string with quotes ("").

---

## Common DAC Query Procedures

This section includes instructions for common query procedures.

### **To create and execute a query in the DAC**

- 1** In the top or bottom pane of the DAC, click Query on the toolbar or in right-click menu.

A blank row in a list appears.

- 2** Enter the query criteria in the appropriate fields.
- 3** Click Run Query on the toolbar.

The query is executed and the records appear.

### **To enter a query value in a date field**

- 1** In the date field, click the calendar icon on the right side of the cell.

The Date dialog box appears.

- 2** Enter the date and time for which you want to search, and select the appropriate query condition.

You can use the query commands and operators listed in ["DAC Query Commands and Operators."](#)

## Executing an ETL with the DAC

Execution plans are subject areas that are used to execute ETL processes. Before you attempt to run an ETL, do the following:

- Set ETL preferences. For more information, see [“Setting DAC ETL Preferences” on page 73](#).
- Set database connections to the OLTP and the data warehouse. For more information, see [“Creating Database Connections” on page 80](#).
- Register the Informatica Servers. For more information see, [“Registering Informatica Servers” on page 79](#).

### **To execute an ETL with the DAC**

- 1 Launch the DAC client by double-clicking the startclient.bat file in the DAC directory or by running startclient from a command prompt.
- 2 Navigate to the Execution Plans tab in the Design view.
- 3 Select the execution plan you want to execute.
- 4 Click Run Now.

Once the ETL starts running you can monitor its progress in the Current Run tab of the Execute view. For more information about monitoring an ETL, see [“About the DAC Current Run Tab” on page 152](#) and [“About the DAC Run History Tab” on page 156](#).

## Handling ETL Failures with the DAC

This section includes the following topics:

- [“When the Execution of an Execution Plan Fails” on page 126](#)
- [“In Case of Abnormal Termination of the DAC Server” on page 126](#)
- [“Discarding the Current Run Execution Plan” on page 127](#)
- [“Failure of Aggregator Transformation Tasks with Sorted Input” on page 127](#)

- [“Performing a Full Load Again” on page 127](#)

### When the Execution of an Execution Plan Fails

When an execution plan is executed, if a task fails, the status of the tasks that are dependent on the failed task is changed to Stopped. While tasks are still running, the execution plan’s status is Running. When all the tasks have been run, and if one or more tasks have failed, the execution plan’s status is changed to Failed.

You can check the tasks that have failed in the Current Run tab of the Execute view, fix the problems, and then requeue the failed tasks by changing the status to Queued. You can then restart the ETL. All the tasks will then be rerun. You can also manually run a task, change its status to Completed, and then restart the ETL. Tasks with a Completed status are skipped.

---

**CAUTION:** The DAC server does not validate tasks that have been run manually.

---

To restart a Failed ETL, click Run Now from the Current Run tab of the Execute view.

### In Case of Abnormal Termination of the DAC Server

If the DAC server fails during the execution of the ETL, the status of the ETL execution will remain as Running. When the DAC server is started again, it will automatically run the ETL if the Auto Start ETL system preference is set to True. If the same system preference is set to False, when the server restarts, it will set the correct status as Failed. In order to execute the ETL from the point of failure, submit the request to the server again.

The DAC server will automatically terminate if it loses connection to the DAC repository.

## Discarding the Current Run Execution Plan

You can discard an execution plan that failed by navigating to the Current Run tab, right-clicking on the execution plan and changing its status to Mark as Completed. This will force the run status to be updated as Completed. When you submit a request for another run, the DAC server creates another instance of it.

---

**CAUTION:** Perform this procedure in a development or testing environment only, since it might leave the data in an inconsistent state, causing you to have to reload all of the data.

---

## Failure of Aggregator Transformation Tasks with Sorted Input

Tasks that use Informatica Aggregator transformation can fail when the Sorted Input option is active. The tasks SDE\_DTLFORECASTFACT and SDE\_COSTLIST are examples of tasks that can fail in such a situation.

To prevent such tasks from failing, in Informatica Designer, navigate to Mapping Designer, open the corresponding mapping, and in the Aggregator transformation, remove the check from the Sorted Input check box.

## Performing a Full Load Again

In test environments, after running several ETLs, you might want to perform a full load again. You do this by accessing Tools > ETL Management > Reset Data Warehouse.

---

**CAUTION:** This procedure will cause the DAC server to perform a full load the next time an ETL is run, which will truncate data from the target tables. Therefore, this action should be used only in development and testing environments. You should not perform this procedure in a production environment unless you plan to rebuild the data warehouse in its entirety.

---

## **DAC Quick Start**

---

*Handling ETL Failures with the DAC*



This section describes the functionality available in the Siebel Data Warehouse Application Console (DAC). The DAC functionality is grouped based on three views: Design, Setup, and Execute. These views are accessible from the DAC toolbar.

## About the DAC Design View

The DAC Design view provides access to functionality related to creating and managing execution plans. The top pane of the Design view contains tabs, each of which represents a component of the execution plan. The bottom pane of the Design view contains subtabs that relate to the tab selected in the top pane. The navigation tree allows you to see the hierarchical relationship among all the DAC repository objects.

This section includes the following topics:

- [“About the DAC Execution Plans Tab” on page 130](#)
- [“About the DAC Subject Areas Tab” on page 134](#)
- [“About the DAC Table Groups Tab” on page 135](#)
- [“About the DAC Tables Tab” on page 136](#)
- [“About the DAC Indices Tab” on page 139](#)
- [“About the DAC Task Definitions Tab” on page 142](#)
- [“Using SQL Files as an Execution Type in the DAC” on page 148](#)

### About the DAC Execution Plans Tab

The Execution Plans tab displays a list of execution plans in the top pane. This list includes the execution plan name and an Inactive column, which indicates whether the execution plan is active or inactive. Inactive execution plans do not participate in the ETL process.

Table 19 provides a description of the subtabs in the bottom pane.

**Table 19. DAC Execution Plans Subtabs**

Tab	Description
Edit	Allows you to edit the execution plan selected in the top pane.
Description	Displays a description of the execution plan selected in the top pane.
Subject Areas	Displays a list of subject areas associated with the execution plan selected in the top pane.
Database Connections	Displays a list of database connections associated with the execution plan selected in the top pane. In order for a task to participate in the ETL process, the task's source and target must match the list of database connections displayed here. You can inactivate a database connection by selecting the Inactive check box. If you inactivate a database connection, all tasks that read or write to it are eliminated.
Tables (Read Only)	<p>Displays a read-only list of the initial tables that participate in an ETL. For a complete list of tables participating in the ETL, use the Preview functionality.</p> <p>The list of tables contains the following column headings:</p> <ul style="list-style-type: none"> <li>■ <b>Name.</b> Name of the target table.</li> <li>■ <b>Main Table for Group.</b> If the table is a main table for a table group, the group name is displayed in this column.</li> <li>■ <b>Table Type.</b> Table type as defined in the table properties.</li> </ul>

**Table 19. DAC Execution Plans Subtabs**

Tab	Description
Preceding Tasks	<p>Displays a list of tasks that must be completed before an ETL is executed.</p> <p>The list of tasks contains the following column headings:</p> <ul style="list-style-type: none"> <li>■ <b>Name.</b> Name of the task.</li> <li>■ <b>Priority.</b> Indicates the order in which the task runs. If two or more tasks have the same priority, the order occurs randomly.</li> <li>■ <b>Inactive.</b> Allows you to inactivate a task by selecting the check box.</li> <li>■ <b>Command.</b> Command associated with the task.</li> </ul>
Following Tasks	<p>Displays a list of tasks that must be completed after an ETL is executed.</p> <p>The list of tasks contains the following column headings:</p> <ul style="list-style-type: none"> <li>■ <b>Name.</b> Name of the task.</li> <li>■ <b>Priority.</b> Indicates the order in which the task runs. If two or more tasks have the same priority, the order occurs randomly.</li> <li>■ <b>Inactive.</b> Allows you to inactivate a task by selecting the check box.</li> <li>■ <b>Command.</b> Command associated with the task.</li> </ul>

### **Actions Available**

The following actions are available in the top pane toolbar and in the right-click menu when the Executions Plan tab is active.

#### **Preview Command**

This command generates a preview of all tasks and their dependencies by reviewing the subject areas, table groups, tables, and tasks. Information provided includes the approximate order in which tasks are executed, a list of source tables, target tables, and tables for which an image will be built. Only the tasks that source or write to database connections as defined in the execution plan are included. For more information about the Preview functionality, see [“Using the DAC Preview Functionality to View Tasks and Dependencies”](#) on page 117.

### Copy Record

This command creates a copy of the selected execution plan and also copies the references to subject areas and database connections.

### Generate Index Scripts Command

This command generates drop index, create index, and analyze table scripts for all the tables that participate in the ETL process. The results are stored in the log\scripts directory. The index scripts are separated by ETL indices and query indices. The DAC server automatically drops and creates indices and analyzes the tables. This option can be turned off by setting the system preferences. It is recommended that this option be turned off only for DB2-390 environments.

### Run Now Command

This command submits a request to the DAC server to execute the execution plan. Inactive execution plans are not included in the process.

When an execution plan is run, the following logic is used to identify the tasks that need to be run and their order of execution.

- 1** Check on the previous run status.
  - a** If the previous run executed a different execution plan from the one currently submitted (either manually or through the scheduler), abort the run.
  - b** If the previous run executed is the same as the current execution plan and did not complete, continue from where it left off.
  - c** If the previous run has completed, create a new run.
- 2** Check resources.
  - a** If the DAC server cannot establish connections to the databases defined for that execution plan, abort the run.
  - b** If there are no Informatica servers registered, abort the run.
  - c** If the Informatica servers are registered and the DAC server is unable to ping them, abort the run.
- 3** Select initial tables.
  - a** Find all the subject areas.

- b** Find all the table groups that belong the subject areas.
- 4** Recursively select related tables.
  - a** Recursively find all the table groups and their tables with the above list of tables.
- 5** Select initial tasks.
  - a** Find all the tasks that load into the above selected tables. (Tasks whose target table is one of the selected tables and that are non-optional.)
- 6** Recursively select all tasks.
  - a** Depending on the non-optional dependency rules, figure out the prerequisite tasks recursively.
- 7** Select optional tasks.
  - a** With the tasks already chosen, recursively find the optional tasks.
- 8** Select dependencies.
  - a** Load dependencies and compute for each task the number of all Depends On tasks. This number will be decremented for tasks higher up in the hierarchy as each of the tasks complete. When the number of Depends On tasks reaches zero, the task itself will become eligible to be run.
- 9** Identify the task details.
  - a** Iterate through all the tasks selected, compute the task dependencies.
- 10** Identify the list of source tables for change capture and create tasks for change capture and sync for Siebel sources.
  - a** Iterate through all the tasks and find the primary or auxiliary tables and group them by source database connection.
- 11** Execute pre-ETL tasks. If there are errors, stop the run.
- 12** Execute the change capture process. If there are errors, stop the process.
- 13** Execute the dependency graph. If there are errors, stop all the tasks that depend on the failed tasks. Continue to execute all others, until there are no more to run, and then stop.

**14** Execute the change capture sync process. If there are errors, stop the process.

**15** Execute the post-ETL process. If there are errors, stop the process.

## About the DAC Subject Areas Tab

A subject area is a logical grouping of table groups that is related to a particular subject or application context.

The Subject Areas tab displays a list of subject areas in the top pane. This list includes the subject area name and an Inactive column, which indicates whether the subject area is active or inactive. Inactive subject areas do not participate in the ETL process.

[Table 20](#) provides a description of the subtabs in the bottom pane.

**Table 20. DAC Subject Areas Subtabs**

Tab	Description
Edit	Allows you to edit the subject area selected in the top pane.
Description	Displays a description of the subject area selected in the top pane.
Table Groups	Displays a list of table groups associated with the subject area selected in the top pane. You can inactivate a table group by selecting the Inactive check box.
Tables (Read Only)	<p>Displays a read-only list of the initial tables that participate in an ETL. For a complete list of tables participating in the ETL, use the Preview functionality.</p> <p>The list of tables contains the following column headings:</p> <ul style="list-style-type: none"><li>■ <b>Name.</b> Name of the target table.</li><li>■ <b>Main Table for Group.</b> If the table is a main table for a table group, the group name is displayed in this column.</li><li>■ <b>Table Type.</b> Table type as defined in the table properties.</li></ul>

### Actions Available

The following actions are available in the top pane toolbar and in the right-click menu when the Subject Areas tab is active.

**Preview Command**

This command generates a preview of all tasks and their dependencies by reviewing the table groups, tables, and tasks. Information provided includes the approximate order in which tasks are executed, a list of source tables, target tables, and tables for which an image will be built. The preview will include all tasks regardless of the source or target. Unlike the preview generated from the Execution Plans tab, this preview includes all the tasks, regardless of the source or target.

**Copy Record**

This command creates a copy of the selected subject area and also copies the references to table groups.

**About the DAC Table Groups Tab**

The Table Groups tab displays a list of table groups in the top pane. This list includes the name and type of the table group, the main table of the table group, and an Inactive column, which indicates whether the table group is active or inactive. Inactive table groups do not participate in the ETL process.

[Table 21](#) provides a description of the tabs in the bottom pane.

**Table 21. DAC Table Groups Subtab**

Tab	Description
Edit	Allows you to edit the table group selected in the top pane.
Description	Displays a description of the table group selected in the top pane.
Tables	<p>Displays a list of the initial tables that participate in an ETL. For a complete list of tables participating in the ETL, use the Preview functionality.</p> <p>The list of tables contains the following column headings:</p> <ul style="list-style-type: none"> <li>■ <b>Name.</b> Name of the target table.</li> <li>■ <b>Main Table for Group.</b> A read-only column showing if the table is a main table for another table group.</li> <li>■ <b>Relationship Type.</b> Relationship type describing how the table relates to the main table. For documentation purposes only.</li> <li>■ <b>Inactive.</b> Indicates whether the table’s relationship to the table group is active or inactive.</li> </ul>

### Actions Available

The following actions are available in the top pane toolbar and in the right-click menu when the Table Groups tab is active.

#### Copy Record

This command creates a copy of the selected table group and also copies the references to tables.

#### Preview Command

This command generates a preview of all tasks and their dependencies by reviewing the tables and tasks. Information provided includes the approximate order in which tasks are executed, a list of source tables, target tables, and tables for which an image will be built. Unlike the preview generated from the Execution Plans tab, this preview includes all the tasks, regardless of the source or target.

## About the DAC Tables Tab

The Tables tab displays a list of tables in the top pane. This list includes the following information shown in [Table 22](#).

**Table 22. DAC Tables Tab List Columns**

Column	Description
Name	Name of table.
Table Type	Indicates the type of target table. There is no differentiation for source tables.
Image Suffix	Applicable to Siebel source tables only. For more information, see <a href="#">“Change Capture Scripts Command” on page 138</a> .
Inactive	Indicates whether a table is active or inactive. Inactive tables do not participate in the ETL process.



Table 23 provides a description of the tabs in the bottom pane.

**Table 23. DAC Tables Subtabs**

Tab	Description
Edit	Allows you to edit the table selected in the top pane.
Description	Displays a description of the table selected in the top pane.
Database Connections	Displays a list of data sources associated with the table selected in the top pane. The Edit button allows you to add or remove data sources.
Indices	Displays a read-only list of indices that belong to the table selected in the top pane.
Source for Tasks	Displays a read-only list of tasks that use the table selected in the top pane as a source.
Target for Tasks	Displays a read-only list of tasks that use the table selected in the top pane as a target.

### Actions Available

The following actions are available in the top pane toolbar and in the right-click menu when the Tables tab is active.

#### Copy Record

This command creates a copy of the selected table and also copies the references to database connections.

#### Generate Index Scripts Command

This command generates drop index, create index, and analyze table scripts for all the tables that participate in the ETL process. You can query for one or more tables, and the command will generate all the scripts for all the tables. The results are stored in the log\scripts directory. Files are generated that provide the following information: drop indices of type ETL, create indices of type ETL, drop indices of type Query, create indices of type Query, and analyze tables.

### Change Capture Scripts Command

This command generates change capture scripts for tables with defined image suffixes. The scripts include creation of the image tables to assist in the change capture process and their indices. You can also create the triggers for deletes. However, you should do so with caution because preconfigured tables for the change capture are created by applying SIF files on the Siebel OLTP. Triggers for deletes should be used only for new source tables for which you want to use change capture. The scripts will be generated for the selected table or all the tables shown in the list applet. You can narrow down the tables of interest by querying. The results are shown in a text box, which can be supplied to the DBA for executing.

### Import Database Tables Command

This command allows you to import table definitions from a selected database. You will be prompted to choose a database connection from a list that has already been defined in the DAC repository. When this command is executed, a list of tables in the data source is shown in a list applet. Check the Import flag in the list applet to select the tables that you want to import. The tables are then brought into the DAC repository. The default table type associated with the newly imported tables is Other. After you import the tables, change the table type to the appropriate type and associate the table with a database connection.

### Import Indices Command

This command allows you to import index definitions from the database for one or more tables as listed in the result of the query. You will be prompted to select a data source. The client connects to the database and retrieves the index definitions for the table. The list includes new indices that may have been created in the database, indices whose definitions are modified in the database, or indices that do not exist in the database. To accept the changes, check the Merge? check box. When you select the list to be merged, one of the following actions will take place: If an index definition does not exist in the DAC repository, it will be created; if an index definition exists but the definition is the same, the new definition will be brought in; or if an index as defined in the repository does not exist in the database, it will be inactivated.

---

**NOTE:** In MSSQL Server databases, when you import indices with columns whose sort order type is “Desc,” the sort order type appears in the DAC as “Asc.” You have to set the sort order manually to “Desc.”

---

## About the DAC Indices Tab

The Indices tab displays a list of all the indices of all the data warehouse tables. It is recommended that you do not register any indices for source tables. During the ETL process, when a table is going to be truncated, all the indices as defined in the repository will be dropped before the data is loaded and will be created after the data is loaded automatically. While this improves the ETL performance, the preconfigured workflows have the bulk load option turned on. The bulk load will fail if there are indices on the table. Therefore, it is important to keep the index definitions in sync with the database. For example, if you create an index on the database, and it is not registered in the repository, the index will not be dropped and the load will fail.

[Table 24](#) provides a description of the index properties displayed in the list.

**Table 24. DAC Indices Tab List Columns**

Column	Description
Name	Unique name with which the index is created.
Table Name	Table for which an index is created.
Index Type	Type of index: ETL or Query. An ETL index is typically used during the ETL process. A Query index is an index is used only during the reporting process. It is recommended that you have a clear understanding of when and where the index will be used at the time of registering the index.
# Unique Columns	For unique indices, the number of columns that will be unique.
Is Unique	Indicates whether the index is unique.
Is Clustered	Indicates whether the index is clustered. There can be only one clustered index per table.
Is Bitmap	Indicates whether the index is of the bitmap type.
Allow Reverse Scan	Applicable only for DB2-UDB databases. The index will be created with the Allow Reverse Scan option.

**Table 24. DAC Indices Tab List Columns**

Column	Description
Table Space Name	Applicable only for Oracle databases. If a table space is mentioned, an index is created in that particular table space. If all indices go to a table space different from the data area, you can define it in the ETL Preferences (Tools > ETL Management > Preferences).
Inactive	Indicates whether an index is active or inactive. Inactive indices do not participate in the ETL process.

Table 25 shows which index properties are available for the different database types that are supported.

**Table 25. Index Properties and Database Types Matrix**

Column/ Database	Is Unique	Is Clustered	Is Bitmap	Allow Reverse Scan	# Unique Columns
Oracle	X		X		
DB2	X	X		X	X
DB2-390	X				
MSSQL	X	X			

Table 26 provides a description of the tabs in the bottom pane.

**Table 26. Indices Subtabs**

Tab	Description
Edit	Allows you to edit the table selected in the top pane.
Description	Displays a description of the table selected in the top pane.

**Table 26. Indices Subtabs**

Tab	Description
Columns	Displays the list of columns the index is made of. The list of columns contains the following column headings: <ul style="list-style-type: none"><li>■ <b>Name.</b> Name of the column.</li><li>■ <b>Position.</b> Position of the column in the index.</li><li>■ <b>Sort Order.</b> Indicates whether the sort order is ascending or descending.</li><li>■ <b>Inactive.</b> Indicates whether the column is active or inactive.</li></ul>
Databases	Indicates whether an index is applicable only to a certain type of database. If no database is indicated, the index will always be created.

### Actions Available

The following actions are available in the top pane toolbar and in the right-click menu when the Indices tab is active.

#### Copy Record

This command creates a copy of the selected index.

## About the DAC Task Definitions Tab

The Task Definitions tab displays a list of tasks in the top pane. [Table 27](#) provides a description of the task properties displayed in the list.

**Table 27. Task Definitions List Columns**

Column	Description
Name	A logical, unique name for the task.
Command for Incremental Load	<p>A table can be loaded in Full Mode or Incremental Mode. Full Mode refers to data loaded for the first time or data that is truncated and then loaded. Incremental Mode refers to new or changed data being added to the existing data.</p> <p>The DAC maintains a last refresh timestamp whenever a table is changed during the ETL process. (You can view this timestamp under Design &gt; Tables &gt; Database Connections or Setup &gt; Database connections &gt; Tables.) If a table has a timestamp, the command appearing in this column is executed. If a table does not have a timestamp, the command for a full load is executed. If the execution type is Informatica, the workflow name is used as the command.</p>
Command for Full Load	If a table has no last refresh timestamp, this command is executed.
Folder Name	For execution type of Informatica only. The folder in which the workflow resides.
Source DB	Source database that is queried.
Target DB	Target database that is populated.
Task Phase	Task phase of the ETL process. The DAC server uses the task phase to prioritize tasks and to generate a summary of the time taken for each of the phases.

