



Siebel Database Upgrade Guide

Version 8.0, Rev B

May 2008

ORACLE®

Copyright © 2005, 2008, Oracle. All rights reserved.

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

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

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 Oracle sales representative.

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

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

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

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

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

Contents

Chapter 1: What's New in This Release

Chapter 2: About Siebel Database Upgrade Topics

- How the Siebel Database Upgrade Topics Are Organized 15
- About Applicability of Siebel Database Upgrade Topics 16
- About Terms Used in Siebel Database Upgrade Topics 17
- About File Paths and Commands in Siebel Database Upgrade Topics 18

Chapter 3: Simplified Overview of Siebel Database Upgrade Process

- About Database Environments 21
- Simplified Overview of the Siebel Database Upgrade 21

Chapter 4: How to Perform a Siebel Database Upgrade

- Roadmap for Performing a Siebel Database Upgrade 25
- Process of Planning a Siebel Database Upgrade 27
- Process of Upgrading a Siebel Development Environment 28
- Process of Upgrading a Siebel Production Test Environment 35
- Process of Tuning Siebel Upgrade Performance 40
- Process of Upgrading a Siebel Production Environment 44

Chapter 5: How the Siebel Database Upgrade Works

- Upgrade Flow Orientation 54
- About Supported Siebel Upgrade Paths 54
- About Siebel Upgrade Environments 55
- About the Siebel Database Upgrade Process 58
- About the Siebel Database Configuration Utilities 62
- About the Siebel Upgrade Wizard and Driver Files 68
- About Siebel Additive Schema Changes Mode 70

About the Siebel Database Upgrade Log Files	72
About the Siebel Case Insensitivity Wizard	75
Example of a Siebel Development Upgrade Flow	84
About the Siebel Repository Merge	91
About Inheriting Upgrade Behavior in a Siebel Upgrade	96
About the Siebel Postmerge Utilities	98
About the Siebel Incorporate Custom Layout (ICL) Upgrade Option	102
About the Database Configuration Utilities	112
About Tuning Siebel Production Upgrade Files	112

Chapter 6: Siebel Database and UI Upgrade Planning

Upgrade Flow Orientation	120
About Siebel Upgrade Planning Resources	121
About Best Practices for Doing Your Siebel Database Upgrade	123
About New Siebel Upgrade Features and Requirements	126
About Upgrading Your RDBMS in the Siebel Environment	127
About Siebel User Interface Changes	128
About Database Sort Order in the Siebel Environment	129
About the Siebel Party Model	130
About Migrating Siebel HTML Attachments to Base Tables	133
Upgrade Planning for Multilingual Siebel Deployments	134
Upgrade Planning for Siebel Unicode Support	135
Upgrade Planning for Siebel RC2 or AES Encryption	137
Upgrade Planning for Siebel Web Template Files and Style Sheet	137
Upgrade Planning for Siebel Access Control	139

Chapter 7: Application Planning for a Siebel Upgrade

Upgrade Flow Orientation	142
Determining Your Siebel 7.5.x Release Level	143
Upgrade Planning for Migrating Siebel Address Data	144
Upgrade Planning for Siebel Employee Relationship Management (ERM)	146
Upgrade Planning for Siebel Marketing	146

Upgrade Planning for Siebel Workflow Designer	152
Upgrade Planning for Handheld Devices in the Siebel Environment	156
Upgrade Planning for Resonate Central Dispatch in the Siebel Environment	157
Upgrade Planning for Siebel String Translation	157
Upgrade Planning for Siebel Configurator	158
Upgrade Planning for Siebel Personalization	158
Upgrade Planning for Siebel Pricer and Order Management	159
Upgrade Planning: Additional Siebel Application Changes	159

Chapter 8: Basic Database Preparations for a Siebel Upgrade

Upgrade Flow Orientation	162
Verifying Siebel Database Connectivity	162
Preparing Siebel Tables and Views for Upgrade	163
Preparing Siebel Custom Indexes for Upgrade	164
Exporting Siebel Interface Table Data	164
Archiving Unneeded Siebel Repositories	165
Preserving Siebel Dock Objects and Visibility Rules	165
Securing AIX Memory Allocation Segment Space for the Siebel Database	166
Preparing for a Multilingual Upgrade	166