**Table 27. Task Definitions List Columns**

Column	Description
Execution Type	<p>Tasks are executed based on their execution type. The following types are supported:</p> <ul style="list-style-type: none"> <li>■ <b>Informatica.</b> Task is invoked on an Informatica server using pmcmd.</li> <li>■ <b>External Program.</b> Task is an operable program on the operating system where the DAC server is running. This program can be a batch file, shell script, or any other program that can be run like a bulk loader.</li> <li>■ <b>SQL File.</b> Task is a SQL script in .xml or .sql format. For more information, see <a href="#">“Using SQL Files as an Execution Type in the DAC” on page 148.</a></li> <li>■ <b>Stored Procedures.</b> Task is a stored procedure that has been defined on the databases.</li> </ul> <p>In addition, there are several internal execution types that you will not be able to select when creating new tasks. These tasks are categorized as either internal change capture tasks or internal data warehouse tasks; all of these tasks are color-coded in pink in the Task Definitions tab.</p> <ul style="list-style-type: none"> <li>■ <b>IMG_BUILD</b> - internal change capture. If you are using multiple Siebel transactional sources, you cannot change the behavior of the change capture process. This task requires change capture tables to be created on the other sources also. When adding additional Siebel sources, go to Setup &gt; Database Connections and select the appropriate Siebel source, right-click and select Change Capture Tasks. This action generates change capture tasks. Use this same action to disable or delete change capture tasks.</li> <li>■ <b>IMG_SYNC</b> - internal change capture. If you are using multiple Siebel transactional sources, you can create this task for the additional tasks for doing similar change capture sync processes. You cannot change the behavior of the change capture sync process. This task requires change capture tables to be created on the other sources also. This task should be used with discretion for Siebel sources only.</li> </ul>

**Table 27. Task Definitions List Columns**

Column	Description
Execution Type	<ul style="list-style-type: none"> <li>■ QUERY_INDEX - internal data warehouse. This task allows you to alter when the Query indices are created. The DAC server drops all indices before loading when the CreateQueyIndexesAtTheEnd setting is set to True. When this setting is set to False, all the indices, regardless of the index type, get created as part of the task that does the loading.</li> <li>■ UPDATE_ETL_PARAM - internal data warehouse. This task is used for only to update W_PARAM_G from the DAC server. This task populates the preferences to the W_PARAM_G table in the Data Warehouse by querying values defined in the DAC repository. Because only one Data Warehouse per DAC repository is supported, this execution type should not be chosen for any task.</li> </ul>
Build Image	Applicable for Siebel transactional sources only. When this check box is selected, the change capture for the primary/auxiliary source tables executes.
Truncate Always	When this check box is selected, the target tables are truncated regardless of whether a full or incremental load is occurring. Any indices registered for this table are dropped before the command is executed and are recreated after the command completes successfully. When indices are dropped and created, the table is analyzed so that the index statistics are up-to-date.
Truncate for Full Load	When this check box is selected, the target tables are truncated only when a full load is occurring. Any indices registered for this table are dropped before the command is executed and are recreated after the command completes successfully. When indices are dropped and created, the table is analyzed so that the index statistics are up-to-date. When the Truncate Always option is selected, this option has no meaning.



**Table 27. Task Definitions List Columns**

<b>Column</b>	<b>Description</b>
Optional	<p>When one or more optional prerequisite tasks (Depends On tasks) is run, the task is also included if this check box is selected. Otherwise, the task is ignored.</p> <p>For example, when table W_Person_F is loaded, W_Person_D is also loaded. However, the task UpdateRecencyCat in Person Dimension that writes to W_Person_D is optional. Only when the tasks Load into OrderDimension or Load into OrderItemFact (optional dependencies) are run does UpdateRecencyCat in Person Dimension need to be executed. So, this optional task will not be executed unless one or more of the tasks that it is depending on optionally is chosen for execution.</p>
Continue on Error	<p>When this check box is selected, if the command fails, the dependent tasks are not stopped. However, if any auto-generated tasks fail, the dependent tasks are stopped.</p>
Inactive	<p>When this check box is selected, the record is inactive. Inactive tasks do not participate in the ETL process.</p>

[Table 28](#) provides a description of the tabs in the bottom pane.

**Table 28. Task Definitions Subtabs**

<b>Tab</b>	<b>Description</b>
Edit	Allows you to edit the table selected in the top pane.
Description	Displays a description of the task selected in the top pane.

**Table 28. Task Definitions Subtabs**

Tab	Description
Source Tables	<p>Displays the list of tables used for getting data for the task selected in the top pane.</p> <p>The list contains the following column headings:</p> <ul style="list-style-type: none"> <li>■ <b>Name.</b> Name of the table.</li> <li>■ <b>Primary.</b> Indicates whether the table is a primary source of data.</li> <li>■ <b>Auxiliary.</b> Indicates whether the table is a secondary source of data.</li> </ul> <p><b>Note:</b> If a table is marked as Primary or Auxiliary and the Build Image property of the task is selected, the change capture process is invoked. There are special tasks that force the base table data to be extracted when data in auxiliary tables change.</p> <p>A table can be neither Primary nor Auxiliary but still be used for getting some attributes to populate a dimension or fact table. The changes in these kinds of source tables are not reflected in the dimension or fact table once the data is populated.</p>
Target Tables	<p>Displays the list of tables where data is loaded for the task selected in the top pane.</p> <p>The list contains the following column headings:</p> <ul style="list-style-type: none"> <li>■ <b>Name.</b> Name of the table.</li> <li>■ <b>Primary.</b> Indicates whether the table is a primary target table. For documentation purposes only.</li> </ul>

**Table 28. Task Definitions Subtabs**

Tab	Description
Depends On Tasks	<p>Displays a list of tasks that are prerequisite for the execution of the task selected in the top pane. If any of the prerequisite tasks fail, the task selected in the top pane will be marked as Stopped.</p> <p>The list contains the following column headings:</p> <ul style="list-style-type: none"> <li>■ <b>Name.</b> Name of the prerequisite task.</li> <li>■ <b>Task Type.</b> Type of task (read only).</li> <li>■ <b>Optional Flag.</b> When this flag is selected, the prerequisite task optionally depends on the task selected in the top pane. For example, task 1 loads table A, and task 2 loads table B and optionally depends on task 1. If an execution plan requires both tables A and B to be loaded, task 2 will wait for task 1 to successfully complete. If an execution plan requires only table B to be loaded, task 2 will be executed immediately, and task 1 will not be executed.</li> </ul>
Dependent Tasks	<p>Displays a list of tasks that are dependent on the task selected in the top pane. These tasks wait for the successful completion of the current task before they are executed.</p> <p>The list contains the following column headings:</p> <ul style="list-style-type: none"> <li>■ <b>Name.</b> Name of the dependent task.</li> <li>■ <b>Task Type.</b> Type of task (read only).</li> <li>■ <b>Optional Flag.</b> See the description for Depends On Tasks.</li> </ul>

### Actions Available

The following actions are available in the top pane toolbar and in the right-click menu when the Task Definitions tab is active.

#### Copy Record

This command creates a copy of the selected task and also copies the references to the tasks it depends on and source and target tables.

## Using SQL Files as an Execution Type in the DAC

There are two types of custom SQL files that can be executed through the DAC: XML formatted .xml files, and plain text .sql files. For examples of XML and SQL files, go to the DAC\CustomSQL folder.

### XML Formatted Files

An XML file consists of a set of SQL statements for which the name, type, and Continue on Fail option are defined using XML attributes. The set of SQL statements is in a CDATA section which allows for special characters (like <, >, \) to be used without breaking the XML structure. A CDATA section looks like the following:

```
<![CDATA[this is a place for a SQL statement]]>
```

An example of an XML file follows:

```
<?xml version="1.0"?>
<CustomSQLs>
<sql name="Create Temp" type="SQL" continueOnFail="true">
<![CDATA[CREATE TABLE w_etl_temp (row_wid varchar(50))]]>
</sql>
<!-- This is how a comment section looks in XML -->
<!-- It will be ignored-->
<sql name="Update Temp" type="SQL">
<![CDATA[UPDATE w_etl_temp SET row_wid = 'qwe' WHERE row_wid LIKE
'5*']]>
</sql>
<sql name="Drop Temp" type="SQL">
<![CDATA[DROP TABLE w_etl_temp]]>
</sql>
</CustomSQLs>
```

This example consists of three SQL statements: Create Temp, Update Temp, and Drop Temp. These names will be used in DAC run task detail description to give details of the errors encountered during execution (if any). In this particular example all three statements are of type SQL. Regular update SQL statements and the Stored Procedure type can also be used.

If the Continue on Fail option is set to True for a given statement, the execution will continue even if the task fails. If the Continue on Fail option is set to False for a given statement, or is omitted, the subsequent statements will not be executed and the Task Run status will be Failed.

### Plain Text SQL Files

Plain text SQL files consist of a set of SQL statements (no stored procedure calls). The SQL statements are separated by a semicolon (;), and comment tags are allowed (//, /\* comment \*/, --). If any of the SQL statements fail, the Task Run status will be Failed.

An example of a plain text SQL file follows:

```
CREATE TABLE w_etl_temp (name varchar(50))
;

UPDATE w_etl_temp

SET name = 'that's right' //this line demonstrates the use of ' in
a text area

WHERE name LIKE 'gone fishing%';

/*

*some

*query

*statement

*/

SELECT * FROM w_etl_temp
```

```
;  
  
DROP TABLE w_etl_temp  
  
;  
  
/*end of file*/
```

## About the DAC Setup View

The Setup View provides access to functionality related to setting up DAC system properties, Informatica servers, database connections, and email notification.

This section includes the following topics:

- [“About the DAC System Properties Tab” on page 150](#)
- [“About the DAC Informatica Servers Tab” on page 150](#)
- [“About the DAC Database Connections Tab” on page 151](#)
- [“About the DAC Email Recipients Tab” on page 151](#)

## About the DAC System Properties Tab

The DAC System Properties tab allows you to configure various properties that determine the behavior of the DAC server. The top pane displays a list of the properties and their current values. For more information about the DAC system properties, see [“Setting Up DAC System Properties and Resources” on page 75](#).

## About the DAC Informatica Servers Tab

The Informatica Servers Tab allows you to register one or more Informatica servers and specify how many workflows can be executed in parallel on each server. The DAC server automatically load balances across the servers. For more information about Informatica server properties, see [“Registering Informatica Servers” on page 79](#).

## **Actions Available**

The following actions are available in the top pane toolbar and in the right-click menu when the Informatica Servers tab is active.

### **Test Connection**

This command allows you to test the connection to the database. You should use this command on the machine where the DAC server is running.

### **Copy Record**

This command creates a copy of the selected record.

## **About the DAC Database Connections Tab**

This tab provides access to the database connection properties. The top pane Indices tab displays a list of the properties and their current values. You can edit the property values in the top pane by double-clicking in a cell. For more information about database connection properties, see [“Creating Database Connections” on page 80](#).

## **Actions Available**

The following actions are available in the top pane toolbar and in the right-click menu when the Database Connections tab is active.

### **Test Connection**

This command allows you to test the connection to the database. You should use this command on the machine where the DAC server is running.

### **Change Capture Tasks**

This command allows you to create, enable, or disable internal tasks for change capture. Such tasks are needed only for connections of the Source type. Avoid creating change capture tasks for non-Siebel sources.

## **About the DAC Email Recipients Tab**

This tab allows you to set up a list of email addresses that will be notified about the status of the ETL process. The top pane displays a list of the properties and their current values. You can edit the property values in the top pane by double-clicking in a cell. For more information about the Email Recipients properties, see [“Configuring Email Recipients” on page 82](#).

### Actions Available

The following actions are available in the top pane toolbar and in the right-click menu when the Email Recipients tab is active.

#### Copy Record

This command creates a copy of the selected record.

## About the DAC Execute View

The Execute View provides access to functionality that allows you to run, schedule, and monitor execution plans.

This section includes the following topics:

- [“About the DAC Execution Plans \(Read Only\) Tab” on page 152](#)
- [“About the DAC Current Run Tab” on page 152](#)
- [“About the DAC Run History Tab” on page 156](#)
- [“About the DAC Scheduler Tab” on page 156](#)
- [“About the DAC Validation Tab” on page 158](#)

### About the DAC Execution Plans (Read Only) Tab

The Execution Plans tab in the Execute view displays, in read-only mode, the available execution plans. This information is the same as that displayed in the Execution Plans tab in the Design view. For more information, see [“About the DAC Execution Plans Tab” on page 130](#).

### About the DAC Current Run Tab

This tab displays a list of queued, running, and failed current ETL processes in the top pane. This list includes comprehensive information about each process.



Table 29 provides a description of the information provided in the top pane.

**NOTE:** Once an ETL process completes, it is accessible from the Run History tab.

**Table 29. Current Run Tab Columns**

<b>Column</b>	<b>Description</b>
Name	Name of the ETL process
Execution Plan Name	The execution plan whose run-time instance is this record. This field is read only.
Process ID	ID for the process. This value is an integer that is incremented by 1 for every run. This value is stored as ETL_PROC_WID in all the Data Warehouse tables. This field is read only.
Run Status	Status of the run. The possible values are the following: <ul style="list-style-type: none"> <li>■ Queued. A request has been submitted. Displayed in yellow in the Current Run list.</li> <li>■ Running. The server has picked up the run and starts processing it. Displayed in blue.</li> <li>■ Failed. One or more tasks has failed. Displayed in red.</li> <li>■ Completed. All tasks have completed without errors. Displayed in green.</li> </ul>
Total Number of Tasks	Total number of tasks for this run. This field is read only.
Number of Failed Tasks	Sum total of tasks that have failed and that have stopped. This field is read only.
Number of Successful Tasks	Number of tasks whose status is completed. This field is read only.
Number of Tasks Still in Queue	Number of tasks whose prerequisite tasks have not completed, and the number of tasks whose prerequisite tasks are completed and are waiting for resources. This field is read only.
Start Timestamp	Start time of the ETL. Reflects the start time of every ETL attempt. For example, if the ETL fails and is run again, it gets a new start timestamp. The history of attempted runs is maintained in the audit trail for the run. This field is read only.

**Table 29. Current Run Tab Columns**

<b>Column</b>	<b>Description</b>
End Timestamp	End time of the ETL. Reflects the end time of every ETL attempt. For example, if the ETL fails and is run again, it gets a new end timestamp when the ETL is complete. The history of attempted runs is maintained in the audit trail for the run. This field is read only.
Duration	A calculated field that shows the difference between start and end time stamps.
Status Description	Displays messages generated during run time. You can add notes to this field for Completed runs.

[Table 30](#) provides a description of the tabs in the bottom pane.

**Table 30. Current Run Subtabs**

<b>Tab</b>	<b>Description</b>
Edit	Displays the status of the ETL selected in the top pane.
Description	Provides a description of the ETL process, including status and environment information. This description is updated at the beginning of the ETL process and when the process is complete.

**Table 30. Current Run Subtabs**

<b>Tab</b>	<b>Description</b>
Tasks	<p>Displays run-time instances of the tasks. As the execution proceeds, the tasks are executed based on the dependency rules and some prioritization.</p> <p>As tasks complete, the tasks that depend on the completed tasks are notified and once their dependencies are completed, they become eligible to run. If a task fails, the administrator can address the failure and then requeue the task or mark it as completed. The DAC server polls for any changes in the failed task's detail status. If a failed task detail is queued, the task itself gets back into the ready to run queue and all its dependent tasks get into the queued status.</p> <p>The rules of prioritization are as follows:</p> <ul style="list-style-type: none"> <li>■ Tasks with no dependencies are executed first.</li> <li>■ If a task has failed and has been requeued, it gets the maximum priority.</li> <li>■ Tasks with greater phase priorities are executed next. When several tasks of the same phase are eligible to run, the tasks with greater task priorities are executed next.</li> </ul>
Summary (Read Only)	Provides a summary (based on dynamic SQL) of the ETL selected in the top pane. A start and end time for each phase is included.
Task Details	Provides a execution details about each task associated with the ETL selected in the top pane.
Audit Trail (Read Only)	Provides the history of the ETL selected in the top pane.

### **Actions Available**

The following actions are available in the top pane toolbar and in the right-click menu when the Current Run tab is active.

#### **Start**

This command restarts the selected ETL, after the ETL has failed, stopped, or been aborted.

#### **Abort**

This command causes an ETL in progress to abort. All currently running tasks will be aborted. The status of queued tasks and the ETL itself will change to Stopped.

### **Auto Refresh**

This command allows you to turn on and off the automatic screen refresh functionality and set the refresh interval.

### **Get Log File**

This command fetches the log file for this run from the DAC server and saves it in the ServerLog folder.

### **Mark as Completed**

This command changes the status of a stopped or failed ETL to Completed. In the audit trail for this ETL, the status is Marked as Completed. Use this command with caution; it can cause the data warehouse to be inconsistent.

### **Analyze Run**

This command saves a description of the run as an HTML file in the Log/Statistics folder.

## **About the DAC Run History Tab**

The Run History tab displays information about completed ETL processes. The information displayed in the top and bottom panes is the same as in the Current Run tab. For a description of the information displayed in the Run History tab and the actions available, see [“About the DAC Current Run Tab” on page 152](#).

## **About the DAC Scheduler Tab**

The Scheduler tab allows you to schedule ETL processes to be run either once at a later time or periodically. When you schedule an ETL or make changes to a schedule, the DAC server picks up the information from the DAC client. The DAC server polls the DAC repository for changes periodically at a frequency set in the DAC system properties. For information about DAC system properties, see [“Setting Up DAC System Properties and Resources” on page 75](#).

The top pane of the Scheduler tab lists ETL processes that have been scheduled. The bottom pane allows you to schedule an ETL process.

Table 31 provides a description of the information displayed in the top pane list.

**Table 31. DAC Scheduler Tab List Columns**

Column	Description
Name	User defined name of the schedule.
Execution Plan	Execution plan scheduled.
Recurrence	Indicates how often the schedule will be run.
Inactive	Indicates whether the schedule is active or inactive.

### Actions Available

The following actions are available in the top pane toolbar and in the right-click menu when the Scheduler tab is active.

#### Copy Record

This command creates a copy of the selected record.

### Scheduling an ETL

Use this procedure to schedule an ETL in the DAC.

- 1 In the DAC, navigate to the Scheduler tab.  
The current list of schedules appears in the top pane.
- 2 Click New in the top pane toolbar.  
The Edit tab in the bottom pane becomes active.
- 3 Enter a name for the schedule.
- 4 Select an execution plan.
- 5 If you want the schedule to run once, select the Run Only Once check box, and then select a start and end date and time.
- 6 To create a periodic schedule, select a recurrence pattern, and enter the appropriate date and time parameters.
- 7 Click Save.

### About the DAC Validation Tab

The Validation tab allows you to view the results of a validation. The following kinds of validation are available:

- **Resource Validation:** Checks to see if the database connections are validated. This will be checked as a first step of the ETL process. If the database connections are not valid, the ETL will abort. This validation is also available on the DAC client through Setup > Database Connections > Test Connection functionality.
- **Table Validation:** Checks to see if the tables do exist in the database. If the tables do not exist, the ETL will fail. This is not checked during the ETL run. The check also includes verifying the existence of image tables where applicable.
- **Index Validation** checks the following:
  - a** If all the columns in the index definition exist on the table in the database.
  - b** If the index does exist in the database, the validation compares the index definition and points out the differences.
  - c** If there are multiple indices with the same definitions (columns in the same order) for a table.
  - d** If there are indices with the number of columns greater than 255.
  - e** If '# Include columns' is greater than the number of columns provided in the index column list.
  - f** Checks to see if the index column has no columns.
  - g** If the number of columns is more than 16. Some databases allow for more than 16 columns. However, the user should inspect the warning and verify its validity.
  - h** If an index on a table was created on the database but not defined in the repository. This may cause the ETL process to fail, because all the indices need to be dropped prior to the load, otherwise the bulk loaders may fail.

- Informatica Workflow Validation: Verify for the task definitions of type Informatica to match the 'Command for Full Load' and 'Command for Incremental Load' do exist in the Informatica repository. For this validation to work, configure the system preferences 'Main Informatica Repository' to point to the database where the Informatica repository resides with the logical name given to the database under Setup- > Database Connections.
- Check for circular dependencies: While defining task dependencies, care should be taken to not to define circular dependencies. For example task A depends on task B, and task B depends on task C, and task C depends on task A. Such occurrences will be identified. If these errors do exist, the ETL will fail.

## **DAC Functional Reference**

---

*About the DAC Execute View*



This chapter describes concepts and techniques for customizing the Siebel Data Warehouse. It also provides an overview of how incremental Siebel application operational data is captured and how it can be used for adding columns and tables to the Siebel Data Warehouse. It also describes customization techniques using extension tables and Informatica mapping templates.

This information applies only to Siebel Analytics applications versions 7.7.

---

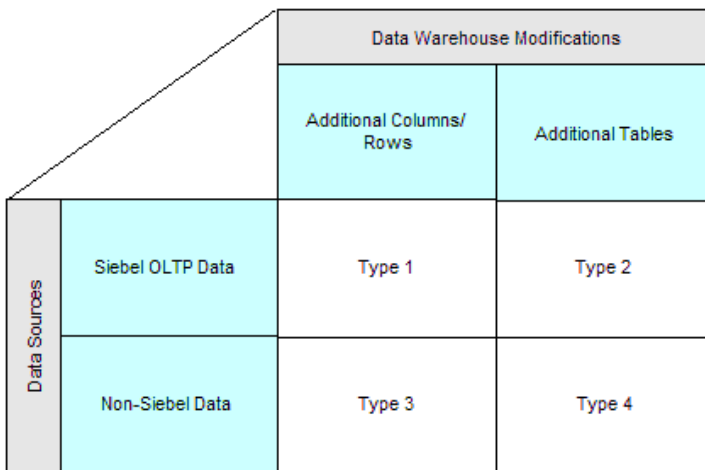
**NOTE:** Customization should be done only by those experienced in Informatica and Siebel Analytics applications.

---

## Siebel Data Warehouse Customization Scenarios

In customizing the Siebel Data Warehouse, various scenarios are available to you based on whether the data source is the Siebel OLTP database or data from a non-Siebel source. The matrix in [Figure 10](#) shows categories of customization scenarios. The kinds of customizations available for each category are described in [Table 32](#). For instructions on how to perform the customizations, see “[Customization Steps for Siebel Data](#)” on page 174 and “[Siebel Data Warehouse Customization Steps for Non-Siebel Data](#)” on page 180.”

For detailed information about ETL processes and tables, see *Siebel Data Warehouse Data Model Reference*.



**Figure 10. Data Warehouse Customization Scenarios**

**Table 32. Types of Data Warehouse Customization Scenarios**

Type	Modification
Type 1	Add a column to the data warehouse from a primary source table in the Siebel OLTP database
	Using Siebel OLTP data, add a column to the data warehouse from a related source table in the Siebel OLTP database
	Using Siebel OLTP data, modify an existing column in the data warehouse

**Table 32. Types of Data Warehouse Customization Scenarios**

Type	Modification
Type 2	Using Siebel OLTP data, add a new dimension table and use it with an existing fact table
	Using Siebel OLTP data, add a new fact table
	Using Siebel OLTP data, add a new dimension table for a new fact table
Type 3	Using non-Siebel data, add new data as a whole row into an existing dimension table
	Using non-Siebel data, add a column for an existing dimension table
	Using non-Siebel data, add a column for an existing fact table
Type 4	Using non-Siebel data, add a new dimension table
	Using non-Siebel data, add a new dimension table and use it with an existing fact table
	Using non-Siebel data, add a new fact table

## Siebel Data Warehouse ETL Process Concepts

This section includes a detailed description of the ETL process. The following topics are covered:

- [“About Siebel Data Warehouse Extension Tables” on page 164](#)
- [“About the Siebel Data Warehouse DATASRC\\_NUM\\_ID Column” on page 166](#)
- [“Working with the Custom Folder of the Siebel Data Warehouse” on page 166](#)
- [“Creating Custom Informatica Workflows” on page 167](#)
- [“Important Notes About Siebel Data Warehouse Customizations” on page 169](#)

### About Siebel Data Warehouse Extension Tables

The extension tables are provided with the Siebel Data Warehouse exclusively to help customize the data warehouse tables.

---

**CAUTION:** Do not customize existing mappings or the database schema. If you need to customize the standard mappings, create a new mapping that loads the extension table instead.

---

For every extension table, there is an extension staging table that is populated by the custom SDE process. The custom SIL process sources data from this staging table and populates the final extension table. For information about how to use extension tables, see [“Customization Steps for Siebel Data” on page 174](#) and [“Siebel Data Warehouse Customization Steps for Non-Siebel Data” on page 180](#).

The extension tables have a 1:1 relationship with their corresponding dimension and fact tables.

The data warehouse extension tables are different from the OLTP database extension tables in the following ways:

- Data warehouse extension tables are not maintained by Siebel Tools.
- There is no relationship between data warehouse extension columns and the corresponding OLTP extension columns. For example, attr01 in S\_ORG\_EXT\_X is not the same as attr01 in W\_ORG\_DX.
- Mappings have to be implemented explicitly to populate the extension columns.

#### Using the Extension Tables

The extension tables have a 1:1 mapping between the dimension and fact records to the extension records. To insert a record into a dimension or fact table, choose the appropriate column and write a new mapping that uses the ETL server's incremental logic to load the staging table. When populating data from the staging table to the final dimension or fact tables using the INTEGRATION\_ID and the DATASRC\_NUM\_ID, find the primary key of the dimension table and use that as the primary key of the extension table.

## Naming New Extension Tables

When you create a new extension table, use the same naming convention as is used in the Siebel Data Warehouse by adding the prefix WC to the table name, for example, WC\_TABLENAME\_DS. This will simply future upgrades to the Siebel Data Warehouse.

---

**NOTE:** If a new table is created, you need to register the table and its indices in the DAC repository. Also, if you modify indices on existing tables, you need to register the indices.

---

## Creating New Dimension or Fact Tables

You can create new tables using the same logic as is used for extension tables.

---

**CAUTION:** The data in any customized table will face upgrade issues.

---

If you are creating a new dimension or fact table, use the required system columns that are part of each of the data warehouse tables to maintain consistency and the ability to reference existing table structures. When you create a new table, you need to register the tables and its indices in the DAC repository.

### Required Columns

For main and extension staging tables the following columns are required:

- **INTEGRATION\_ID.** Stores the primary key or the unique identifier of a record as in the source table.
- **DATASRC\_NUM\_ID.** Stores the Data source from which the data is extracted. For example, Siebel OLTP is 1.

For data from any external data source assign a number greater than 1.

For dimension and fact tables, the required columns are the INTEGRATION\_ID and DATASRC\_NUM\_ID columns as well as the following:

- **ROW\_WID.** A sequence number generated during the ETL process, which is used as a unique identifier for the data warehouse.

- **ETL\_PROC\_WID.** Stores the ID of the ETL process information. The details of the ETL process are stored in the S\_ETL\_RUN table on the OLTP side and W\_ETL\_RUN\_S table on the OLAP side. This is also the Process ID on Current Run/Run History screen in the DAC.

The required columns for dimension and fact extension tables are ROW\_WID and ETL\_PROC\_WID.

### About the Siebel Data Warehouse DATASRC\_NUM\_ID Column

All the tables in the Siebel Data Warehouse schema have DATASRC\_NUM\_ID as part of their unique user key. This means that the rows can be loaded in the same data warehouse tables from different sources if this column is given a different value for each data source.

---

**NOTE:** The DATASRC\_NUM\_ID value of 1 is reserved for the main Siebel data source and cannot be used for other sources.

---

### Working with the Custom Folder of the Siebel Data Warehouse

The Siebel Data Warehouse provides a folder within the repository specifically designed for customized changes to an implementation. This folder is named CUSTOM\_DW\_REP. Using the Informatica Designer, make modifications only in the CUSTOM\_DW\_REP folder.

Do not change objects in the SIEBEL\_DW\_REP folder unless it is explicitly mentioned. This folder may be overwritten in future upgrades. All changes should be placed in the CUSTOM\_DW\_REP folder within the Informatica Designer.

The easiest way to modify an object is to copy an existing object from SIEBEL\_DW\_REP into CUSTOM\_DW\_REP and to reuse existing business components, source and target definitions, transformations, mapplets, and mappings before you make any changes.

---

**NOTE:** If source tables are extended (after copying into the CUSTOM\_DW\_REP folder), then the source tables require manual editing in the Informatica Designer. Do not import the tables from the database, because it changes the source table definitions for the entire Siebel Data Warehouse.

---

When importing new tables from the database into the CUSTOM\_DW\_REP folder, use the Siebel Data Warehouse and Siebel Transaction Database ODBC database connections to connect to the source and target databases.

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

## Creating Custom Informatica Workflows

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

- Create the workflow such that each loads only one table. This helps to integrate the workflow with the DAC.
- The workflow name should match a session name that is used inside the workflow. This helps the DAC to collect certain statistics.
- Fail parent if this task fails should be checked for all the sessions within a workflow.
- Fail parent if this task does not run should be checked for all the sessions within a workflow.
- The Stop on Errors parameter should be set to 1 for all sessions with a workflow. This parameter is located in the Error Handling area of the Config Object tab in Informatica Designer.

- Set the appropriate source and target connection values in Informatica Designer.
- If the workflow is going to be used for full load command, you can choose to load in bulk mode (applicable only to Oracle and DB2-UDB databases). If the workflow is going to be used for the full load command in the DAC, then in the Properties tab of Informatica Designer, set the Target Load type to Bulk. This requires that the target table does not have any indices during the load. The DAC will drop the indices automatically; no action is required on your part.
- For all entities, such as dimension and fact tables, create two workflows, one to be used for a full load and the other to be used for an incremental load. This provides an opportunity to tune each of these load scenarios.
- If the workflow is designed for loading a dimension in a full mode, make sure the workflow also includes a session for the unspecified row creation.
- You can decide to truncate the target table in the workflow, but it is necessary that while defining a task in the DAC, you choose appropriate truncate option. This helps the DAC to decide whether to drop and create indices on the table.
- If some sessions have to be run in serial order and if all the sessions need to be rerun upon failure of the workflow, then you should consider designing a single workflow that has sessions run in serial. If upon failure, not all of the sessions need to be run again, then consider designing separate workflows and define dependencies in the DAC.
- Use views created by the DAC as the main source table in all extract processes.
- If a workflow uses a list of values, make it depend on the Fix List Of Values and Copy to OLTP task definition in the DAC.
- If a workflow uses exchange rates, make it depend on the Extract for ExchRate task definition in the DAC.
- If a workflow uses cost lists, make it depend on the Load Cost Lists task definition in the DAC.
- You can import the workflows into the DAC repository's temporary area while defining tasks.

### **Customization Categories**

There are three categories of customizations:



- **Customizing preconfigured workflows.** Copy the mapping from the Siebel\_DW\_Rep folder into the Custom folder. Modify it as necessary. Create new workflows in the Custom folder. In the DAC, search for the task that executes the workflow and change the folder to the Custom folder. Update the command for full and incremental loads appropriately if the workflow names have been changed.
- **Creating new workflows to populate additional custom entities.** The custom workflows can be plugged into an ETL process by registering them in the DAC. A new task needs to be registered in the DAC, with the appropriate properties set. Also, you must define source and target tables, task definitions, and dependencies. In the Depends On Tasks tab, select the list of tasks that are prerequisite for the current task to execute. In the Dependent Tasks tab, define the tasks that need to wait for the successful completion of this task. Set the Optional flag to True.
- **Pruning preconfigured ETL processes.** Pruning is the process in which you modify a preconfigured execution plan to meet your needs. You can prune an execution plan by using the DAC to inactivate certain components of the execution plan, such as subject areas, table groups, tables, tasks, and dependencies. Inactivating the component means the component does not participate in the ETL process when the execution plan is run. For more information about pruning, see [“Handling ETL Failures with the DAC” on page 125](#).

## **Important Notes About Siebel Data Warehouse Customizations**

All custom work, unless stated otherwise, must be done in the Custom folder so that the custom work can be preserved during an Informatica repository upgrade. Doing work on the standard folder should be avoided whenever possible. Whenever custom work is done in the standard folder, a careful note of the custom work should be recorded so that the customizations can be reapplied whenever the Informatica repository is upgraded. An upgrade of the Informatica repository overrides any changes to the standard folder.

### **Additional Customization Instructions**

- **Table definitions in Informatica.** Make sure that the SQL style is set to Oracle while importing the table definitions from external data sources. Even if the actual data source is of another database type, such as DB2 or MSSQL, it does not affect the logic of how data gets loaded.

- **Update strategy.** For loading new fact and dimension tables or loading data into the existing tables, design a custom process on the source side to detect the new and modified records. The SDE process should be designed to load only the changed data (new and modified). If the data is loaded without the incremental process, the data that was previously loaded will be updated again, which is a costly process. For example, the logic in the OTB SIL mappings looks up the destination tables based on the INTEGRATION\_ID and DATASRC\_NUM\_ID and returns the ROW\_WID if the combination exists, in which case it updates the record. If the lookup returns null, it inserts the record instead.
- **ETL process.** When using multiple sources for the data warehouse, you can decide to load from all of them at the same time or at different time frequencies using different execution plans.
- **Truncating target tables.** Truncating should be done through the DAC. A single task has place holders for a full load, and one for an incremental load.
  - For the SDE workflows, the commands for full load and incremental load are the same. They should have Truncate Always flag checked. For these kinds of tasks, the command for full load and incremental load are based on the same mapping.
  - For SIL workflows, the command can be different for full and incremental loads. They should have the Truncate For Full Load option checked. When a table gets truncated, the indexes are automatically dropped and created after the data is loaded. And so, the workflow for full load command can have the Bulk Load option turned on, and, therefore, you can have an optimized version of the mapping that does not need to look up for finding if a record needs to be inserted or updated.
  - Auxiliary tasks need to be run only during incremental mode. So, for these tasks, the full load command is empty. No truncate options should be set.

	<b>SDE Sessions</b>	<b>SIL Sessions</b>	<b>Aux Sessions</b>
<b>Truncate Option</b>	Truncate Always	Truncate for Full Load	None

- **ETL\_PROC\_WID.** Use the same ETL\_PROC\_WID in W\_PARAM\_G table in custom mappings. ETL\_PROC\_WID is a reference key to Run History in the DAC. To use the same ETL\_PROC\_WID, copy the reusable lookup (called LKP\_ETL\_PROC\_WID) defined in the Siebel\_DW\_Rep folder. The input to the lookup is a constant (hard coded to 1).
- **DATASRC\_NUM\_ID.** Even though preconfigured mappings look up W\_PARAM\_G for getting this value, for customizations that move data from other sources, you should hard code the value with a number other than 1, which is reserved for the main Siebel transaction database in custom mappings.
- **Creating indexes and naming conventions.** Staging tables typically do not require any indexes. Use care to determine if indexes are required on staging tables. Create indexes on all the columns that the ETL will use for dimensions and facts (for example, ROW\_WIDs of Dimensions and Facts, INTEGRATION\_ID and DATASRC\_NUM\_ID, and flags). Carefully consider which columns or combination of columns filter conditions should exist, and define indexes to improve query performance. Inspect the OTB objects for guidance. Name all the newly created tables as WC\_. This helps visually isolate the new tables from the OTB tables. Keep good documentation of the customizations done; this helps when upgrading your data warehouse. Once the indexes are decided upon, they should be registered in the DAC, either manually or by right-clicking on the certain table and invoking the Import Indices command.
- **Currency.** For currency-related data, populate the table with the base currency and exchange date field (in order to convert the data appropriately). The data for currency conversion should be maintained in the main Siebel data source. (The currency data is maintained by converting all the currency information into a single Base Currency code specified in the DAC's ETL preferences.)
- **Day Dimension.** For the data that is related to W\_DAY\_D, use the reusable transformation EXP\_DAY\_DIMENSION\_FK\_RESOLUTION. Note that this transformation will take a date as input and return the foreign key to the Day dimension as output in the appropriate format (YYYYMMDD). Copy the reusable transformation and use it.

- **List Of Values.** This applies in particular to Type 1 and 2. The preconfigured columns that depend on a list of values have a language-dependent column and language-independent column. Use the mapplet MPLT\_LOV\_TRANSLATION to populate the language-dependent and independent columns in the dimension tables. For fact tables, use MPLT\_LOV\_D\_ROW\_WID to create a new foreign key to the LOV dimension. One could also deal with translations directly in SQL overrides to improve performance.

## Standard Templates for Siebel Data Warehouse Customizations

This section describes the standard templates.

### CustomTemplateSDE\_PersonDimension

Using the SDE mapping, the ATTRIB\_02 column of the extension stage table (W\_PERSON\_DSX) is populated with the value from S\_CONTACT.RACE source column. This is a straightforward mapping. The different transformations are the following:

- **SQ\_S\_CONTACT.** The SQL in the SQL override section contains the link to the view created by the DAC so that only the incremental data is captured.
- **EXPTRANS.** The lookup (LKP\_DATASOURCE\_NUM\_ID) obtains the DATASOURCE\_NUM\_ID from the W\_PARAM\_G table. The W\_PARAM\_G is populated in the beginning of the ETL by the DAC. Note that all the system columns are populated in addition to the attrib column. The ATTRIB\_01 column is populated with the string Contact, because the combination of INTEGRATION\_ID, DATASOURCE\_NUM\_ID, and CONTACT\_TYPE is used to look up the ROW\_WID value. In most of the dimensions, only the INTEGRATION\_ID and DATASOURCE\_NUM\_ID values are used for this purpose.

### CustomTemplateSIL\_WaveDimension

Using the SIL mapping, the ATTRIB\_02 column of the final extension table is populated with the person dimension (W\_WAVE\_DX). The different transformations are the following:

- **SQL\_W\_WAVE\_DSX.** There is no SQL in the override. It is a straight query from W\_PERSON\_DSX table that has been populated with incremental rows.
- **LKP\_W\_WAVE\_D.** Looks up the W\_WAVE\_D table to get the ROW\_WID for the row.
- **LKP\_W\_WAVE\_DX.** Looks up the target table to find out whether to insert or update the row. This information is used by the update strategy.
- **EXPTRANS.** Used to get the Insert/Update flag and ETL\_PROC\_WID by using the lookup LKP\_ETL\_PROC\_WID from the W\_PARAM\_G table. This table is populated in the beginning of the Refresh Siebel DW session.
- **FILTRANS.** Passing only those rows that have entry in the parent dimension table.
- **UPDTRANS.** Insert or updates a row based on the flag.

## Fact Templates for Siebel Data Warehouse Customizations

This template shows how to attach an existing dimension (account dimension) to a fact (asset fact).

### CustomTemplateSDE\_AssetFact

Using this SDE mapping, the staging extension table for the Asset fact is populated with incremental data. The different transformations are the following:

- **SQL\_S\_ASSET.** Selects the OWNER\_ACCNT\_ID that will ultimately be used to join with the account dimension in the fact. Note the extra *where* clause joining to the incremental image table.
- **EXPTRANS.** Gets DATASOURCE\_NUM\_ID using lookup LKP\_DATA\_SOURCE\_NUM\_ID from table W\_PARAM\_G. This table is populated in the beginning of the session by the DAC server.

Finally, the FK\_01\_ID field of the extension stage table W\_ASSET\_FSX is populated. All the system columns are also populated.

#### CustomTemplateSIL\_AssetFact

This SIL map is used to populate the final extension table for the Asset fact. The different transformations are the following:

- **SQ\_W\_ASSET\_FSX.** Note the SQL in the override. This joins with the dimension table W\_ACCOUNT\_D to get the ROW\_WID of the dimension row to be populated in the foreign key id column of the extension table.
- **LKP\_W\_ASSET\_F.** Retrieves the ROW\_WID of the parent fact table W\_ASSET\_F.
- **LKP\_W\_ASSET\_FX.** Gets the Insert/Update flag that is used by the update strategy to find out whether to insert or update this row.
- **EXPTRANS.** Gets ETL\_PROC\_WID using the lookup LKP\_ETL\_PROC\_WID from table W\_PARAM\_G. This table gets populated in the beginning of Refresh Siebel DW. Also generates Insert/Update flag based on the previous transform LKP\_W\_ASSET\_FSX.
- **UPDTRAN.** Updates or inserts a row based on the flag.

## Customization Steps for Siebel Data

This section includes information about adding columns and tables to the Siebel Data Warehouse using data from the Siebel OLTP. It includes the following topics:

- [“Adding a Column From a Main Source Table in the Siebel OLTP” on page 175](#)
- [“Adding a Column From a Related Source Table in the Siebel OLTP” on page 175](#)
- [“Modifying an Existing Column in the Siebel Data Warehouse” on page 177](#)
- [“Adding a New Dimension in the Siebel Data Warehouse and Using It with an Existing Fact” on page 177](#)
- [“Adding a New Fact Table in the Siebel Data Warehouse” on page 179](#)
- [“Adding a New Dimension Table for a New Fact Table in the Siebel Data Warehouse” on page 180](#)

## Adding a Column From a Main Source Table in the Siebel OLTP

Example:

- Adding *race* in W\_PERSON\_D because it is not in the supplied model. The value of race is sourced from S\_CONTACT, which is already the main source table for W\_PERSON\_D.
- Adding measure XYZ in W\_ORDERITEM\_F because it is not in the supplied model. The value of XYZ comes from S\_ORDERITEM, which is the main source table for W\_ORDERITEM\_F.

### To add a column from a main source table in the Siebel OLTP

- 1 Choose one of the following options:
  - Use predefined columns in an existing extension table.
  - Create a new extension table with appropriate columns.
- 2 Create a new map (SDE and SIL) to load the extension table column, based on the provided templates (CustomTemplateSDE\_PersonDimension).
- 3 Create new individual workflows for the new mappings in the Custom folder.
- 4 For extract mappings (SDE), use views as the main source of information.
- 5 Create new tasks in the DAC repository, one for SDE and the other for SIL.
- 6 Make the SDE task depend on any auxiliary workflows, so that it waits for all the auxiliary processing for the base table to complete.
- 7 For the CustomSIL mapping make SDE mapping and the SIL\_PersonDimension as the Depends On tasks. This is because you cannot execute CustomSIL workflow until the CustomExtract is completed. Also, because the primary key lookup has to be performed, this task needs to wait on SIL\_PersonDimension.

## Adding a Column From a Related Source Table in the Siebel OLTP

- Adding ABC in W\_PERSON\_D because it is not in the supplied model. The value of ABC is sourced from S\_CONTACT\_X, which is not the main source table for W\_PERSON\_D.

- Adding measure *XYZ* in *W\_ORDERITEM\_F* because it is not in the supplied model. The value of *XYZ* comes from *S\_DOC\_ORDER*, which is not the main source table for *W\_ORDERITEM\_F*.

#### **To add a column from a related source table in the Siebel OLTP**

- 1 Choose one of the following options:
  - Use predefined columns in an existing extension table
  - Create a new extension table with appropriate columns.
- 2 Create new mappings and workflows (SDE and SIL) to load the extension table column based on the provided templates. The related table should be joined in the Source Qualifier's SQL. Even though additional information is being extracted from a related table in the SQL override, always use the view created on the base table to extract information.
- 3 If necessary (for the new table), write the auxiliary change capture mappings so that whenever a row changes in the source table for the new column, the corresponding row in the main table is also marked changed, by inserting rows in the Incremental Row Image table.
- 4 Create a new workflow for the new map and register the task in the DAC repository. Create the dependencies.
- 5 Verify if the related table is registered in the DAC. If an entry is not there, you need to create a new record for the table and assign it an image suffix. If the table is registered, make sure there is an image suffix assigned for this table.
- 6 If an image suffix does not exist for the table under consideration, assign one. The image suffix should be three characters long. The recommended naming convention is to start with C. For example, use C1, C2, CA1, CA2, and so on. Make sure this suffix is not used by other tables. The DAC client does not validate this information when entering data.
- 7 Create Image tables in the OLTP. Right-click on the table record in the DAC client, and choose Generate Change Capture Scripts. This will assist you with creating the image tables, necessary indices, and triggers if you plan on tracking deletes on the OLTP database. Execute these scripts on the OLTP.
- 8 For the task that is created for the extract process set the Build Image flag to True and select the new table as an auxiliary table.



## Modifying an Existing Column in the Siebel Data Warehouse

- Modifying *Full Name* in W\_PERSON\_D with a name in a different format.
- Modifying *Tax Total* and XYZ in W\_ORDERITEM\_F with different tax calculation logic.

### **To modify an existing column in the data warehouse**

- 1 Do not modify the existing column.
- 2 Use extension table columns for the new value.
- 3 Refer to the steps for adding an extra column.

## Adding a New Dimension in the Siebel Data Warehouse and Using It with an Existing Fact

- Adding a new custom dimension WC\_NEWDIM\_D and linking it to W\_REVN\_F.