Chapter 9: Preparing an IBM DB2 Database for a Siebel Upgrade

Verifying the IBM DB2 Client for a Siebel Upgrade	169
Verifying IBM DB2 Sort Order for a Siebel Upgrade	170
Setting IBM DB2 Parameters for a Siebel Upgrade	170
Verifying IBM DB2 Permissions for a Siebel Upgrade	172
Verifying IBM DB2 Instance Owner Permissions for a Siebel Upgrade	173
Creating IBM DB2 Temporary Tablespaces and Bufferpools for a Siebel Upgrade	173
Analyzing IBM DB2 Custom Tablespace Requirements for a Siebel Upgrade	174
Verifying the DB2 Application Development Client for a Siebel Upgrade	176
Identifying IBM DB2 Long Columns for Truncation in a Siebel Upgrade	177

Chapter 10: Preparing an Oracle Database for a Siebel Upgrade

- Verifying Oracle Database Sort Order for a Siebel Upgrade 179
- Setting Oracle Database Configuration Parameters for a Siebel Upgrade 180
- Verifying Oracle Database Parameters for Multiple CPUs in a Siebel Upgrade 181
- Verifying the Oracle Database ODBC Definition for a Siebel Upgrade 182

Chapter 11: Preparing a Microsoft SQL Server Database for a Siebel Upgrade

- Verifying Microsoft SQL Server Sort Order for a Siebel Upgrade 185
- Setting Microsoft SQL Server Temporary Space Size for a Siebel Upgrade 186
- Setting Microsoft SQL Server Configuration Parameters for a Siebel Upgrade 187
- Rebuilding Microsoft SQL Server Clustered Indexes for a Siebel Upgrade 188

Chapter 12: Preparing Siebel Application Data for Upgrade

- Upgrade Flow Orientation 190
- Preparing Siebel Workflow Processes for Upgrade 191
- Preparing Siebel Customized Seed Data for Upgrade 192
- Migrating Siebel Household Data 193
- Setting Up Campaign Status Values for Siebel Marketing 196
- Preserving Siebel Marketing Segment Descriptions 198
- Preparing Siebel Mobile User Data for Upgrade 198
- Setting the Value of Siebel S_SRC_PAYMENT.TYPE_CD for Upgrade 200
- Preparing Siebel Address Data for Upgrade 200
- Migrating Siebel Address Data from Custom Extension Columns 201
- Preparing Siebel Territory Management Rules for Upgrade 208
- Preparing Siebel Customizable Product Data for Upgrade 209

Chapter 13: Upgrading the Siebel Database

- Upgrade Flow Orientation 212
- Renaming the Siebel Repository 213
- Changing the Siebel Database Configuration Utilities Language 214
- Preparing to Run the Siebel Database Configuration Wizard 214

Running the Siebel Database Configuration Wizard Under Windows	218
Running the Siebel Database Configuration Wizard Under UNIX	221
Starting the Siebel Upgrade Wizard	224
Moving the Siebel Repository Files	227
Running the Siebel Case Insensitivity Wizard	228
Applying Siebel Additive Schema Changes	232
Regenerating SQL Files for a Siebel Upgrade	233
Identifying and Dropping Obsolete Indexes for a Siebel Upgrade	234
Preparing for a No-Development-Environment Siebel Upgrade	235
Installing New Siebel License Keys During an Upgrade	236

Chapter 14: Reviewing the Siebel Upgrade Log Files

Upgrade Flow Orientation	238
Summarizing Siebel Log Files Using the Logparse Utility	239
Reviewing Siebel Upgrade Log Files for Errors	241
Manually Archiving Siebel Upgrade Log Files	244

Chapter 15: Performing the Siebel Repository Merge

Upgrade Flow Orientation	248
248	
Preparing for the Repository Merge	249
Configuring Siebel Repository Objects to Inherit Upgrade Behavior	249
Configuring Siebel Tools for the Repository Merge	250
Performing a Siebel Repository Merge	252
Reviewing the Siebel Repository Merge Log Files	255
Generating Siebel EIM Temporary Columns	257
Reviewing Siebel Repository Object Property Conflicts	258
Regenerating the Siebel Repository Definition Files	259
Setting Label Alignment for Siebel Text Fields	262
Copying UI Files to a New Siebel Environment	263
Deleting Unneeded Siebel Repository Files	265
Migrating Siebel Repository Objects to the Standard UI	265