Reference:

- SDE and SIL\_WaveDimension, SDE and SIL\_AssetFact

### **To add a new dimension and use it with an existing fact table**

- 1 Create a DDL for the new dimension based on the standard structure (with appropriate system columns). Create a staging table for this dimension.
- 2 Register the new source table (if it does not already exist) in the DAC repository and associate it with the appropriate database connection.
- 3 Assign it an image suffix if you plan on incrementally loading this table.
- 4 Create a new custom map SDE\_XYZ to populate the dimension stage. Instead of the actual source table (for example S\_ABC), use the view that will be generated by the change capture process (for example V\_ABC) in the SQL so that it extracts only the incremental data. Use existing reference maps as examples of how to populate the system columns. Make sure you truncate the stage table in corresponding tasks.

- 5** Create a new custom map SIL\_XYZ to populate the new dimension from the stage table. Use the above referenced map as example for how to populate the system columns.
- 6** Make the following changes to fact loading mapping to link the new dimension:
  - a** Do one of the following:
    - Use the extension table's predefined fact table foreign keys that join to this new dimension. (In the above example, the extension table for W\_REVN\_F must be used to store the foreign key of this new dimension.)
    - Create a new extension table for the fact table with predefined structure to hold a new foreign key, if the foreign keys in existing the extension table have been exhausted.
  - b** Create new mappings (SDE and SIL) to populate the extension table columns of the fact extension table.
- 7** Register the new dimension table in the DAC and associate it with the appropriate database connection.

If you are planning to build a new dimension incrementally, assign an image suffix to the source table. For instructions, see [“Adding a Column From a Related Source Table in the Siebel OLTP” on page 175](#).

- 8** Add the new dimension to all the fact groups that may reference it.
- 9** Register the workflows as tasks in the DAC.
- 10** For SDE mapping of the dimension make sure you set the Build Image flag to True, and the Truncate Always option to True. And in the list of source tables, mark the primary/auxiliary source(s) of this dimension.
- 11** For SIL workflows of the dimension make sure you set only Truncate for Full Load option to True.
- 12** Make SIL\_XYZ depend on SDE\_XYZ.
- 13** Make CustomSIL\_RevenueFact depend on load dimension task (SIL\_XYZ) and load base fact task (SIL\_RevenueFact).

## Adding a New Fact Table in the Siebel Data Warehouse

- Creating a new custom WC\_PROJECT\_F by sourcing data from S\_XYZ table.

Reference:

- SDE and SIL\_AssetFact.

### **To add a new fact table**

- 1** Create a DDL for the new fact based on the standard structure (with appropriate system columns). Create a staging table for this fact.
- 2** Register the new source table (if it does not already exist) in the DAC repository and associate it with a database connection.
- 3** Assign it an image suffix.
- 4** Create the change capture tables by right-clicking and selecting Generate Change Capture Scripts.
- 5** Create SDE mappings to populate the custom stage table. Use the view created by change capture as the main table in the SQL so that it extracts only the incremental data. Use the reference maps (above) as examples of how to populate the system columns. Be sure to truncate the stage table in corresponding workflows.
- 6** Create SIL mapping to populate the custom fact table. Use reference maps as examples of how to populate the system columns.
- 7** Use lookups or SQL override joins to dimension tables for populating dimension foreign keys (ROW\_WIDs) pointing to the existing dimension.
- 8** In the DAC, register the target tables.
- 9** Because this is a fact table, create a new table group with this fact table as the main table.
- 10** Associate the fact table to the group with the relationship as Self.
- 11** Associate all the dimension tables.
- 12** Add the newly created group to the appropriate subject area.
- 13** Create new tasks for the workflows.

- 14** For the SDE task, make sure you have the Build Image flag set to True, and list all the source tables that it queries from. Choose one or more tables as primary or auxiliary. For the target tables choose the staging table. Set the Truncate Always flag to True.
- 15** For the SIL task, list all the dimensions that will be required under source tables, and make it depend on the SDE task of the custom fact table and SIL tasks of all the dimension tables.

## Adding a New Dimension Table for a New Fact Table in the Siebel Data Warehouse

- Adding a new custom dimension WC\_NEWDIM\_D and linking it to new custom WC\_BKLOG\_F fact table.

### ***To add a new dimension table for a new fact table***

The steps for creating a new dimension table are similar to the steps for incremental change capture.

- 1** In the new custom fact loading mapping (SIL), use lookups for getting foreign keys to the new dimension.
- 2** Use existing maps as examples.

## Siebel Data Warehouse Customization Steps for Non-Siebel Data

This section includes information about adding non-Siebel data to the data warehouse. It includes the following topics:

- [“Adding New Data as a Whole Row into the Standard Dimension Table in Siebel Data Warehouse” on page 181](#)
- [“Adding New Data as a Whole Row into the Standard Fact Table in Siebel Data Warehouse” on page 184](#)
- [“Adding a Column in a Standard Dimension Table in Siebel Data Warehouse” on page 189](#)

- [“Adding a Column for a Standard Fact Table in Siebel Data Warehouse” on page 191](#)
- [“Building a Custom Dimension Table from External Sources in Siebel Data Warehouse” on page 192](#)
- [“Linking the Custom Dimension to a Standard Fact Table in Siebel Data Warehouse” on page 193](#)
- [“Building Custom Fact Tables from External Sources in Siebel Data Warehouse” on page 194](#)

## Adding New Data as a Whole Row into the Standard Dimension Table in Siebel Data Warehouse

Example: Loading Contact Data From an External Source

- New DDLs: None
- New Mappings: SDE Mappings for dimensions, SDE and SIL mappings for related fact tables
- Case: Bringing data into dimension tables
- Reference Mappings: SDE and SIL\_WaveDimension mappings

### **To add new data as a whole row into the standard dimension table**

- 1** Identify and understand the existing structure of staging tables. Refer to *Siebel Data Warehouse Data Model Reference* for the table structure. Non-system columns can include the null value.
- 2** Create a custom SDE mapping to load the data into the staging table in the Custom\_DW\_Folder for this purpose. The staging table needs to be populated with incremental data (rows that have been added or changed since the last Refresh ETL process), for performance reasons.

- 3 Populate the INTEGRATION\_ID column with the unique identifier for the record.

The combination of INTEGRATION\_ID and DATASRC\_NUM\_ID is unique. When importing the data, make sure that a unique identifier for the external data source is inserted in the DATASRC\_NUM\_ID column. The DATASRC\_NUM\_ID is set to 1 for mappings that source data from the Siebel OLTP. This is a reserved value and is used in all standard mappings. For example, a value of 2 can be defined for DATASRC\_NUM\_ID in the custom SDE mapping. The standard SDE mappings populate the INTEGRATION\_ID column of the dimension staging table (used for resolving the dimension's Siebel OLTP ROW\_ID value). The custom process must be used to populate the same column with a unique identifier from the external data source.

- 4 After the data is populated in the staging table, use the standard SIL mappings to populate the dimension target tables.

- 5 Use one data source per dimension.

This is a best practice. To consolidate data for a dimension row from multiple data sources, complete that process prior to the custom SDE mapping process. For example, to consolidate contacts from Siebel and Peoplesoft applications, complete the consolidation process (including removing duplicates, cleansing, and so on) before loading the contact list into the Siebel staging table.

- 6 Modify the SDE and SIL mappings of all the related fact tables (fact tables that need to be linked to this dimension).

The custom fact SDE mappings must populate the foreign key column of the changed dimension (using a custom map table process to convert from Siebel's row IDs to the external data source row IDs). The custom SIL mapping should be modified to use the appropriate DATASRC\_NUM\_ID, because the standard SIL mappings assume DATASRC\_NUM\_ID for the dimensions are the same as the fact table's DATASRC\_NUM\_ID.

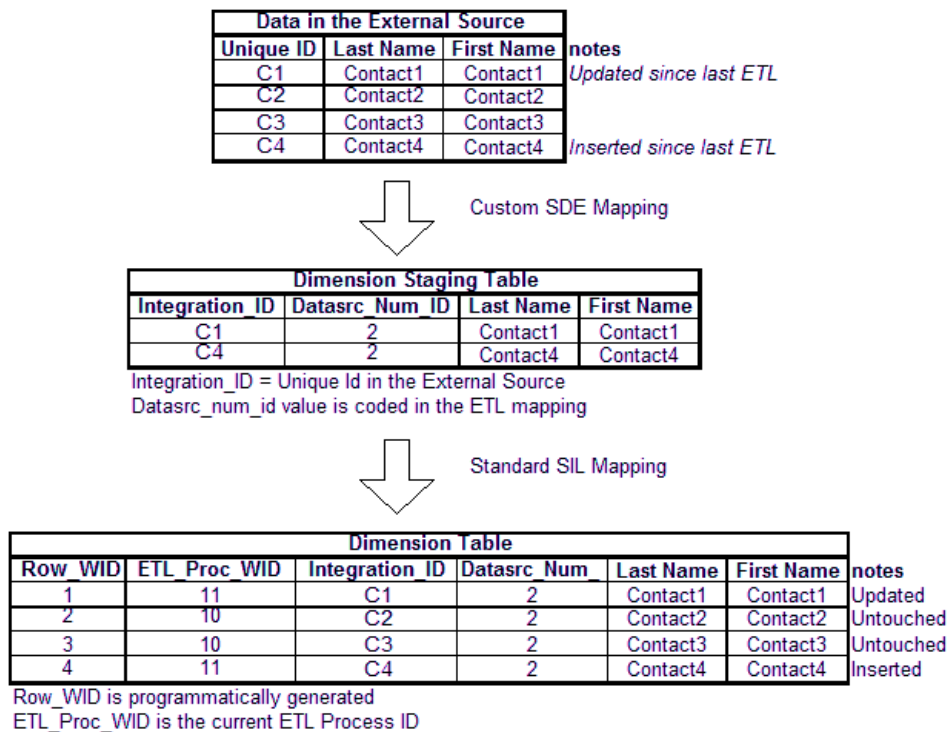
- 7** It is very important to decide when the data is going to be loaded. If it is going to be loaded along with the Siebel source data, you must be careful with how you handle failure recovery. The preconfigured workflows truncate the target staging table prior to loading. Upon failure, when the DAC server restarts the task, all the data is truncated and all the data is loaded again.

If the data from the external source gets loaded into the same staging table, be careful with how you handle this situation, since you cannot use the truncate table functionality. The data migrating into the staging table is not incrementally loaded, and, therefore, should be cleaned up prior to attempting to load this table again.

In such a case, it is recommended that you encapsulate the extract part from both the sources inside an Informatica workflow. Upon failure of either of the extracts, the whole workflow gets rerun. Note that the data from both the sources should be run at the same time, all the time.

If it is decided that the data is going to be loaded at different time frequencies, then the new SDE workflows need not depend on the preconfigured SDE workflows and can use the Truncate Table option for failure recovery. In this case, in the DAC Design view, define a new execution plan in the Execution Plans tab, and define the new data source in the Database Connections subtab. Make sure the shared SIL process depends on the SDE processes from both sources.

[Figure 11](#) provides a graphical representation of this process.



**Figure 11. Adding New Data as a Whole Row into the Standard Dimension Table**

## Adding New Data as a Whole Row into the Standard Fact Table in Siebel Data Warehouse

Example: Bringing in activity data from external sources.

- New DDLs: None
- New Mappings: SDE and SIL mappings
- Case: Bringing data into fact tables.
- Reference: SDE and SIL\_AssetFact



**To add new data as a whole row into a standard fact table**

- 1** Identify and understand the existing structure of staging tables. Refer to *Siebel Data Warehouse Data Model Reference* for the table structure.  
  
Fact tables contain foreign keys to existing dimension tables. These dimensional key columns in the fact table need to be populated with the correct ROW\_WID values from the dimension tables. For dimensions that are populated with Siebel data, identify the correct ROW\_WID by looking up the INTEGRATION\_ID in the dimension table. For example, to load Order Item details from SAP, the Order Item Fact staging table has the PROD\_ID column that needs to be resolved from the main Siebel data source. Looking up the ROW\_WID of the S\_PROD\_INT table later resolves the foreign key to product dimension.
- 2** Develop custom SIL mappings to handle the specific situation required. The reference SIL mapping can be copied to the Custom folder and customized for specific requirements. If the SIL mappings were copied as a base from the standard folder, it is possible to leverage the existing logic for ROW\_WID generation, update strategy, and so on. It may be easier to trim an existing SIL mapping than to create a new one. Apply discretion in creating a custom SIL mapping.
- 3** While joining with the dimensions in the SIL process, use the DATASOURCE\_NUM\_ID and INTEGRATION\_ID to resolve foreign key values. Note that for dimensions populated entirely from Siebel, DATASOURCE\_NUM\_ID will be 1.
- 4** For the dimensions populated entirely from other external sources, use the appropriate DATASRC\_NUM\_ID (not 1) and the corresponding INTEGRATION\_ID. This needs to be hard coded in the custom mapping.
- 5** Use an update strategy that would update the existing records and insert new ones. Inspect SDE and SIL mappings provided as reference to understanding the programmatic logic.
  - **Scenario 1.** All the dimensions get data from the external source as the fact table. In this case it is not necessary to change the existing standard SIL mappings. The standard SIL process assumes that the dimension rows that a fact needs to join are from the same data source as the fact rows. (See [Figure 12](#) for an example.)

- Scenario 2: Some dimensions are populated from external sources and some from the Siebel OLTP. In this case, standard SIL mappings need to be modified so that the appropriate DATASRC\_NUM\_ID is used when joining to a dimension table. This scenario needs to be custom designed and developed. As in any customization, it will require ongoing maintenance. (See [Figure 13](#) for an example.) Copy the mapping over to the Custom folder and do the modifications. Create a workflow that points to the mapping in the Custom folder. In the DAC repository, search for the task that uses this workflow as a command and change the folder to Custom folder. Change the name as necessary.

---

**NOTE:** Changing the standard mapping is an exception. It is important to document changes as updates to standard mapping will be lost during the Informatica repository process. Reapply changes to the standard mappings after the upgrade process. Prior to modifying copy the mapping as-is from OTB folder to the Custom folder, and make changes in the Custom folder. In the DAC repository, find the task that is using the workflow name and change the folder to point to the Custom folder. Adding new data as a whole row into the standard Fact table: Scenario 1.

---

Data in the External Source - Activity				
Unique Id	Activity No.	Contact	Product	Fact
Act1	Act1	C1	CP1	100
Act2	Act2	C2	CP2	200
Act3	Act3	C2	CP2	300
Act4	Act4	C4	CP1	400

New data to be loaded into Activity Fact



Custom SDE Mapping

Activity Fact Staging Table				
Integration_ID	Datascr_Num_ID	Contact_Id	Product_Id	Fact
Act1	2	C1	CP1	100
Act2	2	C2	CP2	200
Act3	2	C2	CP2	300
Act4	2	C4	CP1	400

Contact Id = Unique Ids as-is from external data source

Product Id = Unique Ids as-is from external data source



Standard SIL Mapping

Activity Fact Table						
Row_WID	ETL_Proc	Integration_ID	Datascr_Num_ID	Contact_WID	Product_WID	Fact
100	11	Act1	2	1	101	100
101	11	Act2	2	2	102	200
102	11	Act3	2	2	102	300
103	11	Act4	2	4	101	400

Can reuse OTB SIL mapping in this scenario

Row\_WID is programmatically generated

ETL\_Proc\_WID is the current ETL Process ID

Existing Dimension Tables - needed for the dimension Row\_WIDs

Product Dimension Table.				
Row_WID	ETL_Proc_WID	Integration_ID	Datascr_Num_ID	Product Name
101	10	CP1	2	Product1
102	10	CP2	2	Product2

Contact Dimension Table					
Row_WID	ETL_Proc_WID	Integration_ID	Datascr_Num_ID	Last Name	First Name
1	11	C1	2	Contact1	Contact1
2	10	C2	2	Contact2	Contact2
3	10	C3	2	Contact3	Contact3
4	11	C4	2	Contact4	Contact4

Figure 12. Adding New Data as a Whole Row into the Standard Fact Table - Scenario 1

## Customizing the Siebel Data Warehouse

### Siebel Data Warehouse Customization Steps for Non-Siebel Data

Data in the External Source - Activity				
Unique Id	Activity No.	Contact	Product	Fact
Act1	Act1	C1	CP1	100
Act2	Act2	C2	CP2	200
Act3	Act3	C2	CP2	300
Act4	Act4	C4	CP1	400

New data to be loaded into Activity Fact



Custom SDE Mapping

Activity Fact Stage Table				
Integration_ID	Datasrc_Num_ID	Contact Id	Product Id	Fact
Act1	2	C1	P1	100
Act2	2	C2	P2	200
Act3	2	C2	P2	300
Act4	2	C4	P1	400

**Product\_ID is Siebel's Row\_ID - resolved via a custom reference table**

Integration Id = Unique Id of External Source

Datasrc\_num\_id value is coded in the ETL mapping



Changed Standard SIL Mapping

Activity Fact Table						
Row_WID	ETL_Proc_WID	Integration_ID	Datasrc_Num_ID	Contact_WID	Product_WID	Fact
100	11	Act1	2	1	101	100
101	11	Act2	2	2	102	200
102	11	Act3	2	2	102	300
103	11	Act4	2	4	101	400

Needs custom SIL mapping

Existing Dimension Tables - needed for the dimension Row\_WIDs

Product Dimension Table.				
Row_WID	ETL_Proc	Integration_ID	Datasrc_Num_ID	Product Name
101	10	P1	1	Product1
102	10	P2	1	Product2

Data entirely from Siebel

Contact Dimension Table						
Row_WID	ETL_Proc_WID	Integration_ID	Datasrc_Num_ID	Last Name	First Name	
1	11	C1	2	Contact1	Contact1	
2	10	C2	2	Contact2	Contact2	
3	10	C3	2	Contact3	Contact3	
4	11	C4	2	Contact4	Contact4	

**Figure 13. Adding New Data as a Whole Row into the Standard Fact Table - Scenario 2**

In the [Figure 13](#) because the contact data and product data are from different data sources, it is necessary to modify the SIL session and the SQL override in the following manner:

```

SELECT
    ACT.INTEGRATION_ID,
    P.ROW_WID,
    C.ROW_WID
FROM
    W_ACTIVITY_FS ACT,
    W_CONTACT_D C,
    W_PRODUCT_D P
WHERE
    ACT.PRODUCT_ID = P.INTEGRATION_ID
    AND P.DATASRC_NUM_ID = 1 /* Product is from Siebel Source having
datasrc_num_id = 1
ln dimension */
    AND ACT.CONTACT_ID = C.INTEGRATION_ID
    AND ACT.DATASRC_NUM_ID = C.DATASRC_NUM_ID
/* The rows in Contact dimension that the fact refer to are from
the same external datasrc, 2 */

```

## Adding a Column in a Standard Dimension Table in Siebel Data Warehouse

Example: Loading an attribute called SKU for existing products that have already been loaded.

- New DDLs: New extension tables or preconfigured extension tables
- New Mappings: SDE and SIL mappings for extension tables

- Case: Bringing data in for dimension tables
- Reference: CustomTemplateSDE and SIL\_WaveDimension in Custom\_DW\_Folder

### **To add a column in a standard dimension table**

- 1** Add a column either in the standard or custom extension tables. There should be a 1:1 relationship with the parent table. Do not modify the standard tables, because doing so will affect upgrades.
- 2** The custom stage table should be populated with incremental data. This is an important performance consideration. It requires a process on the source side that detects new or modified records. This allows custom SDE mappings to process incremental data during each Refresh ETL execution.
- 3** While populating the extension table, design lookups (or write SQL overrides) for the SQL Qualifier Transformation in the Informatica mapping to identify the primary key (ROW\_WID) of the parent row for the additional attribute. The primary key (ROW\_WID) of the parent row is then used as the primary key (ROW\_WID) of the extension table. Template mappings are provided in the CUSTOM\_DW\_REP folder (for example, CustomTemplateSDE\_WaveDimension').
- 4** The custom SDE mappings should populate the stage extension tables (standard or custom). The custom extension table for staging should include the INTEGRATION\_ID and DATASRC\_NUM\_ID. This combination enables the unique identification for each row by the data source. The data for this INTEGRATION\_ID column should be populated by a process that identifies such an ID from the parent data source. The DATASRC\_NUM\_ID column should have the same value of the DATASRC\_NUM\_ID of the parent it is extending.
- 5** Once stage data is populated, a custom SIL mapping can be developed to move data from the extension stage to the extension table (of either the factor dimension table). The process should contain a look up either based on the INTEGRATION\_ID and DATASRC\_NUM\_ID to find the ROW\_WID for the extension table from the parent fact/dimension table. For an example, refer to CustomTemplateSIL\_WaveDimension.
- 6** If there are rows for which there is no matching ROW\_WID in the parent tables, they should be ignored. Use the appropriate update strategy for modifying existing data.

- 7 Make the custom SIL process depend on the custom SDE process and SIL process of the dimension table you are extending, because this mapping will require the borrowing of the foreign key for the link.

Figure 14 provides an example of a product dimension that already exists and the addition of the attribute SKU.

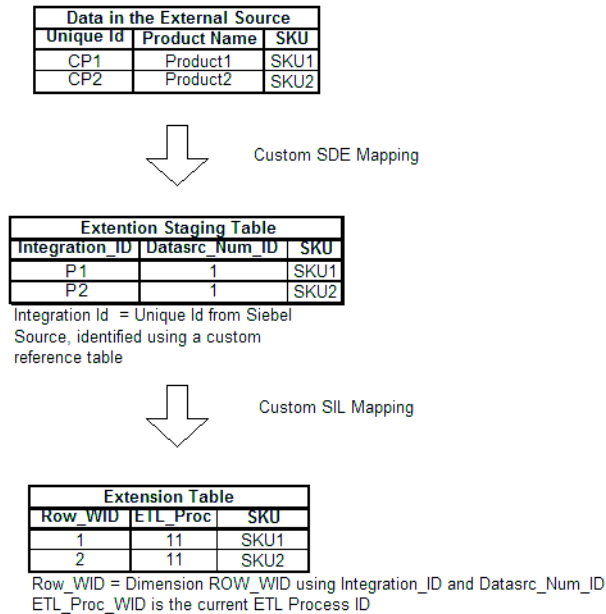


Figure 14. Adding a Column in a Standard Dimension Table

## Adding a Column for a Standard Fact Table in Siebel Data Warehouse

Example: Loading shipping order information into the data warehouse.

- New DDLs: Custom or standard extension tables.

- New Mappings: SDE and SIL mappings.
- Case: Bringing data in for fact tables.
- Reference: CustomSDE and SIL\_AssetFact in Custom\_DW\_Folder.