Running the Siebel Postmerge Utilities 270

Chapter 16: Reviewing the Siebel User Interface

Upgrade Flow Orientation 274

Troubleshooting Postmerge Siebel UI Problems 275

Verifying Siebel Business Address Applet Configuration 276

Reviewing Siebel Grid-Based Applets 276

Reviewing Siebel UI Navigation 278

Reviewing Siebel Multi-Value Group (MVG) Shuttle Applets 279

Revising Siebel UI Rich Text Controls 282

Reviewing New Siebel UI Aggregate Categories 283

Revising Siebel Visibility Filters to Display Correctly 284

Assigning a Category and Type to Siebel Chart Views 285

Assigning a Category and Type to Siebel Explorer Views 286

Setting Up Navigation to Inaccessible Siebel Detail Views 287

Eliminating Obsolete Siebel UI Fields 288

Reviewing Siebel UI Objects Affected by Incorporate Custom Layout 289

Reviewing Required Fields in the Siebel UI 290

Assigning an Item Identifier to Siebel Web Template Items 291

Chapter 17: Siebel Postmerge Development Tasks

Upgrade Flow Orientation 294

Reviewing Objects Deleted from the Siebel Repository 295

Reviewing Obsolete Objects in the Siebel Repository 295

Upgrading to the Siebel Symbolic String Model 296

Dropping IBM DB2 8-KB Tablespaces and Buffers After a Siebel Merge 297

Updating Siebel Enterprise Application Integration (EAI) 297

Chapter 18: Postupgrade Tasks for the Siebel Database and File System

Upgrade Flow Orientation 300

Reapplying Schema Customizations in the Siebel Database 301

Checking for Inactivated EIM Table Columns in the Siebel Database 301

Validating Dock Objects and Rule Definitions in the Siebel Database	302
Verifying an Upgraded Oracle RDBMS After a Siebel Upgrade	303
Setting Oracle Database Parameters After a Siebel Upgrade	304
Updating Siebel File System Attachments	305

Chapter 19: Postupgrade Tasks for Siebel Applications

Upgrade Flow Orientation	311
Generating Siebel Reporting Relationships	312
Setting Up Siebel Global Time Zone Support	313
Upgrading Siebel ERM Approval Business Process Workflows	313
Upgrading Siebel ERM Customized Microsite and Group News Pages	314
Migrating Course Duration Information for Siebel Training	316
Upgrading the Launch Field in Siebel Training LOV	316
Upgrading the Test Status in Siebel Training	317
Verifying Class and Session Times in Siebel Training	318
Upgrading Responsibilities in Siebel Marketing	319
Reviewing Siebel Marketing Campaign Data	319
Reviewing Renamed Fields in Siebel Marketing	320
Displaying Regions in Siebel Marketing	321
Revising Program Flowchart Icons in Siebel Marketing	322
Setting Default Campaign Execution Options in Siebel Marketing	322
Upgrading Activity Plans for Programs and Campaigns in Siebel Marketing	323
Upgrading the Newsletter Offer Type in Siebel Marketing	323
Configuring Universal Inbox in Siebel Marketing	324
Configuring Siebel Marketing Purchase Orders for Display	324
Upgrading Siebel Purchase Orders	325
Configuring Siebel Asset-Based Ordering	326
Reviewing Siebel Address Data	328
Upgrading Siebel Attribute Pricing	329
Verifying Aggregate Discounts in Siebel Pricer	329
Upgrading Inbound Siebel Workflows	331
Migrating Data to the Bankruptcy Status Field in Siebel Financial Services	331

Universal Customer Master (UCM) Upgrade Steps After Repository Merge 331
Migrating Clinical Trip Report Data in Siebel Pharma 335

Chapter 20: Tuning the Siebel Upgrade Files

Upgrade Flow Orientation 338
Starting and Stopping Siebel Upgrade Tuner 338
Managing Parallel Threads Using Siebel Upgrade Tuner 340
Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner 343
Transferring UNIX Files for Use by Siebel Upgrade Tuner 346
Rolling Back Siebel Upgrade Tuner Changes 349

Appendix A: Siebel Marketing Upgrade Reference

Obsolete Siebel Marketing Data 353
Obsolete Business Objects in Siebel Marketing 359
Obsolete or Replaced Views in Siebel Marketing 360

Appendix B: Tables Modified or Seeded During a Siebel Upgrade

Important Schema Changes at Siebel 7.8 369

Index

1

What's New in This Release

NOTE: Oracle's Siebel 8.x does not support upgrades from Siebel 6.x, except as noted. You must first upgrade to Siebel 7.7 and then to Siebel 8.x. For information on upgrading to Siebel 7.7, see the latest Siebel 7.7 version of *Siebel Database Upgrade Guide* on Oracle Technology Network.

IBM z/OS upgrades of Siebel 6.2.1 Financial Services directly to Siebel Industry Applications 8.x are supported. You do not have to upgrade to Siebel 7.7 first. To perform this upgrade, see version 8.x of the *Siebel Database Upgrade Guide for DB2 UDB for z/OS*.

What's new in Siebel Database Upgrade Guide, Version 8.0, Rev B

Table 1 lists changes in this version of the documentation to support Siebel 8.0.

Table 1. New Features in Siebel Database Upgrade Guide, Version 8.0, Rev B

Topic	Description
Chapter 3, "Simplified Overview of Siebel Database Upgrade Process"	New chapter. Provides a high-level graphical and informational overview of the Siebel Database upgrade process.
"Upgrade Planning for Multilingual Siebel Deployments" on page 134	New Caution note. Contains information about the lack of support for changing your primary language during upgrade.
"Preparing for the Repository Merge" on page 249	New topic. Provides information on preparing for repository merge, and information on recovering from a failed repository merge.

This guide has been updated to reflect product names changes.

What's New in Siebel Database Upgrade Guide, Version 8.0, Rev A

Table 2 lists changes in this version of the documentation to support Siebel 8.0.

Table 2. New Features in Siebel Database Upgrade Guide, Version 8.0, Rev A

Topic	Description
"Pre-Upgrade Tasks for the Siebel Database" on page 29	Revised topic. Added note indicating that the Database Configuration Wizard will not run unless the Gateway Name Server is running.
"About Upgrading Your RDBMS in the Siebel Environment" on page 127	Revised topic. When using Oracle CBO with a Siebel Database, you must perform certain configuration tasks.
"Upgrade Planning for Siebel String Translation" on page 157	Revised topic. Enhanced discussion of string translation along with a new example.
"Renaming the Siebel Repository" on page 213	Revised topic. Enhanced discussion, including new examples, for conventions when renaming Prior Customer Repository.
"Performing a Siebel Repository Merge" on page 252	Revised topic. Enhanced discussion of ICL option in Merge Options dialog box. To preserve look and feel of pre-7.7 Siebel releases, you must choose Label on Top.
"Performing a Siebel Repository Merge" on page 252	Revised topic. Added note specifying exact location from which to start Siebel Tools to ensure merge0.txt file is correctly generated.

What's New in Siebel Database Upgrade Guide, Version 8.0

Table 3 lists changes in this version of the documentation to support Siebel 8.0.

Table 3. New Features in Siebel Database Upgrade Guide, Version 8.0

Topic	Description
Book Title	The book title is revised from <i>Upgrade Guide</i> to <i>Siebel Database Upgrade Guide</i> . The subject matter of the book and the scope of the book have not changed.
"About Siebel Additive Schema Changes Mode" on page 70	New topic. You can perform part of the production environment upgrade without taking the production database offline. This reduces production upgrade downtime.
"About the Siebel Case Insensitivity Wizard" on page 75	New topic. You can use a Tools wizard to configure repository columns to support case insensitive and accent insensitive queries.
"About the Siebel Incorporate Custom Layout (ICL) Upgrade Option" on page 102	Revised topic. For ICL objects, the ICL Upgrade Path property determines the releases for which Upgrade Behavior = Preserve is effective.

Table 3. New Features in Siebel Database Upgrade Guide, Version 8.0

Topic	Description
Chapter 4, “How to Perform a Siebel Database Upgrade”	Revised chapter. Upgrade process topics are revised to include steps for new upgrade features at Siebel 8.0.
“About New Siebel Upgrade Features and Requirements” on page 126	Revised topic. Lists new upgrade features at Siebel 8.0.
“Upgrade Planning for Multilingual Siebel Deployments” on page 134	Revised topic. The Database Configuration Utilities upgrade the base language plus all deployed languages. Manually upgrading deployed languages is not required.
“Upgrade Planning for Siebel Workflow Designer” on page 152	Revised topic. Customizations to seeded workflows in the Prior Customer Repository are merged into workflows in the New Customer Repository by the repository merge process.
“Preparing for a Multilingual Upgrade” on page 166	New topic. You must review the records in S_LIST_OF_VAL and verify that the deployed languages prior to upgrade.
“Preparing Siebel Territory Management Rules for Upgrade” on page 208	New topic. Territory rules cannot have overlapping effective date ranges.
“Running the Siebel Database Configuration Wizard Under Windows” on page 218	Added task. There are now two tasks within this topic. One task must be run by all users, the other must be run by users with unshipped languages in their system.
“Running the Siebel Database Configuration Wizard Under UNIX” on page 221	Revised topic. The Database Configuration Utilities run in an InstallShield type interface. After the Database Configuration Utilities run, the Siebel Upgrade Wizard must be started manually.
“Running the Siebel Database Configuration Wizard Under UNIX” on page 221	Added task. There are now two tasks within this topic. One task must be run by all users, the other must be run by users with unshipped languages in their system.
“Running the Siebel Case Insensitivity Wizard” on page 228	New topic. The Case Insensitivity wizard configures repository columns to support case insensitive and accent insensitive queries.
“Applying Siebel Additive Schema Changes” on page 232	New topic. You can generate and run an SQL script that performs part of the production environment upgrade without taking the production database offline.
“Reviewing Siebel Upgrade Log Files for Errors” on page 241	Revised topic. For upgrades on IBM DB2 and Oracle, Siebel Logparse no longer displays benign errors. All errors displayed in Siebel Logparse for these upgrades must be resolved.
“Reviewing the Siebel Repository Merge Log Files” on page 255	Revised topic. After a repository merge, you must review both the main repository merge log file and the workflow merge log file.