### **To add a column for a standard fact table**

- 1** Follow the steps in [“Adding a Column in a Standard Dimension Table in Siebel Data Warehouse” on page 189.](#)
- 2** Use the CustomTemplateSDE\_AssetFact, and CustomTemplateSIL\_AssetFact as examples.

If the additional columns include foreign keys to dimension tables, consider the following:

- While joining with the dimensions in the SIL process, use the DATASRC\_NUM\_ID as 1 for the dimensions that have only data from the main Siebel source and join based on INTEGRATION\_ID.
  - For the dimensions that have only data from external sources use the appropriate DATASRC\_NUM\_ID and the corresponding INTEGRATION\_ID.
  - Use the appropriate update strategy to change and insert rows.
- 3** Quite likely the SDE process for this will not depend on anything else.
  - 4** Make the Custom SIL process to depend on its custom SDE process and SIL process of the fact table you are extending.

## **Building a Custom Dimension Table from External Sources in Siebel Data Warehouse**

- New DDLs: New staging and new dimension tables.
- New Mappings: Dimension SDE and SIL mappings.
- Case: Loading data for custom dimension tables.
- Reference: SDE and SIL\_WaveDimension in Siebel\_DW\_Folder.



**To build a custom dimension table from external sources**

- 1 Follow the steps in “[Adding a New Dimension Table for a New Fact Table in the Siebel Data Warehouse](#)” on page 180. Use a different DATASRC\_NUM\_ID while populating the staging table.
- 2 Populate the custom table with incremental data. This is an important performance consideration that requires a process on the source side that detects new or modified records.
- 3 Make sure the SIL process takes care of the update strategy when loading data incrementally. Look for examples of how it is done by inspecting any of the SIL mappings in the standard repository like SIL\_WaveDimension.
- 4 In the DAC, make the SIL process depend on SDE process for this custom table.
- 5 Register the table.
- 6 Add it to appropriate table groups.

**Linking the Custom Dimension to a Standard Fact Table in Siebel Data Warehouse**

- New DDLs: New fact extension tables or standard fact extension tables
- New Mappings: Fact SDE and SIL mappings
- Case: Loading data for fact tables that reference the new custom dimension tables
- Reference: CustomSDE and SIL\_AssetFact in Custom\_DW\_Folder

**To link the custom dimension to a standard fact table**

- 1 Extend the fact tables by creating or using fact extension tables.
- 2 A custom SDE mapping has to be designed to populate the additional foreign key column in the extension stage table. Make sure that the dimensions ID in the fact staging is the same as the integration of the custom dimension. This may require some preprocessing.
- 3 A custom SIL process loads data from the staging table to the fact extension table.

- 4 Develop the SQL override for populating the fact extension table to extract foreign keys for the new dimension table. Specifically, write the query to extract the ROW\_WID of the custom dimension table using INTEGRATION\_ID and DATASRC\_NUM\_ID of the dimension in combination.
- 5 Ignore the rows for which there is not a matching ROW\_WID value.
- 6 In the DAC, register the SDE and SIL tasks. Make the SIL mapping depend on its corresponding SDE mapping and the SIL mapping of the base fact table.
- 7 Add the custom dimension to the fact table group.

## Building Custom Fact Tables from External Sources in Siebel Data Warehouse

- New DDLs: New staging and new fact tables
- New Mappings: SDE and SIL mappings
- Case: Loading data for new fact tables
- Reference: SDE and SIL\_AssetFact from Siebel\_DW\_Rep Folder

### **To build custom fact tables from external sources**

- 1 Follow the steps in [“Building a Custom Dimension Table from External Sources in Siebel Data Warehouse”](#) on page 192.
- 2 In the custom fact SDE process, make sure that fact columns that need to reference standard Siebel dimensions contain ROW\_IDS from the Siebel data source.
- 3 Make sure that the dimension IDs in fact staging table are the same as the INTEGRATION\_ID in the corresponding dimension tables. This may require some preprocessing.
- 4 While creating a custom fact SIL process, make sure the SQL override statement joins with dimension tables. Use DATASRC\_NUM\_ID as 1 for the dimensions that have data from Siebel and join based on INTEGRATION\_ID.
- 5 For the dimensions that use data entirely from external sources, use the appropriate DATASRC\_NUM\_ID and the corresponding INTEGRATION\_ID. Use the appropriate update strategy to add new or change rows.

- 6** In the DAC, register the tasks. Make the SIL task depend on its corresponding SDE workflow and all the SIL tasks of its dimensions.

## **Customizing the Siebel Data Warehouse**

---

*Siebel Data Warehouse Customization Steps for Non-Siebel Data*

Exception reports provide insight about the source data used for the ETL processes. They provide information about the source data that can lead to erroneous results in the Siebel Data 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.

## Understanding Siebel Data Warehouse Exceptions

Exception reports are defined for the following components:

- **List of Values.** Identifies gaps and overlaps for certain LOV types.
- **Cost Lists.** Identifies products for which the cost lists have not been defined. Identifies the cost lists which define costs for a product in a certain currency for overlapping periods of time.
- **Exchange Rates.** Currency Exchange rates that do not change over a period of time. If exchange rates are not defined for more than 30-day intervals, then they are flagged as an exception.
- **Hierarchies.** Entities that have circular references are flagged as exceptions. The Siebel Data Warehouse supports 10 levels of hierarchies. If there are entities that have more than 10 levels of hierarchies defined, they are flagged as exceptions.

### List of Values Exceptions

List of Values include High and Low values that can be used as bucket values in categories for effective analysis. If these values are not contiguous (such as gaps or overlaps in defined values), the ETL process cannot accurately categorize the values.

An example of List of Values exceptions is shown in [Table 33](#).

**Table 33. List of Values Exceptions (Example)**

Type	Name	Low	High
MY_TYPE	< 100	10	100
MY_TYPE	80 – 200	80	200
MY_TYPE	250 – 300	250	300
MY_TYPE	350 – 400	350	400

Notice that, in the example for Type MY\_TYPE, there are overlaps and gaps between records. There is an overlap of ranges between the first and second row. There is a gap between second and third row, and between third and fourth rows.

The following LOV types are analyzed for List of Value Exceptions:

- ACCNT\_REVENUE\_SIZE
- ACCNT\_EMP\_SIZE
- LEAD\_AGE\_DAYS
- OPTY\_REVENUE\_SIZE
- OPTY\_UNIT\_SIZE
- ACCNT\_REVENUE
- QUOTE\_AGE\_DAYS
- ACCNT\_REVN\_GROWTH
- APPROVAL\_AUTH\_SIZE
- SR\_CHART\_AGE
- ASSET\_COST\_CATEGORY

## 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 Siebel Data Warehouse.

An example of Cost List exceptions is shown in [Table 34](#).

**Table 34. Cost List Exceptions (Example)**

Cost List	Product Name	Currency	Start Date (MM-DD-YYYY)	End Date (MM-DD-YYYY)	Cost
Cost List 1	Product 1	USD	01-01-2000	12-31-2000	10.00
Cost List 2	Product 1	USD	06-01-2000	06-01-2001	12.00
Cost List 3	Product 1	USD	06-01-2001	06-01-2002	13.00

In the example, Cost List 1 and 2 have definitions of cost overlapping over 06-01-2000 to 12-31-2000.

## Products Without a Cost List

During the ETL process, the costs of the products are calculated based on the Cost List table. If the cost lists are not defined correctly, the cost of the products cannot be calculated correctly in the Siebel Data Warehouse. This exception mapping queries the product table and looks for a minimum of one cost list to be defined. The products with no cost list definition are flagged as exceptions.

## Exchange Rate Exceptions

The Siebel Data Warehouse supports transactions in many different currencies. Siebel Analytics converts all currencies in the Siebel Data 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 Siebel transactional database. If the currency exchange rates do not change for a period of 30 days, then Siebel Analytics 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 Siebel Data Warehouse.

---

**NOTE:** Exchange rates are derived from records that are of type “Daily” in the Siebel transactional database. If any other types have been defined, they are not handled without some customization.

---

### Invalid Hierarchy Exceptions

Accounts, divisions, products, and opportunities can all have hierarchical relationships. These entities are denormalized within the Siebel Data Warehouse database to a fixed number of levels. Siebel Analytics supports up to ten hierarchies levels in the Siebel transactional database. If the depth of hierarchies extends beyond this number, results become inconsistent or incomplete.

### Circular Hierarchy Exceptions

Circular Hierarchies arise when the parent-child relationship has circular references. For example:

Child	Parent
A1	A2
A2	A1

Siebel Analytics flags exceptions for two levels. Circular references over two hierarchies are not flagged. For example:

Child	Parent
A1	A2
A2	A3
A3	A1



These produce infinite levels of hierarchies. The same records will be captured under the Invalid Hierarchy exceptions as their hierarchy depths will increase beyond 10 levels.

## Executing Siebel Data Warehouse Exception Reports

Before loading the Siebel Data Warehouse for the first time and for the subsequent refreshes, you should plan to spend time cleansing your Siebel 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 Siebel 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 Siebel transactional database, repeat these actions until the exception report is empty:

- In the DAC, run the Exception Reports execution plan.
- In the Siebel user interface, navigate to Analytics Administration > Exception Reports.

---

**NOTE:** In Siebel Financial Services, this screen is called DataMart Administration.

---

- For every line in the ETL Exception Reports list, fix the cause of the problem. For information on fixing problems, see [“Cleansing Data.”](#)

### Cleansing Data

Use a combination of the Exception Reports and the Diagnostic views to assess changes that need to be made external to the Siebel transactional database, and changes to the Siebel transactional database directly.

The ETL Exception Reports list contains one record for each exception flagged in the Siebel transactional database. The ETL Exception Explanation form, located below the ETL Exception Reports list, describes the selected exception, its effect on the Siebel Data Warehouse building process, and offers suggestions for repairing the data.

***To fix an exception***

- 1** Select an exception record.
- 2** Read and understand the text in the ETL Exception Explanation form.
- 3** Click the report link.

The object's data appears. (For example, if the object is an account, then the Account form appears. If the object is a cost list, then the Cost List list appears.)

- 4** Repair the problem, using the text in the ETL Exception Explanation form as a guide.
- 5** Return to the ETL Exception Reports list and place a check mark in the Fixed column to indicate to others that this exception has now been fixed.

## Using the List of Values View

Use the List of Values view, shown in Figure 15, to visually compare how the list of values data extracted from the Siebel transactional database coordinates with the values loaded into the Siebel Data Warehouse. The ETL process removes duplicates and overlaps and fills data gaps. Values are extended to span the List of Values (LOV) minimum and maximum values. Duplicates, Range Gaps, and Overlaps are flagged by the exception reports.

ETL History   List of Values   Exchange Rates   Cost Lists						
Query   Clear Cache   1 - 7 of 7+						
Type	Language Independent Code	Display Value	Low	High	Order	Language Name
ABS_COST_FCT_AREA	Field Service	Field Service			1	English-American
ABS_COST_FCT_AREA	Professional Services	Professional Services			2	English-American
ABS_COST_FCT_AREA	Workforce Management	Workforce Management			3	English-American
ABS_COST_FCT_TYPE	Normal	Normal			1	English-American
ABS_COST_FCT_TYPE	Emergency	Emergency			2	English-American
ABS_COST_FCT_VARIABLE	Constraint Violation	Constraint Violation			1	English-American
ABS_COST_FCT_VARIABLE	FSE Overtime	FSE Overtime			2	English-American

List Of Values (Data Warehouse)						
Query   1 - 7 of 7+						
Type	Language Independent Code	Display Value	Low	High	Order	
ABS_COST_FCT_AREA	Field Service	Field Service	0	9,999,999,999	1	
ABS_COST_FCT_AREA	Professional Services	Professional Services	0	9,999,999,999	2	
ABS_COST_FCT_AREA	Workforce Management	Workforce Management	0	9,999,999,999	3	
ABS_COST_FCT_TYPE	Normal	Normal	0	9,999,999,999	1	
ABS_COST_FCT_TYPE	Emergency	Emergency	0	9,999,999,999	2	
ABS_COST_FCT_VARIABLE	Constraint Violation	Constraint Violation	0	9,999,999,999	1	
ABS_COST_FCT_VARIABLE	FSE Overtime	FSE Overtime	0	9,999,999,999	2	

Figure 15. List of Values List

The top List of Values list shows values from the Siebel transactional database and the bottom List of Values (Data Warehouse) list shows the data that is to be used in ETL process. You can edit the Siebel transactional database data directly in this view, but the Siebel Data Warehouse list is read-only.

**NOTE:** The List of Values is extracted into the Siebel Data Warehouse where the language is the same as the ETL Default Language set in the DAC ETL Preferences, or whose translate flag is set to “N,” or those that are active.

### Using the Exchange Rates View

Use the Exchange Rates view to diagnose currency translation issues in the Siebel Data 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 Siebel Data Warehouse values for active currencies and their exchange rates, and the bottom Exchange Rates (Data Warehouse) list shows the values loaded into the Siebel Data 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 Siebel transactional database.
- The exception reports flag any exchange rates to the ETL Base Exchange Currency that have not been defined within a specified period (30 days) in the DAC ETL Preferences.

### Using the Cost List View

Use the Cost List view to display the cost lists from the Siebel transactional database from the point of view of the product, and a read-only view of the values to be loaded into the Siebel Data 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 Siebel Data Warehouse.

- The exception reports flag products that do not appear in the Cost List list or have Cost List time gaps and overlaps.
- The Siebel Data Warehouse contains only one Cost List for a product and a currency at a time.

## **Using the ETL History View**

After all of the exceptions are corrected, the building of the data warehouse can be initiated. This view lists the history of the ETL processes and their statuses. When each ETL batch starts, the name of the process along with the timestamp is set, the status is set to STARTED. When the batch completes, its status is updated to COMPLETED.

## **Additional Exceptions**

The above mentioned exceptions are not an exhaustive list of all possible exceptions. Other exceptions are:

- The Analysis start and end date in the DAC ETL Preferences 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 ETL Preferences are crucial for the building of Day Dimension, flattening of Exchange Rates, Cost Lists, and KPI (Key Performance Indicator fact) calculations.
- The DAC ETL Preferences — 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 Siebel transactional database will not be translated.
- There must be exchange rates defined for the currencies your organization deals with. If the appropriate exchange values are not found, the ETL process uses the ETL Unknown Exchange Rate defined in the DAC ETL Preferences.

## **Using Siebel Data Warehouse Exception Reports**

*Executing Siebel Data Warehouse Exception Reports*

This section includes the following topics:

- [“Upgrade Requirements for Siebel Data Warehouse 7.7” on page 208](#)
- [“Siebel Data Warehouse 7.7 Product Changes and Customization Impacts” on page 208](#)
- [“Siebel Data Warehouse Upgrade Paths” on page 209](#)
- [“Approaches to Upgrading the Siebel Data Warehouse” on page 210](#)
- [“Process of Siebel Data Warehouse Upgrade” on page 211](#)
- [“Data Migration Workflows for Siebel Data Warehouse Data Upgrades” on page 213](#)
- [“Siebel Data Warehouse Upgrade Customizations” on page 216](#)
- [“Data Migration Workflows for Siebel Industry Applications Data Warehouse Upgrades” on page 221](#)

# Upgrade Requirements for Siebel Data Warehouse 7.7

A thorough understanding of Informatica, Siebel Data Warehouse, Siebel OLTP and Data Warehouse Application Console (DAC) is a prerequisite before proceeding with the upgrade.

## Siebel Data Warehouse 7.7 Product Changes and Customization Impacts

The changes in Siebel Data Warehouse mainly involve performance enhancements. Included are enhancements in the areas of change capture, reorganization of data, new data elements (like mini dimension), streamlining Informatica mappings, and so on. Your previous customizations may be affected by these changes. In these cases, reworking may be required to make sure that the customizations still function after the data warehouse is upgraded.

In addition to these enhancements, a new tool called the Data Warehouse Application Console (DAC) has been provided to efficiently manage Informatica sessions. All custom Informatica sessions need to be registered and managed using the DAC.

See the following list to identify whether or not user customizations will be impacted:

- All custom data warehouse tables, indexes, and Informatica sessions will need to be registered with the DAC
- Any reference to image tables (change capture mechanism)
- Any reference to the following data warehouse tables:
  - W\_MAP\_DIM
  - W\_ORDERITEM\_F
  - W\_QUOTEITEM\_F
  - S\_ETL\_SRC\_TABLE
  - S\_ETL\_I\_IMAGE



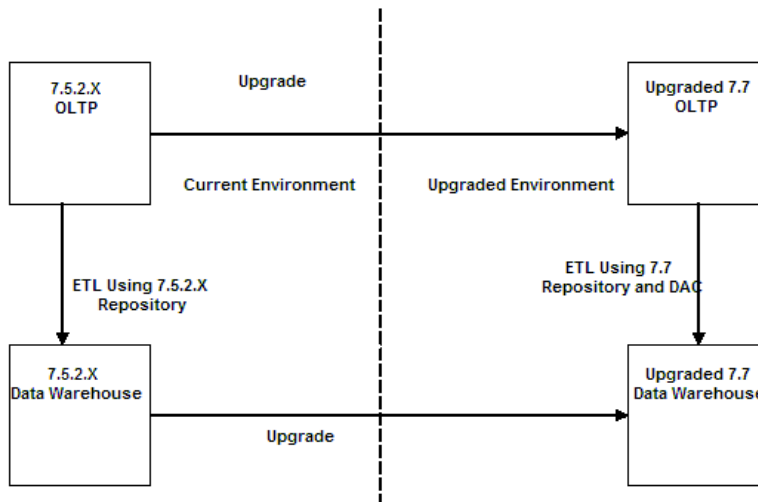
- S\_ETL\_R\_IMAGE
- Any reference to existing aggregates

After the customization impact is determined, an upgrade plan needs to be formalized for making the necessary changes. See [“Siebel Data Warehouse Upgrade Customizations” on page 216](#) for procedures.

## Siebel Data Warehouse Upgrade Paths

Following are the upgrade paths:

- **7.5.2.x to 7.7 Upgrade Path.** [Figure 16](#) shows the upgrade path to upgrade from an Analytics 7.5.2.x environment to Analytics 7.7.



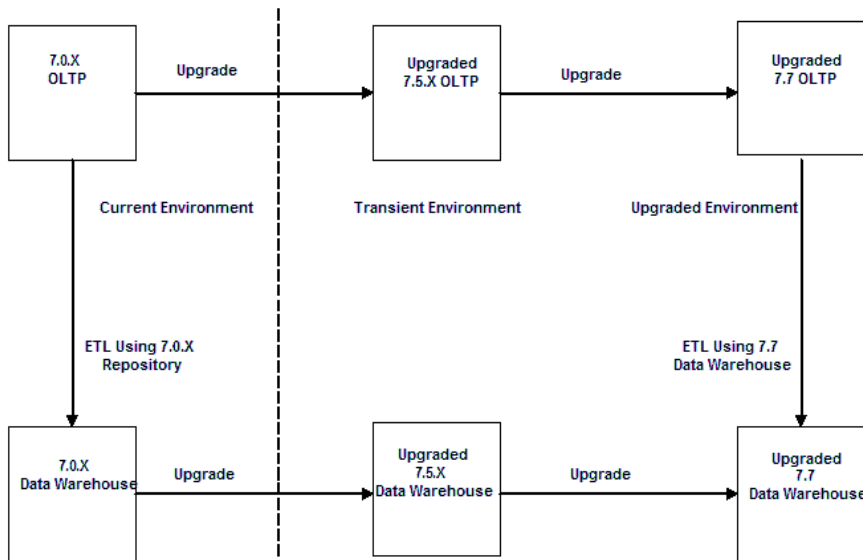
**Figure 16. Upgrade Path Analytics 7.5.2.X to Analytics 7.7**

---

**NOTE:** You must upgrade from Informatica 5.x to Informatica 6.2.1 before using the 7.7 Repository.

---

- **7.0.x to 7.7 Upgrade Path.** Figure 17 shows the upgrade path to upgrade from an Analytics 7.0.x environment to Analytics 7.7. The upgrade path from 7.0.x to 7.7 is a two-step process—first, you need to upgrade to 7.5.2.x and then use the procedures in this guide to upgrade to 7.7.



**Figure 17. Upgrade Path Analytics 7.0.X to Analytics 7.7**

---

**NOTE:** After upgrade, the new or updated data will go through fixed transformation. Old data might be different due to defects. The best way to avoid this issue is to build a new Data Warehouse instead of upgrading the Data Warehouse data.

---

## Approaches to Upgrading the Siebel Data Warehouse

There are two approaches to upgrading Siebel Data Warehouse, upgrading with Data Warehouse migration and upgrading without Data Warehouse migration.

**Upgrade with Data Warehouse data migration.** In this approach all the existing data in the Data Warehouse is upgraded using the 7.7 upgrade processes. The advantages of this approach are that you retain all your historical data, such as slowly changing dimension, pipeline, and so on. This process is faster than reloading the entire data again. To implement this kind of upgrade, see the procedures in [“OLTP Environment \(Analytics-Specific\) Upgrade” on page 212](#), and [“Data Migration Workflows for Siebel Data Warehouse Data Upgrades” on page 213](#).

**Upgrade without Data Warehouse data migration.** In this approach all the data from OLTP is reloaded again into the Data Warehouse (Full Load) using the 7.7 repository. The advantages of this approach are that you retain all the bug fixes, and the data is consistent. The disadvantages to this approach are that you would lose historical data like slowly changing dimension, pipeline, and so on. This process takes a longer time to complete than upgrading with data migration. To implement this kind of upgrade follow procedures listed in [“Process of Siebel Data Warehouse Upgrade” on page 211](#), [“Data Migration Workflows for Siebel Data Warehouse Data Upgrades” on page 213](#), and [“Running the Data Migration Workflow for Preconfigured Content in Siebel Data Warehouse” on page 215](#). After completing these procedures, run a full load from DAC.

## Process of Siebel Data Warehouse Upgrade

To upgrade Data Warehouse, the administrator typically performs the following tasks:

- [“OLTP Environment \(Analytics-Specific\) Upgrade” on page 212](#)
- [“Siebel Data Warehouse Environment Upgrade” on page 212t](#)
- [“Data Migration Workflows for Siebel Data Warehouse Data Upgrades” on page 213](#)
- Either of the following:
  - [“Running the Data Migration Workflow for Preconfigured Content in Siebel Data Warehouse” on page 215](#)
  - [“Running the Data Migration Workflow for Preconfigured Siebel Industry Applications Content” on page 222](#)

- [“Siebel Data Warehouse Upgrade Customizations” on page 216](#)

### **OLTP Environment (Analytics-Specific) Upgrade**

The Siebel eBusiness .sif files are located under the dwrep directory, and the performance\_hor.sif file contains last\_upd indexes. INCA\_OLTP.sif file contains the OLTP schema changes required for Analytics 7.7.

### **Siebel Data Warehouse Environment Upgrade**

Upgrading the Siebel Data Warehouse environment requires upgrading both the Informatica software and the Informatica repository.

**Informatica software upgrade.** See documentation from Informatica on how to upgrade to Informatica 6.2.1 and to upgrade your existing repository to Informatica 6.2.1. See *Siebel eBusiness Third-Party Bookshelf*.

---

**NOTE:** Back up your existing repository before you upgrade.

---

**Informatica repository upgrade.** The following is the high-level process to upgrade Informatica Repository. You must upgrade your software to Informatica 6.2.1, upgrade your current repository to Informatica 6.2.1, and install and configure DAC before you start this process. You also need an additional database for the new Siebel Analytics 7.7 Informatica repository.

- 1** Create and configure the new Siebel Analytics 7.7 Informatica repository using DAC.
- 2** Upgrade any customizations. See [“Siebel Data Warehouse Upgrade Customizations” on page 216](#) for information on how to upgrade customizations.

---

**NOTE:** See [Chapter , “Installing and Configuring Siebel Data Warehouse”](#) for installing and configuring DAC and step-by-step instructions to create and configure a new Informatica repository.

---

**Siebel Data Warehouse schema upgrade.** Upgrade the 7.5.x Siebel Data Warehouse schema using the ddlimp application.

---

**NOTE:** The ddlsme.ctl file can be found under the dwrep directory and ddlimp under the bin directory as per the internal builds, the structure of the official builds shipped to customers need to be verified.

---

```
ddlimp /u database_user /p database_password /c odbc_datasource /f  
path of ddlsme.ctl file/G SSE_ROLE /R Y /s N /l log file path
```

For example:

```
ddlimp /u ORA04521 /p ORA04521 /c qa4e /f C:\upgrade\ddlsme.ctl /G  
SSE_ROLE /R Y /s N /l C:\upgrade\incaolapcr.log
```

---

**NOTE:** Use the Merant/DataDirect driver while creating the ODBC data source to be used in ddlimp. You can get the Merant/DataDirect driver when you install Siebel Components like Siebel Server or twsiebel. Use the “/w Y” option, if your database is Unicode.

---

---

**NOTE:** When creating Unicode tables, you need to use the parameter /Z for DB2 and MSSQL databases, and /W for Oracle databases.

---

## Data Migration Workflows for Siebel Data Warehouse Data Upgrades

Upgrading Siebel Data Warehouse data requires use of data migration workflows to migrate preconfigured content. The data upgrade process consists of the following Informatica workflows:

- **Upgrade\_DAW.** This workflow contains mappings to move data from the old set of system tables, such as S\_ETL\_RUN\_S, which contains the run information, to the new table, W\_ETL\_DEFN\_RUN, and so on.

- **Upgrade\_R\_Image.** This workflow contains mappings to load the r\_tables with data for the period instead of a monolithic r\_image table consisting of data for all processed rows for all the tables.
- **Upgrade\_Unspecifieds.** This workflow contains mappings for modifying the Unspecified values for List Of Val, Product Attribute, Sub Set Dimension is loaded.
- **Upgrade\_Agree.** This workflow contains mappings to load the Agree mini dimension and update the Agree Dimension, Agree Fact and Agree Item Fact with the mini wid. Also columns SEX\_MF\_CD, SEX\_MF\_CD\_I in W\_PERSON\_D and W\_PERSON\_SCD are updated. In addition column DATASOURCE\_NUM\_ID is set to 1 for the Unspecified rows as it has been modified in Analytics version 7.7.
- **Upgrade\_Asset.** This workflow contains mappings to load the Asset mini dimension and update the Asset Dimension and Asset Fact with the mini wid. Also, some denormalized columns are loaded to the Asset Fact.
- **Upgrade\_Opty.** This workflow contains mappings to load the Opty mini dimension and update the Opty Dimension, Revenue Fact, Opty Contact Fact and Pipeline Fact with the mini wid. Also, some denormalized columns are loaded to the Revenue Fact, Opty Contact Fact, and Opty Stage Fact, and Pipeline Fact.
- **Upgrade\_Order.** This workflow contains mappings to load the Order Mini Dimension and update the Order Dimension and Order Item Fact with the mini wid. Also some denormalized columns are loaded to the Order Fact. In addition, the Order Item Fact is now split into Order Fact containing only the header rows, and Order Item fact containing only the line item details the header rows that previously existed are deleted.
- **Upgrade\_Quote.** This workflow contains mappings to load the Quote Mini Dimension and update the Quote Dimension and Quote Item Fact with the mini wid. Also, some denormalized columns are loaded to the Quote Fact. In addition, the Quote Item Fact is now split into Quote Fact containing only the header rows and Quote Item fact containing only the line item details the header rows that previously existed are deleted.
- **Upgrade\_Response.** This workflow contains mappings to load the Response mini dimension and update the Response Dimension and Response Fact with the mini wid. Also some denormalized columns are loaded to the Response Fact.

- **Upgrade\_ServiceRequest.** This workflow contains mappings to load the ServiceRequest mini dimension and update the ServiceRequest Dimension and ServiceRequest Fact with the mini wid. Also, some denormalized columns are loaded to the ServiceRequest Fact.
- **Upgrade\_Others.** This workflow contains mappings to load Region to Geo Dimension and load Geo Wid to Person Dimension, Person SCD, and Org Dimension. Also, some denormalized columns are loaded into Survey and Activity Fact. In addition, columns VENDOR\_NAME and PR\_PROD\_LN in W\_PRODUCT\_D and W\_PRODUCT\_SCD are modified.
- **Upgrade\_Visibility.** This workflow contains mappings to update table\_wid, which have changed in Analytics version 7.7, and the previous source for the table\_wid S\_ETL\_SRC\_TABLE, which is deprecated in this release. The visibility tables that are modified in this workflow are W\_ACT\_PARTY, W\_CON\_PARTY, W\_HHOLD\_PARTY, W\_OPTY\_PARTY, W\_ORDER\_PARTY, W\_ORG\_PARTY, W\_PROG\_PARTY, W\_QUOTE\_PARY, W\_RESP\_PARTY, W\_SEG\_PARTY, and W\_SR\_PARTY.
- **UpgradeSlowlyChangingDimensionStartDates.** This workflow contains mappings to update SCD\_START\_DT\_WID, which has changed in Analytics version 7.7. The Slowly Changing Dimensions tables that are modified in this workflow are W\_PERSON\_SCD, W\_HHOLD\_SCD, W\_PL\_ITEM\_SCD, and W\_PRODUCT\_SCD.

## Running the Data Migration Workflow for Preconfigured Content in Siebel Data Warehouse

Run the workflows under the Upgrade Folder. The following is a list of the upgrades to be run and the order in which to run them:

- Upgrade\_DAW
- Upgrade\_R\_Image
- Upgrade\_Unspecifieds (This workflow should be run before running any of the subsequent workflows.)
- Upgrade\_Agree

- Upgrade\_Asset
- Upgrade\_Opty
- Upgrade\_Order
- Upgrade\_Quote
- Upgrade\_Response
- Upgrade\_ServiceRequest
- Upgrade\_Others
- Upgrade\_Visibility
- UpgradeSlowlyChangingDimensionStartDates

**To run the data migration workflow for preconfigured content**

- Right-click on the desired workflow, and click on Start Workflow to execute it.

## Siebel Data Warehouse Upgrade Customizations

This section describes how to upgrade your customizations. Customization upgrade consists of upgrading Informatica mappings customizations and registering them in DAC.

### Upgrading Informatica Repository Customizations

To upgrade Informatica repository customizations, use the following procedure.

**To copy your custom folder from your existing Informatica repository to the newly restored 7.7 Informatica repository**

- 1** Launch Informatica Repository Manager and connect to both your current and newly restored 7.7 Informatica Repositories (restored in the Informatica repository upgrade, see [“Siebel Data Warehouse Environment Upgrade” on page 212](#)).
- 2** Select the custom folder in your current repository and copy it (Edit > Copy).



- 3 Select the newly restored 7.7 Informatica Repository and paste it (Edit > Paste). (Rename the folder if necessary to paste it in this location.)

This action will create the custom folder and all its contents (mappings, workflows and so on).

- 4 Create an individual workflow for each of the mappings in the custom folder.

These mappings, once registered, would be used by the DAC, and the workflow previously batched would not be used by the DAC.

## Modifying Data Extracts

In Analytics version 7.7, the changed data is loaded into image table for incremental loads but not for full loads. Therefore, the source qualifier for extracts (that is, all mappings whose name starts with SDE\_), uses views that are built differently for incremental and full load by DAC. Use the procedures in this section to access the views.

If your mapping is using a new source table (that is, if this source table does not exist in the DAC repository) see [“Registering Customization in the DAC” on page 219](#) for the procedure to register a new source table and create the corresponding new image tables. (In Analytics version 7.7, every source table which is incrementally loaded will have a corresponding three image tables.)

### **To upgrade customization in the Custom Folder**

- 1 Replace the source primary table (primary table is the one which joins with the image table) to use the corresponding view.

You can get the view name by replacing the prefix of the source table name S\_ with V\_.

- 2 Remove the old image table (S\_ETL\_I\_IAMGE) and S\_ETL\_SRC\_TABLE from the “from” list.
- 3 Remove the join between the source table, old image table (S\_ETL\_I\_IMG) and S\_ETL\_SRC\_TABLE.

### **Example of a 7.5.3 Override**

The following example shows the previous and current version of the SQL override of the Source Qualifier for the Wave Dimension. The changes are highlighted.

```
SELECT
S_DD_CAMP_WAVE.ROW_ID,
S_DD_CAMP_WAVE.SRC_ID, S_DD_CAMP_WAVE.SEQ_NUM,
S_DD_CAMP_WAVE.ALLOC_PCT, S_DD_CAMP_WAVE.DT_LAG
FROM
S_DD_CAMP_WAVE,
S_ETL_I_IMAGE IMG,
S_ETL_SRC_TABLE ISRC
WHERE
((S_DD_CAMP_WAVE.ROW_ID = IMG.ROW_ID
AND
IMG.TABLE_WID = ISRC.ROW_WID
AND
ISRC.TABLE_NAME = 'S_DD_CAMP_WAVE'
))
```

#### 7.7 SQL Override

```
SELECT $$Hint1
S_DD_CAMP_WAVE.ROW_ID,
S_DD_CAMP_WAVE.SRC_ID, S_DD_CAMP_WAVE.SEQ_NUM,
S_DD_CAMP_WAVE.ALLOC_PCT, S_DD_CAMP_WAVE.DT_LAG
FROM
V_DD_CAMP_WAVE S_DD_CAMP_WAVE
$$Hint2
```

If you have customized W\_ORDERITEM\_F or W\_QUOTEITEM\_F you need to redo them according to the new way (Both these facts are now split For Example: W\_ORDERITEM\_F is split into W\_ORDER\_F containing only header rows and W\_ORDERITEM\_F containing only line item rows). See the new *Siebel Data Warehouse Installation and Administration Guide* for more details on these facts.

If your customized mappings are using W\_MAP\_DIM tables modify them to get value from the base tables as map tables are no longer used. See the modifications in the new mappings for W\_PERSON\_D and W\_REVN\_F to compare the changes due to the removal of W\_MAP\_DIM

If you have customized mappings in the Main Folder, use the following procedure.

**To upgrade customization in the Main Folder**

- Manually apply the changes to the corresponding mappings (find equivalent mappings using the mapping name) in the newly restored 7.7 Informatica repository and follow the steps in the previous procedure for upgrading customization in the Custom Folder.

**Registering Customization in the DAC**

New Source Table: If your mapping is using a new source table (that is, if this source table does not exist in the DAC repository), then you need to do the following.

**To register customization in the DAC for a new source table**

- 1 Register the tables in the DAC Repository (Design - Tables - Create New Record for the corresponding table), and assign new suffix to it if this is a Siebel source table for which you want to do incremental ETL. (Use a suffix > 500 and < 999).
- 2 Create the image tables for those tables by right-clicking on the entry and choosing 'Generate Change Capture Scripts' and then manually creating the tables by applying the scripts generated on the OLTP database.

If your mapping is using a new target table (that is, if this target table does not exist in the DAC repository) use the following procedure.

**To register customization in DAC for a new target table**

- 1 Register the tables in the DAC Repository (Design - Tables - Create New Record for the corresponding table).

**2** Register the indices in the DAC Repository (Design - Index - Create New Record for the corresponding index).

**3** Add the table under the group (Design - Table Groups - Create New Record for the corresponding table group).

Typically there is one table group per star. Also, when multiple tables need to be built together, you can create an independent group as well. If there is no group that you can associate with the table, then create one.

**4** Under the desired Subject Area (Design - Subject Area), add the table group, if any were created.

If none of the subject areas satisfies your needs, then create a new one.

**5** Under the desired Execution Plan (Design - Execution Plan), add any newly created Subject areas, if any.

You can run Execution Plans either manually or create schedules to run them periodically.

### **To register customization in the DAC for the Custom Folder**

**1** Import the Informatica Sessions by choosing DAC Repository Management > Import Informatica Sessions.

This imports all the workflow names and folders into DAC repository.

**2** If your Informatica Repository is on a database other than the Siebel Data Warehouse database, define the database connection on your DAC repository by choosing Edit Setup > DAC System Properties and then editing Preferences Main Informatica Repository to reflect the name of the connection that you created.

If you have created a new mapping in the Custom Folder use the following procedure.

---

**NOTE:** If you are using a new source or target table, complete the procedures for registering a new source or target table, before proceeding further.

---

### **To register customization in the DAC for a new mapping in the Custom Folder**

**1** Create separate workflows to populate each table.

- 2 Go to Tools > Import Informatica Workflows in DAC to bring in the newly created workflows in DAC's temporary table.
- 3 Register those as Tasks in DAC Repository (Design - Task Definitions).
  - a Choose the appropriate workflow name for full and incremental command names.
  - b Choose Execution Type to be Informatica.
- 4 Define the execution order by specifying the dependencies.

For example, if the task needs to be run after certain other tasks, then add these tasks to Depends On (Design > Task Definition > Depends On Tasks).
- 5 Specify the source and target tables.

For the procedures using source tables that require change capture, check the Build Image property of the Task Definition, and choose those tables as Primary or Auxilliary.

---

**NOTE:** Use the change capture procedure only for Primary or Auxilliary tables.

---

If you have copied over an existing mapping to the Custom Folder and modified it use the following procedure.

**To register a customized mapping inside the Custom Folder**

- 1 Modify the existing task to point to the Custom Folder.
- 2 Modify the execution order if necessary.

## Data Migration Workflows for Siebel Industry Applications Data Warehouse Upgrades

Siebel Industry Applications upgrade workflows include the following functionality:

**SIA\_Upgrade\_R\_Image.** This workflow contains mappings to load the r\_tables with data for the period instead of a monolithic r\_image table consisting data for all processed rows for all the tables.

**SIA Upgrade Unspecifieds.** This workflow contains mappings for modifying the Unspecified values of DATASOURCE\_NUM\_ID for each Dimension is loaded. It also modifies the Unspecified values for List Of Value Dimension, Activity Product Fact, Fund Fact, Medical Education Fact, Promotion Fact, Revenue Fact, Syndicated Data Fact, Syndicated Market Fact, and Vehicle Fact.

**LS Upgrade ActivityProduct.** This workflow contains mappings to update the Activity Product Fact with the PRD\_CALL\_IND & VIS\_PR\_POS\_ID.

**LS Upgrade Others.** This workflow contains mappings to load Geo Wid to Position Contact Dimension. It also contains the mapping to update the Medical Education, Objective, Profile Rank, Syndicated Data, and Syndicated Market Fact with the VIS\_PR\_POS\_ID.

**SIA Upgrade Household.** This workflow contains mapping to update INCOME\_RANGE and INCOME\_RANGE\_I in W\_HOUSEHOLD\_D because the OLTP source is changed.

**FINS Upgrade Visibility.** This workflow contains mappings to update table\_wid, which has changed in Analytics 7.7, and the previous source for the table\_wid S\_ETL\_SRC\_TABLE, which is deprecated in this release. The visibility tables that are modified in this workflow are W\_ASSET\_PARTY, W\_CLAIM\_PARTY, and W\_HLDNG\_PARTY.

**SIA Upgrade Slowly Changing Dimension Start Dates.** This workflow contains mappings to update SCD\_START\_DT\_WID, which has changed in this release. The Slowly Changing Dimensions tables that are modified in this workflow are W\_ASSET\_SCD, and W\_INS\_CLAIM\_SCD & W\_OPTY\_SCD.

## Running the Data Migration Workflow for Preconfigured Siebel Industry Applications Content

Run the workflows under the Upgrade\_7.5.XtoINCA\_1 folder.

The upgrades to be run and the order in which to run them is as follows:

- Upgrade\_DAW
- Upgrade\_R\_Image

- Upgrade\_Unspecifieds (This workflow should be run before running any of the other subsequent workflows.)
- Upgrade\_Agree
- Upgrade\_Asset
- Upgrade\_Opty
- Upgrade\_Order
- Upgrade\_Quote
- Upgrade\_Response
- Upgrade\_ServiceRequest
- Upgrade\_Others
- Upgrade\_Visibility
- UpgradeSlowlyChangingDimensionStartDates

For more detailed information on running data migration workflows, see [“Running the Data Migration Workflow for Preconfigured Content in Siebel Data Warehouse” on page 215](#).

After the preceding workflows are executed successfully, run the workflows under Upgrade\_7.5.XtoINCA\_2 folder.

The upgrades to be run and the order in which to run them is as follows:

- SIA\_Upgrade\_R\_Image
- SIA\_Upgrade\_Unspecifieds (This workflow should be run before running any of the subsequent workflows.)
- LS\_Upgrade\_ActivityProduct
- LS\_Upgrade\_Others
- SIA\_Upgrade\_Household
- FINS\_Upgrade\_Visibility
- SIA\_UpgradeSlowlyChangingDimensionStartDates





Look up any problems, error messages, and other issues in the following topics.

- [“Siebel Data Warehouse Installation and Initial Configuration Issues” on page 225](#)
- [“Informatica and Loading Issues” on page 226](#)
- [“Unicode Issues with Informatica and the Siebel Data Warehouse” on page 230](#)
- [“UNIX Installation Issues with the Siebel Data Warehouse” on page 232](#)
- [“Siebel Data Warehouse Application Console \(DAC\) Issues” on page 235](#)

## Siebel Data Warehouse Installation and Initial Configuration Issues

**Problem.** Cannot connect to the Siebel Data Warehouse running on DB2.

**Possible cause.** Make sure that the DB2 configuration matches the requirements listed in [“IBM DB2 UDB Windows 2000-Specific Database Requirements for Siebel Data Warehouse” on page 29](#).

**Problem.** After installing Informatica’s ODBC driver (V3.5) on a Windows computer, you can not open the Query Analyzer and the Enterprise Manager shows an error message.

**Probable cause.** The ODBC driver installation replaces the odbcbcp.dll file with an older version. The Informatica documentation contains the procedure for correcting this problem.

**Problem.** Cannot connect to Siebel transactional database from Informatica.

**Possible cause.** Make sure that the server running the Informatica Server software has an ODBC connection to the Siebel transactional database using a Siebel ODBC driver, and an ODBC connection to the Siebel Data Warehouse using the Merant Closed 3.5 32-bit driver.

**Problem.** Error: ORA-12541: TNS:no listener.

**Possible cause.** Check the Compatibility tab in Informatica service configuration. It should be Oracle 8. If it is not set to Oracle 8, change it and restart the Informatica server.

**Problem.** Error 2140 (Informatica service failed to start).

**Possible causes.** The server may not be started or the Informatica Service may not be started. See Informatica's installation and configuration guide on *Siebel eBusiness Third-Party Bookshelf* for detailed information.

**Problem.** Informatica installation fails with an Unhandled Exception error and displays a message similar to this: "Error Number: 0x80040707. Description: Dll function call crashed: ISRT.\_DoInstall."

**Possible causes.** Most likely the computer is running out of virtual memory. Restart the computer and reinstall Informatica.

## Informatica and Loading Issues

Double-click the workflow to view the log file details.

**Problem.** Double-clicking the workflow yields a Workflow Manager error message: "The system cannot find the file specified."

**Probable Cause.** The session log files are not set up properly. You also may need to change the text editor.

**Problem.** Using Oracle, some mappings hang when running when performance statistics are switched on.

**Probable Cause.** When running some Informatica mappings for loading the Siebel Data Warehouse, turning on the Performance Statistics can cause the mapping to hang. The only workaround is to increase the values of the `LMSharedMemory` and `MaxSessions` variables in Informatica. The risk of increasing the `LMSharedMemory` too much is that it may start to have a serious effect on overall performance of the machine that the Informatica server is running on.

**Problem.** When you execute a workflow on the Informatica Workflow Manager, you get the following error message:

“Request to start workflow (workflow name) on server (server name) not successful.”

**Probable cause.** This can happen due to a server time-out property that is usually set to 20 or 40 seconds. When you try to run a large workflow, every session in that workflow is fetched into the server's memory. If this takes longer than the server time-out property, the server returns a message that the workflow was unable to run. However, the workflow is running, but the server just needs time to complete fetching the sessions into memory. Double-click the workflow to view the log file details.

**Problem.** When running Full\_Extract\_Siebel\_DW or Refresh\_Extract\_Siebel\_DW, Informatica returns errors similar to:

```
TE_7007 Transformation Evaluation Error; current row skipped...
TE_7007 [<<Transformation Error>> [to_date]: Date function error
to_date('19010101', 'DD-MON-YYYY')
```

**Probable cause.** Incorrect date entry in the SME Date Format field in the System Preferences. The format is YYYYMMDD.

**Problem.** When running Full\_Load\_Siebel\_DW, Informatica returns errors similar to:

```
CMN_1014 Error creating semaphore...
TM_6006 Error initializing DTM for session...
TM_6006 [s_CR18a1. Load W_PROG_DM_TMP - Program Records]
```

**Probable cause.** Insufficient semaphores allocated. Allocate more semaphores on the Data Warehouse Server. The change becomes effective when you reboot.