Table 3. New Features in Siebel Database Upgrade Guide, Version 8.0

Topic	Description
"Updating Siebel File System Attachments" on page 305	Revised topic. Topic rewritten to clarify which file system attachments need to be renamed. Added file system attachments that must be renamed after upgrading to Siebel 8.0.
"Configuring Siebel Marketing Purchase Orders for Display" on page 324	New topic. A new Marketing view associates marketing purchase orders with campaigns. To display marketing purchase orders created in a previous release, you must run a script.
"Universal Customer Master (UCM) Upgrade Steps After Repository Merge" on page 331	New topic. If you are using UCM, you must perform this procedure after the repository merge to maintain your 7.8.2 UCM additions in your 8.0 UCM version.
"Migrating Clinical Trip Report Data in Siebel Pharma" on page 335	New topic. You must verify that the clinical trip reports from the previous releases are upgraded correctly.

2

About Siebel Database Upgrade Topics

This chapter provides information on how to use this guide. This chapter includes the following topics:

- [“How the Siebel Database Upgrade Topics Are Organized”](#) on page 15
- [“About Applicability of Siebel Database Upgrade Topics”](#) on page 16
- [“About Terms Used in Siebel Database Upgrade Topics”](#) on page 17
- [“About File Paths and Commands in Siebel Database Upgrade Topics”](#) on page 18

Related Topics

[Chapter 5, “How the Siebel Database Upgrade Works”](#)

[Chapter 4, “How to Perform a Siebel Database Upgrade”](#)

How the Siebel Database Upgrade Topics Are Organized

Upgrades from: All Supported Siebel releases.

Use the roadmap and process topics in [Chapter 4, “How to Perform a Siebel Database Upgrade”](#) to guide you through the upgrade process. These topics provide a checklist of the steps required to complete an upgrade, in the order you must perform them. Each step includes a link to a topic that explains how to complete the step. The remaining chapters of the *Siebel Database Upgrade Guide* are organized according to the major phases of the upgrade.

NOTE: Topics in the chapters may not follow the order you perform them during the upgrade.

The order of topics in a chapter is as follows:

- A list of all the topics in the chapter.
- **Roadmap topic.** A roadmap is a numbered list of processes. For example, “Roadmap for Performing the Upgrade” lists all the processes required to upgrade your Siebel environments.
- **Process topics.** A process topic has a numbered list of tasks. For example, “Process of Upgrading a Development Environment” lists the tasks required to upgrade a single Siebel environment. These topics begin with “Process of...”.
- **Concept topics.** Concept topics explain key concepts required to perform groups of tasks. To get an overview of how each part of the upgrade works, read the concept topics. These topics usually begin with “About...”. Concept topics do not begin with a gerund, “Process of,” or “Roadmap for.”
- **Task topics.** Task topics explain how to do each step in an upgrade process, and typically include a procedure. Task topics begin with a gerund, for example, “Updating File System Attachments.”

Related Topics

[“About Applicability of Siebel Database Upgrade Topics” on page 16](#)

[“About Terms Used in Siebel Database Upgrade Topics” on page 17](#)

[“About File Paths and Commands in Siebel Database Upgrade Topics” on page 18](#)

About Applicability of Siebel Database Upgrade Topics

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

The applicability of topics in the book is listed at the beginning of each topic. [Table 4](#) lists the applicability categories and their meaning.

For each topic, only the relevant categories are listed.

Table 4. Topic Applicability Categories

Applicability Category	Meaning
Upgrades	<p>Lists the upgrades to which the topic applies, for example:</p> <p>Upgrades from: All Supported Siebel releases. For a list of supported Siebel Releases for your installation, see <i>Siebel System Requirements and Supported Platforms</i> on Oracle Technology Network.</p>
Environments	<p>Lists the Siebel environments to which the topic applies.</p> <p>For example, “Environments: Development environment only” means the topic applies only to a development environment upgrade.</p>