**Problem.** Informatica (RDBMS is DB2) gives this error message:

```
Error occurred unlocking [SDE_ServiceRequestDimension1].
An error occurred while accessing the repository[[IBM][CLI
Driver][DB2/6000] SQL0955C
Sort memory cannot be allocated to process the statement. Reason
code = ".
SQLSTATE=57011]
```

```
DB2 Fatal Error[FnName: ExecuteDirect -- SQLSTATE=57011  
[IBM][CLI Driver][DB2/6000]
```

**Probable cause.** The DB2 parameter “SHEAPTHRES” is too small.

**Problem.** Informatica produces the error “Unable to connect to the server” when running a full load of the Siebel Data Warehouse (Full\_Load\_Siebel\_DW\_Dimensions).

**Probable cause.** The last Designer session was not validated. Part of the development process of working with Designer is to always validate any changes to Informatica mappings definitions and sessions after the change is saved in repository.

**Problem.** When loading the data warehouse, Informatica reports a lock problem.

**Probable cause.** Either someone has a session open or there is a dead session. Make sure no one has any open sessions. If no sessions are open, then follow the Informatica documentation on removing locks caused by dead sessions.

**Problem.** After changing an Informatica mapping, you may get an error message when trying to execute “Full\_Load\_Siebel\_DW\_Facts.” The error message is “Unable to connect to the server.”

**Probable Cause.** This is due to Informatica mapping objects that have been modified and this does not automatically validate the session objects. You must validate all changes to any existing mappings in the Informatica repository.

**Problem.** Session SDE\_RecordLoadStart fails due to unique constraint error while executing Full\_Load\_Siebel\_DW\_Dimensions or Full\_Load\_Siebel\_DW\_Facts.

**Probable Cause.** This could be because the previous full load did not complete successfully. Fix the problem that caused the previous load session to fail. Make sure you start the process from last entry of Load\_RestartNextWorkflow # before the failed session, and restart the workflow process from that point.

If you have to reextract the data from the Siebel transactional database because something had to be fixed in source database to resolve the load error then you must restart the ETL process. Truncate S\_ETL\_INC\_STAT in the Siebel transactional database, then enable the Extract and Load workflows and rerun them.

**Problem.** Session SDEINC\_RecordExtractStart fails due to unique constraint error while executing a Refresh workflow.

**Probable Cause.** This could be because the previous load or refresh did not complete successfully. Fix the problem that caused the previous refresh session to fail. Make sure you start the process from last entry of %RestartNextWorkflow # before the failed session, and restart the workflow process from that point.

**Problem.** The session fails and you receive the following error code:

```
Error "TE_7042 Aggregate Error: File Operation Error
```

**Probable Cause.** This is due to a disk space limitation. Check the /Informatica/PowerMart/Cache/Check directory for available disk space, also check the limits (ulimit) of the account used to start PowerMart.

**Problem.** Informatica sessions get deadlocked and eventually fail when they try to do a “select” from the repository table OPB\_OBJECT\_LOCKS. This problem sometimes occurs on MSSQL server databases.

**Probable Cause.** This is possibly caused by a limited number of resources on the MSSQL Database Server. The workaround is to execute the following MSSQL specific SQL command on the Siebel Data Warehouse:

```
DROP INDEX OPB_OBJECT_LOCKS.OPB_OBJ_LOCKS_IDX
DROP INDEX OPB_OBJECT_LOCKS.OPB_OBJ_LOCKS_IDX2
DROP INDEX OPB_OBJECT_LOCKS.OPB_OBJ_LOCKS_IDX3
```

Upon completion of executing these commands, proceed executing the workflow processes to load the Siebel Data Warehouse.

**Problem.** An error may occur when trying to send a post session email notification using MS Outlook 2000. Refer to Informatica release notes for further information.

**Probable Cause.** After installing Informatica Server on Windows, copy the file mapi32.dll from winnt\system32 to the bin folder where the Informatica Server is installed, overwriting the existing mapi32.dll in that directory. Start the Informatica Server so that the Informatica Server can use the new mapi32.dll.

The Extended MAPI Error. MAPILogonEx failed[2147746065] error indicates that the logon is not configured correctly. Check the following:

- 1 Under Services > Informatica > Logon, make sure the login (domain\username) and password are correct.

- 2** Under Control Panel > Mail (it may also be called Mail and Fax or Exchange) > Services > Show Profiles, make sure the mail profile is correct.
- 3** Under Programs > Informatica Server > Informatica Server Setup > Miscellaneous, make sure the MS Exchange profile is correct.

**Problem.** While creating a custom session, bulk load mode does not work properly with SQL Server.

**Probable Cause.** Change the mode to “normal” in Informatica repository for the session. The “normal” mode must be used everywhere for SQL Server in all of your custom sessions.

**Problem.** When running IMR, you may receive an error message box which states “The evaluation period for this Oracle ODBC driver has expired. Please call Merant to obtain a production version of this Oracle ODBC driver.”

**Probable Cause.** This is caused by a down-level ODBC driver license key. Rename or move ivodbc.lic, lvodbc.lic (if it exists), and lvdw.lic (if it exists). Make sure you have only one license file named ivdw.lic in winnt\system32. This eliminates the problem.

**Problem.** Outlook closes when sending out a notification of finishing the ETL process.

**Probable Cause.** Informatica is closing Outlook. This issue is known to Informatica and is scheduled to be resolved in an upcoming release. Until then, create a second profile in Outlook and add that profile name to the Informatica server setup.

## Unicode Issues with Informatica and the Siebel Data Warehouse

**Problem.** Task fails and generates the error message: "TRANSF\_1\_1\_1\_1 > TE\_7073 Aggregate Error: Expecting keys to be ascending." Tasks known to fail are SDE\_DtlForecastFact and SDE\_CostList

**Workaround.** In Informatica Mapping Designer, open the corresponding mapping and in the Aggregator transformation, remove the check from the Sortedinput check box.

**Problem.** In the Windows 2000 environment, with a double-byte language, such as Japanese, when you try to input the path for the directory where Informatica is installed (such as c:\Program Files\Informatica\Informatica PowerMart 6.2.1 Siebel XXX(JPN char) OEM - Server) for the value \$PMRootDir in the Workflow Manager, the double-byte characters are converted to question marks. This causes the ETL to fail because it cannot find the Informatica PowerMart 6.2.1 Siebel OEM - Server folder and its subdirectories.

**Workaround.** Share the folder under a new name with no double-byte characters, and use the network share as the value for \$PMRootDir.

**Problem.** When Configuring the Informatica Server or Informatica Repository Server, there are major truncations in some of the tabs.

For example, in the Server tab and the Repository tab in the Configure Informatica Service screen (Start > Programs > Informatica Server > Informatica Server Setup > Configure Informatica Service), the field names are truncated.

**Workaround.** Informatica 6.x does not support non-English messages and resources. Delete the following files to display all messages in English.

PC/PM client folder\\*411.dll

PMServer folder\bin\\*411.dll

PMRepServer folder\bin\\*411.dll

**Problem.** Unable to start the Informatica repository with the Informatica Repository Server on UNIX.

The Siebel\_DW\_Rep (MS Windows Japanese and superset of Shift-JIS) is not compatible with that of pmrepagent (UTF-8 encoding of Unicode).

**Solution.** Make sure Japanese locales are installed on the UNIX machine. The environment variables on the UNIX machine are not complete. Set the environment variable PM\_CODEPAGENAME to MS932. For other languages, set the variable appropriately.

**Problem.** The database connection code page is incorrect for Unicode/Non-Latin code pages.

**Workaround.** The database connection code page has to be set manually through the Informatica Workflow Manager.

- 1 Start the Workflow Manager.
- 2 Connect to the Siebel\_DW\_Rep using Administrator as the user.
- 3 Click Menu Connection > Select Relational.
- 4 Select the appropriate OLTP connection (DB2\_OLTP for DB2, ORA\_OLTP for Oracle, MSSQL\_OLTP or ODBC\_OLTP for MSSQL).
- 5 For the code page select the UTF-8 encoding of Unicode or the appropriate code page, and then click OK.
- 6 Select the OLAP connection (DB2\_OLAP for DB2, ORA\_OLAP for Oracle, MSSQL\_OLAP or ODBC\_OLTP for MSSQL).
- 7 For the code page select the UTF-8 encoding of Unicode or the appropriate code page, and then click OK.

**Problem.** Unable to install Informatica PowerMart on non-English UNIX locales (for example, ESN, DEU, FRA, ITA). The following error message appears: "FATAL ERROR during initialization, Invalid language specified, Application terminated."

**Workaround.** This error will occur when the locale environment variables are set inconsistently or not at all. These settings are determined by entering the command `locale` at the UNIX command prompt. Make sure the correct locales are installed on the UNIX machine. Set the environment variable `PM_CODEPAGENAME` correctly (For example set it to MS932 for Japanese).

## UNIX Installation Issues with the Siebel Data Warehouse

**Problem.** When starting Informatica in `pmserver.log`, the following error messages appear:

```
FATAL ERROR: LM_36023 [Tue Mar 11 23:12:41 2003] : (9968|1) The IP address [172.0.129.141] for this machine did not match the IP address [172.20.94] in the repository for server [SIEBEL_DW_SERVER].
```



FATAL ERROR: SF\_34014 [Tue Mar 11 23:13:41 2003] : (9968|1) Server initialization failed.

INFO: SF\_34014 [Tue Mar 11 23:13:41 2003] : (9968|1) Server shut down.

**Solution.** Launch Workflow Manager, then right-click on SIEBEL\_DW\_REP and click More. Make sure the Host Name is the machine name where the Informatica Repository Server resides.

**Problem.** When connecting to the Informatica Repository Server from the Repository Administration Console, the following error messages:

- Oracle databases running on Solaris or HP:

Database Connection Error: Database driver event...Error occurred loading library [Bad magic number for shared library: /export/home/oracle/9.0.1.3/lib/libcintsh.sl]Database driver event...Error occurred loading library [libpmora8.sl]

**Solution.** Use Oracle 32-bit library.

**a** Set the environment variable as follows: LD\_LIBRARY\_PATH = /export/home/oracle/9.0.1.3/lib32

**b** Restart the Informatica Repository Server service.

- DB2 databases running on AIX:

(17764|1) Database Connection Error : Database driver event...Error occurred loading library [No such file or directory]Database driver event...Error occurred loading library [libpmdb2.a]

**Solution.** Set the environment variable as follows:

- For DB2 v8:

```
setenv LIBPATH ${LIBPATH}:/usr/opt/db2_08_01/lib
```

- For DB2 v7

```
setenv LIBPATH ${LIBPATH}:/usr/lpp/db2_07_01/lib
```

Restart the Informatica Repository Server service, after setting the variable.

**Problem.** When restoring the Informatica repository, the following error message appears in the dwdb.log:

ERROR : OBJM\_54543 [Tue Mar 25 17:47:16 2003] : (164|2484) DataBase error:  
ORA-01461: can bind a LONG value only for insert into a LONG column

**Solution.** The database is Unicode and the environment variable is needed.

- For Unicode UTF8, set the environment variable to  
`American_America.UTF8.`
- For Unicode AL32UTF8, set the environment variable to  
`American_America.AL32UTF8.`

**Problem.** When starting pmsserver on UNIX, the following error message appears:

INFO : LM\_36039 [Thu Mar 27 11:40:29 2003] : (6772|1) The maximum number of sessions that can run simultaneously is [10].

FATAL ERROR : CMN\_1011 [Thu Mar 27 11:40:29 2003] : (6772|1) Error allocating system shared memory of [3000000] bytes for [Load Manager Shared Memory].  
Error is [0]: []

FATAL ERROR : SF\_34004 [Thu Mar 27 11:40:29 2003] : (6772|1) Server initialization failed.

INFO : SF\_34014 [Thu Mar 27 11:40:29 2003] : (6772|1) Server shut down.

**Solution.** In the pmsserver.cfg file, change MaxSessions to 15 and change LMSharedMem to 3000000.

---

**NOTE:** For each 10 count increase of MaxSessions, increase the LMSharedMem by 2000000 bytes.

---

**Problem.** A session fails (any session), and the following error message appears:

TM\_6227 Error: codepage incompatible in session [SDE\_RecordExtractStart]. The source database DB2\_OLTP and server SIEBEL\_DW\_SERVER do not have compatible code pages. (One way compatibility is required).

**Solution.** For Informatica servers running on UNIX, in the pmsserver.cfg file, set ValidateDataCodePages to `no`.

For Informatica servers running on Windows, navigate to Informatica Server setup > Configuration, and remove the check from the Validate Data CodePages check box.

**Problem.** When trying to start the Informatica Server, the following error message appears:

```
sdcb50a125{qauser7a}/> pmserver
```

```
exec(): 0509-036 Cannot load program pmserver because of the following errors:
0509-150  Dependent module libpmcef.a could not be loaded.
0509-022  Cannot load module libpmcef.a.
0509-026  System error: A file or directory in the path name does not exist.
```

**Solution.** Set the environment parameters as follows:

```
setenv PATH /export/home/informatica/pm:.${PATH}
setenv LD_LIBRARY_PATH /export/home/informatica/
pm:${LD_LIBRARY_PATH}
```

## Siebel Data Warehouse Application Console (DAC) Issues

### Basic Checks

- Make sure that you have installed JDK 1.4.1 (and not just the JRE).
- Make sure that in the config.bat file for Windows, and the config.sh file for UNIX, JAVA\_HOME is pointing to the JDK (and not the JRE).
- Make sure that the DAC\_HOME variable has been properly set.

**Problem.** Email notification does not work:

**Solution.** To make sure that email notification works with the DAC server, make sure that the following are true:

- To set up the email notification feature, you must bring up the DAC client on the same machine as the DAC server.
- The Email server settings (using the DAC client's server setup) point to a valid SMTP server.
- The SMTP server must be supporting Login Authentication. In Exchange server, this amounts to enabling the Basic Authentication for SMTP module.

**Problem.** The DAC server does not start:

**Solution.** When trying to bring up the DAC server, make sure that the following are taken care of:

- The DAC System properties contain the name of the DAC Server Host machine. This specifies the machine on which the DAC server will be run. Ensure that this entry is correct, and points to where you are bringing up the DAC server. Furthermore, there is an entry for the DAC Server OS, which should specify the OS where the DAC server is running. Ensure that this entry is correct.
- Make sure that the \$DAC\_HOME/lib directory contains the needed JDBC libraries.
- You may want to edit the startserver.bat file temporarily, and run the DAC server in such a way that messages show in the command window. (Use JAVA rather than JAWAW for Windows OS). This will enable you to see a lot of run-time information, as the server boots up.
- In the DAC System properties, there is an entry called Logging Level. You can increase the amount of information logged by decreasing the logging-threshold. In other words, if the value currently is SEVERE, make it INFO. This will show more information. If you can still not find the cause, you may want to change it to FINE or FINEST. These levels are the standard levels of Java's Logging API. (SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST).
  - Be aware that as you lower the logging threshold, a lot of run-time information will start getting logged. While this may be advantageous for troubleshooting, you do not want to leave it like this in production mode. Excessive logging will generate large log files quickly and use excessive disk space. There is also a modest performance hit with excessive logging.
- Ensure that there is no DAC server already running on the machine. Only one DAC server instance can exist per machine (and per DAC repository). Some non-visible forms of the DAC server may be running as a background process.
- Ensure that the DAC repository database instance is actually up and running. The DAC repository database being down or the loss of network connectivity to the database will cause the DAC server not to come up, or to go down if it was running.

- Make sure that the DAC repository database connection pool size is moderate. In other words, do not try to acquire an excessive number of connections when the database administrator may have put restrictions on how many a single application can request.

**Issue.** Starting the DAC client and server for the first time:

After configuring the DAC server by using the DAC client, when you bring up the DAC server, you will notice that the DAC client's icon for the Server status stays red. In other words, it suggests the server is down, even though it is running.

This is expected behavior. The first time the DAC server comes up against a repository, it creates a Unique ID for itself, and the repository. Thereafter, when any client is started against the repository, it knows where to expect the DAC server to be, and furthermore, sends a valid identification token containing the Unique ID.

Therefore, after bringing up the DAC server the first time, you may want to close the existing open clients, and bring them up again. This problem is specific to bringing up the DAC client and server for the first time. Once Unique ID has been created, the behavior of the Server Monitor icon will be as expected.

**Issue.** DAC server on AIX versus non-AIX.

Special care must be taken when bringing up the DAC server on AIX machines. For example, if the repository already contains database connection information entries, and Informatica server entries, these must be modified if you are moving the DAC server to an AIX machine.

In particular, you need to delete all the password fields, and re-enter them. For more information, see [Chapter 2, "Installing and Configuring Siebel Data Warehouse."](#)



# Siebel Data Warehouse for Life Sciences Data Considerations

# A

This section includes the following topics:

- [“Importing Syndicated Data into Siebel Data Warehouse” on page 239](#)
- [“Data Loading Issues with Siebel Data Warehouse” on page 247](#)
- [“Supported Incremental Updates in the Siebel Data Warehouse LS Dimension Tables” on page 249](#)

## Importing Syndicated Data into Siebel Data Warehouse

This section covers how to import syndicated data into the Siebel Life Sciences database.

Pharmaceutical companies purchase weekly and monthly sales and prescription data, known as syndicated data, from third-party vendors such as IMS, NDC, and Cegedim. Syndicated data is read-only data about certain measures that is used for sales force analysis reporting and customer targeting. Once delivered, this syndicated data must be loaded into the Siebel ePharma Analytics database in a timely fashion and made available to users in order for them to make use of sales force analysis reporting.

The options for loading syndicated data into the Siebel Data Warehouse include the following:

- Loading from the Siebel operational application transactional database (OLTP) S\_SYND\_DATA table.

Syndicated data can be loaded from the S\_SYND\_DATA table into the Siebel Data Warehouse. Existing syndicated data from the Siebel application OLTP database, stored in the S\_SYND\_DATA table, is extracted, transformed, and loaded by prebuilt routines and populated into the W\_SYND\_DATA\_F and W\_SYND\_MKT\_F Fact tables in the Siebel Data Warehouse. These tables provide information to the Rx Sales subject areas and prebuilt Rx Sales dashboards. (For more information, see the section on Importing, Extracting, and Routing Syndicated Data in *Siebel Life Sciences Guide*.)

Siebel ePharma Analytics supports multiple data types and period types. Therefore, in order to correctly use the ETL process, S\_SYND\_DATA must be loaded in a special way to make sure support for all data types and correct population of the Siebel Data Warehouse (see [“Siebel Analytics Data Loading Matrix for Syndicated Data”](#) on page 242).



- Loading from a flat file source using syndicated data staging tables.

Syndicated data source files can be loaded directly into Siebel Data Warehouse staging tables. In the staging tables, Siebel foreign keys are matched with external sources keys for joining the external source keys with Siebel Analytics keys.

The Data Loading Matrix (“[Siebel Analytics Data Loading Matrix for Syndicated Data](#)” on page 242) illustrates the correct loading of S\_SYND\_DATA data into the Siebel Data Warehouse. The following data types are supported:

- Monthly Rx data by Contact
- Monthly indirect sales data by Brick
- Monthly indirect sales by Account
- Monthly indirect sales by ZIP Code
- Monthly direct sales by Account
- Monthly direct sales by Brick
- Monthly direct sales by ZIP Code
- Weekly Rx data by Contact
- Monthly plan level Rx Data
- Monthly sales market—incentives by ZIP Code
- Monthly sales market—modified by ZIP Code
- Monthly plan level physician Rx Data
- Monthly Rx data by ZIP Code
- Monthly Rx data by Brick

To make sure the data is correctly loaded in the Siebel Data Warehouse, the Siebel source table, S\_SYND\_DATA needs to be populated correctly, based on [Table 35 on page 242](#) and [Table 36 on page 246](#).

### Siebel Analytics Data Loading Matrix for Syndicated Data

The Data Loading Matrix, shown in [Table 35](#), and the Attribute by Data Source Matrix, shown in [Table 36 on page 246](#), are provided to facilitate loading sales and Rx data. The following list explains the table headings:

- **Screen.** Name of the Siebel Analytics screen.
- **View.** Name of the Siebel Analytics view.
- **Data Source.** Value that determines which view displays the data stored in the record. The valid values are RXPrf, SlsIndBrk, SlsIndAct, SlsIndZip, SlsDirAct, SlsDirBrk, SlsDirZip, RXEVM, RXPT, RXSMI, RXSMM, RXXPT, RXZip, and RXBrk.
- **Mandatory Id.** Value that indicates the ID field that must be non-NULL for a database record to be displayed in the corresponding view. There are four key ID fields: Account Id, Contact Id, Territory Id, and Zip Id. For each view listed in the table, one of these ID fields is required and the other three must be NULL.
- **Plan Type.** Value that indicates the type of plan ID a database record must have to be displayed in the view.

**Table 35. Siebel Analytics Data Loading Matrix**

Screen	View	Dashboard	Page	Source OLAP table	OLTP Data Source from S_SYND_DATA to populate Source Star	S_SYND_DATA Mandatory ID	Plan Type
Rx Sales Analytics	Sales Analytics	Pharma Sales	Market Share	W_SYND_DATA_F W_SYND_MKT_F	RxPrf	Contact ID	TOTAL
Rx Sales Analytics	Sales Analytics	Pharma Sales	Product Growth	W_SYND_DATA_F W_SYND_MKT_F	RxPrf	Contact ID	TOTAL
Rx Sales Analytics	Sales Analytics	Pharma Sales	Sales Force Index	W_SYND_DATA_F W_SYND_MKT_F	RxPrf	Contact ID	TOTAL
Rx Sales Analytics	Sales Analytics	Pharma Sales	Market Share (Brick)	W_SYND_DATA_F W_SYND_MKT_F	SlsIndBrk	Area ID	TOTAL

**Table 35. Siebel Analytics Data Loading Matrix**

<b>Screen</b>	<b>View</b>	<b>Dashboard</b>	<b>Page</b>	<b>Source OLAP table</b>	<b>OLTP Data Source from S_SYND_DATA to populate Source Star</b>	<b>S_SYND_DATA Mandatory ID</b>	<b>Plan Type</b>
Rx Sales Analytics	Sales Analytics	Pharma Sales	Product Growth (Brick)	W_SYND_DATA_F W_SYND_MKT_F	SlsIndBrk	Area ID	TOTAL
Rx Sales Analytics	Sales Analytics	Pharma Sales	Sales Force Index (Brick)	W_SYND_DATA_F W_SYND_MKT_F	SlsIndBrk	Area ID	TOTAL
Rx Sales Analytics	Sales Objectives Analytics	Pharma Objectives	Contact Frequency & Reach	W_OBJECTIVE_F	RxPrf (NRx, TRx Objectives)SlsIndZip (Indirect Sales Objectives)	Contact IDZip ID	TOTAL
Rx Sales Analytics	Sales Objectives Analytics	Pharma Objectives	Account Frequency & Reach	W_OBJECTIVE_F	RxPrf (NRx, TRx Objectives)SlsIndZip (Indirect Sales Objectives)	Contact IDZip ID	TOTAL
Rx Sales Analytics	Sales Objectives Analytics	Pharma Objectives	Contact Frequency & Reach (Brick)	W_OBJECTIVE_F	SlsIndBrk	Area ID	TOTAL
Rx Sales Analytics	Sales Objectives Analytics	Pharma Objectives	Account Frequency & Reach (Brick)	W_OBJECTIVE_F	SlsIndBrk	Area ID	TOTAL
Call Activity Analytics	Contact Call Analytics	Pharma Contact Call Effectiveness	Contact Effort vs. Potential	W_ACT_PROD_F W_SYND_DATA_F W_SYND_MKT_F	RxPrf	Contact ID	TOTAL
Call Activity Analytics	Contact Call Analytics	Pharma Contact Call Effectiveness	Contact Priority & Indication	W_ACT_PROD_F	Not Applicable	Not Applicable	Not Applicable

## Siebel Data Warehouse for Life Sciences Data Considerations

Importing Syndicated Data into Siebel Data Warehouse

**Table 35. Siebel Analytics Data Loading Matrix**

Screen	View	Dashboard	Page	Source OLAP table	OLTP Data Source from S_SYND_DATA to populate Source Star	S_SYND_DATA Mandatory ID	Plan Type
Call Activity Analytics	Contact Call Analytics	Pharma Contact Call Effectiveness	Contact Effort vs. Potential (Brick)	W_ACT_PROD_F W_SYND_DATA_F W_SYND_MKT_F	SlsIndBrk	Area ID	TOTAL
Call Activity Analytics	Contact Call Analytics	Pharma Contact Call Effectiveness	Contact Priority & Indication (Brick)	W_ACT_PROD_F	Not Applicable	Not Applicable	Not Applicable
Call Activity Analytics	Account Call Analytics	Pharma Account Call Effectiveness	Account Effort vs. Potential	W_ACT_PROD_F W_SYND_DATA_F W_SYND_MKT_F	SlsIndZip	Zip ID or Account ID	TOTAL
Call Activity Analytics	Account Call Analytics	Pharma Account Call Effectiveness	Account Priority & Indication	W_ACT_PROD_F	Not Applicable	Not Applicable	Not Applicable
Call Activity Analytics	Account Call Analytics	Pharma Account Call Effectiveness	Account Effort vs. Potential (Brick)	W_ACT_PROD_F W_SYND_DATA_F W_SYND_MKT_F	SlsIndBrk	Area ID	TOTAL
Call Activity Analytics	Account Call Analytics	Pharma Account Call Effectiveness	Account Priority & Indication (Brick)	W_ACT_PROD_F	Not Applicable	Not Applicable	Not Applicable
Pharma ROI Analytics	Call Activity Profit & Loss Analytics	Pharma Call Activity Profit & Loss	Call Activity Profit & Loss	W_ACT_PROD_F W_SYND_DATA_F W_SYND_MKT_F	RxPrf	Contact ID	TOTAL

**Table 35. Siebel Analytics Data Loading Matrix**

Screen	View	Dashboard	Page	Source OLAP table	OLTP Data Source from S_SYND_DATA to populate Source Star	S_SYND_DATA Mandatory ID	Plan Type
Pharma ROI Analytics	Call Activity Profit & Loss Analytics	Pharma Call Activity Profit & Loss (Brick)	Call Activity Profit & Loss (Brick)	W_ACT_PROD_F W_SYND_DATA_F W_SYND_MKT_F	SlsIndBrk	Area ID	TOTAL
Medical Education Analytics	MedEd Analytics	Pharma Medical Education	Event Analysis	W_MED_ED_F	Not Applicable	Not Applicable	Not Applicable
Medical Education Analytics	MedEd Analytics	Pharma Medical Education	Speaker Analysis	W_MED_ED_F	Not Applicable	Not Applicable	Not Applicable
Medical Education Analytics	MedEd Analytics	Pharma Medical Education	Budget Analysis	W_MED_ED_F	Not Applicable	Not Applicable	Not Applicable
Medical Education Analytics	MedEd Analytics	Pharma Medical Education	Event ROI	W_MED_ED_FW_SYND_DATA_F	RxPrf	Contact ID	TOTAL
Sales Rep Analytics	Sales Rep Analytics	Sales Rep Analytics	Activity	W_ACT_PROD_F W_SYND_DATA_F W_SYND_MKT_F	RxPrf	Contact ID	TOTAL
Sales Rep Analytics	Sales Rep Analytics	Sales Rep Analytics	Sales	W_SYND_DATA_F W_SYND_MKT_F	RxPrf	Contact ID	TOTAL
Sales Rep Analytics	Sales Rep Analytics	Sales Rep Analytics	Customer	W_SYND_DATA_F W_SYND_MKT_F	RxPrf	Contact ID	TOTAL
Sales Rep Analytics	Sales Rep Analytics	Sales Rep Analytics	Objective	W_OBJECTIVE_F	RxPrf (NRx, TRx Objectives) SlsIndZip (Indirect Sales Objectives)	Contact ID Zip ID	TOTAL

## Siebel Data Warehouse for Life Sciences Data Considerations

Importing Syndicated Data into Siebel Data Warehouse

**Table 35. Siebel Analytics Data Loading Matrix**

Screen	View	Dashboard	Page	Source OLAP table	OLTP Data Source from S_SYND_DATA to populate Source Star	S_SYND_DATA Mandatory ID	Plan Type
Pharma Executive Analytics	Sales Executive Analytics	Pharma Sales Executive Analytics	Sales Overview Page	W_ACT_PROD_F W_SYND_DATA_F W_SYND_MKT_F W_OBJECTIVE_F W_MED_ED_F	RxPrfSlsIndZip	Contact IDZip ID	
Pharma Executive Analytics	Sales Executive Analytics	Pharma Sales Executive Analytics	Sales Overview Page (Brick)	W_ACT_PROD_F W_SYND_DATA_F W_SYND_MKT_F W_OBJECTIVE_F W_MED_ED_F	RxPrfSlsIndBrk	Contact IDArea ID	

## Data Source Values for Syndicated Data Attributes

Table 36 indicates, by data source, the data values that should be loaded into the syndicated data table (S\_SYND\_DATA) attribute fields.

**NOTE:** Analytics Codes are used in the Life Sciences logical model to identify the syndicated data sources.

**Table 36. S\_SYND\_DATA Attribute by Data Source**

Data Source	Description	Analytics Codes	Attribute 1	Attribute 2	Attribute 3	Attribute 4
RXPrf	Prescription data by Contact	1	NRx	TRx	Market NRx	Market TRx
SlsIndBrk	Indirect Sales Brick Level	22	Product Sales \$	Product Sales Units	Market Sales \$	Market Sales Units
SlsIndAct	Indirect Sales Account Level	21	Product Sales \$	Product Sales Units	Market Sales \$	Market Sales Units
SlsIndZip	Indirect Sales Zip Level	23	Product Sales \$	Product Sales Units	Market Sales \$	Market Sales Units
SlsDirAct	Direct Sales Account Level	11	Product Sales \$	Product Sales Units		

**Table 36. S\_SYND\_DATA Attribute by Data Source**

Data Source	Description	Analytics Codes	Attribute 1	Attribute 2	Attribute 3	Attribute 4
SlsDirBrk	Direct Sales Brick Level	12	Product Sales \$	Product Sales Units		
SlsDirZip	Direct Sales Zip Level	13	Product Sales \$	Product Sales Units		
RXEVM	Weekly RX Data	0	Product NRx	Product TRx	Market NRx	Market TRx
RXPT	Plan Level Rx Data	6	Product NRx	Product TRx	Market NRx	Market TRx
RXSMI	Sales Market - Incentives	3	Product NRx	Product TRx	Market NRx	Market TRx
RXSMM	Sales Market - Modified	4	Product NRx	Product TRx	Market NRx	Market TRx
RXXPT	Plan Level Physician Rx Data	5	Product NRx	Product TRx	Market NRx	Market TRx
RXZip	Prescription data by ZIP Code	7	Product NRx	Product TRx	Market NRx	Market TRx
RXBrk	Prescription data by Brick	8	Product NRx	Product TRx	Market NRx	Market TRx

The following issues are specific to Analytics for Life Sciences and do not affect other products.

- [“Data Loading Issues with Siebel Data Warehouse”](#)
- [“Supported Incremental Updates in the Siebel Data Warehouse LS Dimension Tables” on page 249](#)

## Data Loading Issues with Siebel Data Warehouse

Siebel Data Warehouse for Life Sciences updates the Siebel Data Warehouse with a full refresh or using incremental updates. In Informatica, the Full\_Load\_Siebel\_DW workflow does a complete refresh and is usually only used for the initial build. Refresh\_Siebel\_DW incrementally updates the Siebel Data Warehouse.

Running Full\_Load\_Siebel\_DW deletes all existing information stored in the fact and dimension tables. The following types of incremental updates rules are supported on the fact tables.

- **Activity Product Fact.** Add new records (call activity) to the fact table.
- **Rx Sales (syndicated data) Fact.** ETL does not support incremental updates. When running Full\_Load\_Siebel\_DW, all records in the fact and dimension tables are deleted. To maintain a history in the dimension tables (such as multiple alignments), use Refresh\_Siebel\_DW. If you need to incrementally update the RxSales fact table for incremental syndicated data loading, use one of the following strategies:
  - **For incremental insert.** Load the data directly into staging and modify the session not to truncate the fact table. Then use the existing ETL to load it to the fact table.
  - **For incremental update.** Create a new mapping that does a lookup against the fact table and run the update.
- **Objective.** Add new records as per the new ePharma Objective. Update existing records with a changed ePharma Objective to the fact table.
- **Profile Rank.** Add new records to the fact table and update existing records in the fact table.
- **Med Ed.** Add new MedEd events to the fact table. A MedEd event is assumed to be locked in the Siebel ePharma MedEd. If the information is changed in an existing MedEd event, the ETL creates a new record in the fact table. If an existing MedEd event is changed, the ETL process does not load the data. The ETL always skips already loaded MedEd events.

### **Known Issues with the RxSales Fact Table**

The following are known issues with creation of aggregate measure columns in the RxSales fact table.

- With large volumes of syndicated data, the creation of aggregate measures in the Siebel Data Warehouse can take four times the amount of time needed to load the fact table. This may be unacceptable under circumstances when there are large amounts of syndicated data.



- All aggregate measures have been recreated at the metadata level except for MAT. MAT can be entered as a formula for ad-hoc analysis in Siebel ePharma Answers using the function `msum(measure, window)`. For more information, see *Siebel Analytics User Guide*.
- The LS syndicated fact table contains following aggregate measure columns to speed enterprise wide reporting: TTD, TTDLY, TTDPT, MAT, YTD. To update these aggregate columns ETL process requires following steps:

#### **Upgrading aggregate columns**

- 1 Join the syndicated data to the day dimension to denormalize by time.
- 2 Create a second set of time denormalized syndicated data.
- 3 Join these two data sets to obtain appropriate measure for TTD, TTDLY, TTDPT, MAT, and YTD aggregation.
- 4 Aggregate the measures.
- 5 Update the syndicated data aggregate columns.

## Supported Incremental Updates in the Siebel Data Warehouse LS Dimension Tables

In the LS dimension tables, the following incremental updates are supported. Some dimensions have more than one set of attributes.

### **W\_ALIGNMT\_DH**

#### **Slowly Changing Dimension Type. 2**

**Attribute.** Link between ZIP, Brick, Account, and Contact Hierarchy

**Description.** Tracks historical alignments.

- If ETL Alignment Version is set to N, ETL makes changes to existing alignment if there is a change on an assignment rule.
- If ETL Alignment Version is set to Y, it creates a new alignment version.

The dimension has these characteristics:

- The first alignment after Full load is 1.
- A new version is created when a new assignment rule (new position or position relationship) or an assignment criteria is modified (change in postal code, brick, contact, or account) if ETL Alignment Version is set to Y.
- Activation date changes have to be entered manually. The ETL looks for the change and creates a new version, even though the rules have not been applied to the Siebel transactional database.
- Assignment criteria:
  - Contact ZIP Code or Account ZIP Code cannot use ZIP Code ranges. Each ZIP Code assigned to a territory needs to be on a separate row, so the same value needs to be entered for both ZIP Code Low and ZIP Code High.  
For example, if ZIP Code 09654 is assigned to a territory, the value for both ZIP Code High and ZIP Code Low should be 09654. Also, the same ZIP Code should not be assigned twice to the same territory and a ZIP Code should be unique to a territory when using assignment criteria for Contact ZIP Code or Account ZIP Code.
  - Contact Brick or Account Brick require unique bricks assigned to a territory.
- Every new version increases the counter by one and is a sequential number. When a dimension table gets too large, your administrator can create an SQL query to delete all attributes for version numbers.
- Effective date of the alignment is assignment activation date.

### **W\_ORG\_D**

#### **Slowly Changing Dimension Type. 3**

**Attributes.** ACCNT\_TYPE\_CD, ACCNT\_TYPE\_CD1, ACCNT\_TYPE\_CD2, ACCNT\_TYPE\_CD3

**Description.** Tracks three past account types. If the account type changes a fourth time, the first change is deleted and only the past three changes are maintained. Effective date is ETL run date to each group of attributes.

**Attributes.** NAME, NAME1, NAME2, NAME3

**Description.** Tracks three past account names. If the account name changes a fourth time, the first change is deleted and only the past three changes are maintained. Effective date is ETL run date.

**Attributes.** NUMB\_OF\_BEDS, NUMB\_OF\_BEDS1, NUMB\_OF\_BEDS2, NUMB\_OF\_BEDS3

**Description.** For an account type of hospital and clinic, tracks three past quantities. If the number of beds changes a fourth time, the first change is deleted and only the past three changes are maintained. Effective date is ETL run date.

**Attributes.** PAR\_INTEGRATION\_ID, PAR\_ITGR\_ID1, PAR\_ITGR\_ID2, PAR\_ITGR\_ID3

**Description.** Tracks three past parent accounts. If the parent account changes a fourth time, the first change is deleted and only the past three changes are maintained. Effective date is ETL run date.

### **W\_POSTN\_CON\_D**

#### **Slowly Changing Dimension Type. 3**

**Attributes.** STATUS, STATUS1, STATUS2, STATUS3

**Description.** Tracks three past contact statuses. If the contact status changes a fourth time, the first change is deleted and only the past three changes are maintained. Effective date is ETL run date.

### **W\_POSITION\_D**

**Slowly Changing Dimension Type.** 3

**Attributes.** EMP\_FST\_NAME, EMP\_MID\_NAME, EMP\_LAST\_NAME, EMP\_FST\_NAME\_H1, EMP\_MID\_NAME\_H1, EMP\_LAST\_NAME\_H1, EMP\_FST\_NAME\_H2, EMP\_MID\_NAME\_H2, EMP\_LAST\_NAME\_H2, EMP\_FST\_NAME\_H3, EMP\_MID\_NAME\_H3, EMP\_LAST\_NAME\_H3

**Description.** Tracks three past employee names assigned to a position. If the employee name changes a fourth time, the first change is deleted and only the past three changes are maintained. Effective date is ETL run date.

### **W\_PROD\_RANK\_D**

**Slowly Changing Dimension Type.** 3

**Attributes.** RANKING, RANKING1, RANKING2, RANKING3

**Description.** Keep three previous rankings to allow restatement. Effective date is ETL run date.

**Attributes.** RATING, RATING1, RATING2, RATING3

**Description.** Keep three previous ratings to allow restatement. Effective date is ETL run date.

### **W\_PRODUCT\_D**

**Slowly Changing Dimension Type.** 3

**Attributes.** VENDOR\_LOC, VENDOR\_LOC1, VENDOR\_LOC2, VENDOR\_LOC3

**Description.** Tracks three past vendor locations. If the vendor location changes a fourth time, the first change is deleted and only the past three changes are maintained. Effective date is ETL run date.

# Index

## A

ANSI NULL option, setting 36

## B

bulkload mode, error while creating a  
custom session 230

## C

circular hierarchy exceptions, about and  
example 200

Cost List exceptions

*See also* exception reports

about and example 199

Cost lists, using to display 204

products without a cost list 199

currency, diagnosing translation  
issues 204

## D

data flow

Online Analytical Processing (OLAP)

database, about and diagram 16

data import

data loading matrix, table of 242

syndicated data loading, overview 239

data loading

*See also* Data Loading Matrix

Siebel Analytics for Life Sciences, issues

when loading 247

troubleshooting, Informatica and data

loading issues 226

Data Loading Matrix

table of 242

Data Source Matrix, table of 246

database connections, creating

DB2 installations, creating database  
connections for 55

Oracle installation, creating database  
connections for 55

DB Library Options setting, modifying 36

DB2

database connections, creating for 55

Siebel Data Warehouse requirements 29

Siebel Data Warehouse, cannot connect  
to 225

## E

email notification, Outlook error when  
sending post session email 229

Enterprise Manager, error displayed after  
installing ODBC driver 225

errors

*See* troubleshooting

ETL History view, using to view the history  
of the ETL processes and statuses 205

exception reports

*See also* Cost List exceptions; system  
preferences

additional exceptions, list of 205

circular hierarchy exceptions, about and  
example 200

cleansing the data, process and  
diagram 201

ETL History view, using to view the  
history of the ETL processes and  
statuses 205

exception report components, list of 197

exception, fixing 202

exchange rate exceptions, about 199

- Exchange Rate view, using to diagnose currency translation issues 204
- executing the report, cleansing steps 201
- invalid hierarchy exceptions, about 200
- List of Values exceptions 197
- List of Values view, using to compare values with Siebel Data Warehouse 203
- exchange rate exceptions, about 199
- Exchange Rate view, using to diagnose currency issues 204

## F

- Full\_Extract\_Siebel\_DW, error returned by Informatica 227
- Full\_Load\_Siebel\_DW
  - error when running 227
- Full\_Load\_Siebel\_DW\_Facts, error message when executing 228

## H

- hierarchical relationships, invalid hierarchy exceptions, about 200

## I

- Informatica client software
  - Enable Data Code Pages Validation, checking if conditions met 53
  - PowerMart Client, installing on the Administrator Workstation 52
- Informatica Metadata Reporter (IMR)
  - error when running, probable cause 230
- Informatica PowerMart Client, installing on the Administrator Workstation 52
- Informatica Repository Manager
  - configuring, when starting 70
- Informatica Server
  - installing on other than default location 71
- Informatica Server and repository, configuring
  - Informatica repository, configuring 70

- Informatica Server host name, verifying the change 69
- Unicode mode, about running 40
- Unicode mode, configuring under Windows 46
- Windows, setting up Informatica service under 71

- Informatica Server host name
  - verifying that it has been changed 69
- Informatica Service, unable to start on Windows 72
- installation
  - Informatica client software 52
  - Informatica Server, installing on other than default directory 71
  - troubleshooting issues 225

## L

- List of Values
  - exceptions, about and example 197
  - view, using to compare values with the Siebel Data Warehouse 203
- LS dimension tables, supported incremental updates (table) 249

## M

- Mandatory Id fields, description 242
- Microsoft Outlook
  - closes when finishing ETL process, probable cause 230
  - error when sending post session email 229

## O

- OLAP
  - See Online Analytical Processing database
- Online Analytical Processing database
  - about 28
  - Data Warehouse, data flow into 16
- Online Transaction Processing database

- Online Analytical Processing database, differences from 28
- OPB\_OBJECT\_LOCKS Informatica session deadlocked, probable cause 229
- Oracle
  - database connections, creating for 55
  - mappings hang, probable cause 226
  - Siebel Data Warehouse requirements 32
- Outlook
  - closes when finishing ETL process, probable cause 230
  - error when sending post session email 229
- P**
- parent-child relationship, circular hierarchy exceptions 200
- performance
  - components configured to optimize performance 37
- plan types, data loading matrix value 242
- PowerMart Client, installing on the Administrator Workstation 52
- Q**
- Query Analyzer, can not open after installing ODBC driver 225
- R**
- RDBMS error message, probable cause 228
- Refresh\_Extract\_Siebel\_DW
  - error returned by Informatica 227
- repository
  - See* Informatica Repository Manager; Informatics Repository, administering
- S**
- S\_SYND\_DATA attribute by data source matrix, table of 246
- SDE\_RecordLoadStart session fails, probable cause 228
- SDEINC\_RecordExtractStart session fails, probable cause 228
- server components, installation requirements
  - See also* server component software
- Siebel Data Warehouse, about 28
- Windows 2000, additional Data Warehouse server requirements 39
- Siebel Analytics for Life Sciences
  - data loading issues 247
  - LS dimension tables, supported incremental updates (table) 249
- Siebel Data Warehouse
  - DB2 requirements 29
  - Oracle requirements 32
  - SQL server-specific requirements 36
  - troubleshooting configuration issues 225
- Siebel ePharma
  - data loading matrix, table of 242
- Siebel Transaction Database
  - connection error 225
  - Cost lists, displaying 204
- Siebel transactional database
  - Siebel Data Warehouse, differences from 28
- SQL Server
  - DB Library Options setting, modifying 36
  - Siebel Data Warehouse requirements 36
- syndicated data
  - importing, overview 239
  - table (S\_SYND\_DATA), attribute fields 246
- system preferences
  - See also* exception reports
- T**
- tablespace, Oracle-specific requirements 34
- Test entry 99, 225, 247
- troubleshooting
  - Informatica and data loading issues 226

installation and configuration  
issues 225

**U**

Unicode mode  
multibyte or ISO 8859-1, about running  
Informatica Server in 40  
Windows, configuring under 46

**W**

Windows  
Informatica Server, setting up under 71

Unicode mode, configuring under  
Windows 46

Windows 2000

Data Warehouse server requirements,  
additional 39

workflows

error message when executing on the  
Informatica Server Manager, probable  
cause 226

Server Manager error message, probable  
cause 226