Table 4. Topic Applicability Categories

Applicability Category	Meaning
Databases	<p>Lists the databases to which the topic applies.</p> <p>For example, “Databases: IBM DB2 only” means the topic applies only to IBM DB2 databases.</p>
Platforms	<p>Lists the operating system platform to which the topic applies.</p> <p>For example, “Platforms: Windows only” means the topic applies only to Windows platforms.</p> <p>Windows and UNIX refer to the Enterprise Server platforms (one of which is used by all customers). IBM z/OS refers to the database platform also used by IBM DB2 for z/OS customers.</p>

Related Topics

[“About Siebel Upgrade Environments” on page 55](#)

[“How the Siebel Database Upgrade Topics Are Organized” on page 15](#)

[“About Terms Used in Siebel Database Upgrade Topics” on page 17](#)

[“About File Paths and Commands in Siebel Database Upgrade Topics” on page 18](#)

About Terms Used in Siebel Database Upgrade Topics

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

This guide follows several naming conventions:

- All Supported Siebel releases refer to the upgrade path from which you may upgrade to a certain Siebel Business Applications release. See *Siebel System Requirements and Supported Platforms* on Oracle Technology Network for more information about your upgrade applicability.
- Current release means the currently shipping release of the Siebel Business Applications.
- Siebel CRM 8.x refers collectively to all versions of Siebel CRM 8, for example, Release 8.0. See the Upgrade topic in *Siebel System Requirements and Supported Platforms* on Oracle Technology Network for specific information on which release numbers are meant by Siebel CRM 8.x.
- Siebel 7.x refers collectively to all versions of Siebel 7, for example, Release 7.0.4 or 7.5.3. See the Upgrade topic in *Siebel System Requirements and Supported Platforms* on Oracle Technology Network for specific information on which release numbers are meant by Siebel 7.x.

- Siebel 7.0.x refers to versions of Siebel 7 prior to Release 7.5 that are supported for upgrade to the current release, for example, Release 7.0.4. See the Upgrade topic in *Siebel System Requirements and Supported Platforms* on Oracle Technology Network for specific information on which release numbers are meant by Release 7.0.x.
- Siebel 6.x refers collectively to all versions of Siebel 2000 that are supported for upgrading to the current release. See the Upgrade topic in *Siebel System Requirements and Supported Platforms* on Oracle Technology Network for specific information on which release numbers are meant by Siebel 6.x.
- The term “Windows” refers to all Microsoft Windows operating systems listed as supported for this release in *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.
- The term “UNIX” refers to all forms of the UNIX operating system supported for this release in *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.
- The term *IBM z/OS* refers to the IBM mainframe operating systems collectively referred as z/OS and supported for this release in *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

Related Topics

[“How the Siebel Database Upgrade Topics Are Organized” on page 15](#)

[“About Applicability of Siebel Database Upgrade Topics” on page 16](#)

[“About File Paths and Commands in Siebel Database Upgrade Topics” on page 18](#)

About File Paths and Commands in Siebel Database Upgrade Topics

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Environment variables and path placeholders for both Windows and UNIX paths are used throughout the *Siebel Database Upgrade Guide*. You should enter UNIX commands in a Korn shell. Enter Windows commands in a Windows Command Prompt window.

Windows Paths

The following path conventions specify file system locations in *Siebel Database Upgrade Guide* topics:

- *SIEBEL_ROOT* is the absolute path of the Siebel Server installation directory. When you install a Siebel Server, the installation program queries for the path to the installation directory. The installation program then installs the Siebel Server in a subdirectory of this path called siebsrvr. For example, if you specified C: \sba80 as the installation directory for Siebel CRM 8.0, then *SIEBEL_ROOT* is C: \sba80\siebsrvr.

- *DBSRVR_ROOT* is the absolute path to the Siebel Database Configuration Utilities on the Siebel Server. When you install the Siebel Database Server, the installation script queries for the path to the Siebel Server installation directory. The script then installs the Siebel Database Server files at the same level in a subdirectory called *dbsrvr*. For example, if *SIEBEL_ROOT* is C: \sba8\si ebsrvr, then *DBSRVR_ROOT* is C: \sba80\dbsrvr.
- Throughout this guide, examples use the path C: \ and C: \sba80.

UNIX Paths

The following environment variables and path conventions specify file system locations in *Siebel Database Upgrade Guide* topics:

- *SIEBEL_ROOT* is an environment variable that defines the absolute path of the Siebel Server installation directory. When you install a Siebel Server, the installation script queries for the path to the installation directory. The script then installs the Siebel Server in a subdirectory of this path called *siebsrvr*. For example, if you specified /usr/siebel as the installation directory for Siebel CRM 8.x, then *\$SIEBEL_ROOT* is /usr/siebel/sba80/siebsrvr.
- The definition of *SIEBEL_ROOT* and other environment variables required for doing an upgrade are located in /siebsrvr/siebenv.sh. The Siebel Server installation script sets environment variable definitions in this shell script. Do not edit or delete this file.

TIP: Before performing command line procedures, source *siebenv.csh* first. This refreshes the environment variables required to run commands.

- *DBSRVR_ROOT* is a path convention used in the *Siebel Database Upgrade Guide*. It is not an environment variable and is not defined in *siebenv.csh* or *siebenv.sh*.
DBSRVR_ROOT is the absolute path to the Siebel Database Server files on the Siebel Server. When you install the Siebel Database Server, the installation script queries for the Siebel Server installation directory. The script then installs the Siebel Database Server files at the same level in a subdirectory called *dbsrvr*. For example, if *\$SIEBEL_ROOT* is /usr/siebel/sba8x/siebsrvr, then *DBSRVR_ROOT* is /usr/siebel/sba8x/dbsrvr.
- Throughout this guide, examples use the path /usr/siebel and /usr/siebel/sba8x.
- Run UNIX scripts in a C or Korn shell. Do not run .ksh scripts in a Bourne shell.

Path Navigation

Procedural steps that ask you to navigate to a specified directory should be performed as follows:

- Windows: Open a Command Prompt window and use the *cd* command to make the specified directory the current directory. Do not use the Windows File Explorer to navigate to the directory. For help with the *cd* command, enter the word *help* in the Command Prompt window and click ENTER.
- UNIX: In a shell window, make the specified directory the current directory.

Executing Commands

Procedural steps that ask you to execute a command should be performed as follows, unless specified otherwise:

- Windows: In a Command Prompt window, verify the current directory is correct and enter the command. Do not run the command by entering it in the Run window in the Start Menu.
- UNIX: In a shell window, verify the current directory is correct, source the siebenv script, then enter the command.

Because all versions of the UNIX operating system are case-sensitive, if you are running your Siebel Business Applications under UNIX, treat all file names, directory names, path names, parameters, flags, and command-line commands as lowercase, unless you are instructed otherwise in the product.

If your deployment currently runs under Windows, but you might switch to a UNIX environment or deploy UNIX servers in the future, follow this same practice to avoid having to rename everything later.

Related Topics

[“How the Siebel Database Upgrade Topics Are Organized” on page 15](#)

[“About Applicability of Siebel Database Upgrade Topics” on page 16](#)

[“About Terms Used in Siebel Database Upgrade Topics” on page 17](#)

3

Simplified Overview of Siebel Database Upgrade Process

This chapter provides a simplified overview of the upgrade process of the Siebel Development Database environment. It includes the following topics:

- [“About Database Environments” on page 21](#)
- [“Simplified Overview of the Siebel Database Upgrade” on page 21](#)

About Database Environments

There are potentially three database environments, Development, Production Test, and Production.

The Development environment is where developers use Siebel Tools to customize Siebel applications, and where the customizations are compiled in a Siebel Repository File (SRF). The Production Test environment is where an installed and fully upgraded Development environment is undergoes the quality assurance process. The Production environment is the final step, your live business environment, it includes all of your Siebel Enterprises worldwide.

Simplified Overview of the Siebel Database Upgrade

To upgrade your Siebel database from one release of Siebel Business Applications to another, you must follow the general taskflow outlined in this chapter. This taskflow is, however, general and does not include certain key specifics. For specific tasks associated with each task area, see the chapters and specific topics cross-referenced throughout this chapter.

The examples in this chapter refer to a Development environment. If you have a Development environment, you must upgrade it first, before upgrading other environments. This merges your customizations with the new release. A merged repository file and database schema file are created and become inputs to the production test environment upgrade and production upgrade.

Figure 1 shows a simplified task flow which highlights the general task flow of upgrading your Development Environment. Each chapter in this book which contains tasks related to your upgrade (as opposed to conceptual information) will contain this graphic, as well with the area of interest highlighted to illustrate where you are in the upgrade process.

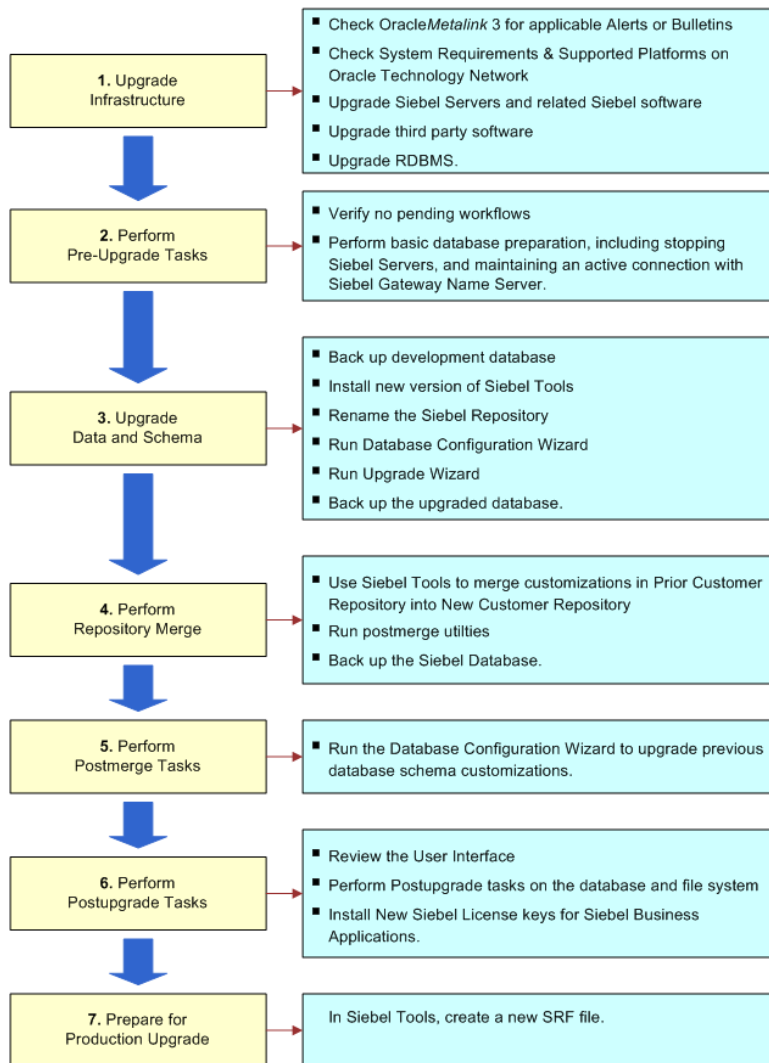


Figure 1. Upgrade Task Flow

Phase 1: Upgrade the Infrastructure

Chapters that include information on Phase 1 include the following:

- **Chapter 2, “About Siebel Database Upgrade Topics.”** Use this chapter for information about the organization of this guide. The chapter also contains information on the applicability of topics to certain environments or upgrade scenarios. Additionally, you can find an overview of terms used throughout the guide and information about file paths and commands used during your Siebel Database upgrade.

- [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#) Use this chapter for an overview of the Siebel Database upgrade process through each environment. Each overview contains hyperlinks to each topic discussed.

When upgrading the infrastructure, first consult Oracle *Metalink* 3 for any Bulletins or Alerts which might be applicable to your installation scenario. Additionally, upgrade all Siebel Server software and other server related software, as well as any third party infrastructure software, and your RDBMS.

Alerts point out key product behavior and important change requests and are often of a time-critical nature. An alert typically includes a description or symptoms of an issue and a recommended action, such as a workaround or application of a maintenance release or fix pack.

Bulletins include detailed, step-by-step information about specific topics related to implementation, configuration and usage of Siebel Business Applications.

For information on Best Practices, see [“Upgrade Planning Best Practices” on page 124](#). For information on specific tasks, see [“Process of Upgrading a Siebel Development Environment” on page 28](#).

Phase 2: Perform Pre-Upgrade Tasks

Chapters that include information on Phase 2 include [Chapter 8, “Basic Database Preparations for a Siebel Upgrade.”](#) Use this chapter for information on preparing your database for your Siebel Upgrade.

Pre-upgrade tasks include verifying Siebel database connectivity, making sure there are no pending workflows, preparing Siebel tables and objects, performing database preparation, including stopping Siebel Servers, and making sure the Siebel Gateway Name Server remains up and running.

Phase 3: Upgrade the Data and Schema

In this phase you will back up your development database, install the new version of Siebel Tools, rename your existing Siebel Repository, and run the Database Configuration Wizard and the Upgrade Wizard.

The Database Configuration Wizard collects information about the upgrade process you want to perform, stores all collected data in a master upgrade file. The SQL generator creates SQL files from these data.

The Upgrade Wizard executes those SQL files which were generated from the master upgrade file as it performs the upgrade.

For more information on Siebel Tools, see *Using Siebel Tools*.

For more information on upgrading the database schema, see [“Upgrade Siebel Database Schema \(upgrep\)” on page 31](#). This topic is part of the process topic, [“Process of Upgrading a Siebel Development Environment” on page 28](#), which walks you through the Upgrade process in greater detail.

Phase 4: Perform the Repository Merge

You prepare for and perform the Repository merge in Siebel Tools. Repository tables store the definitions of the objects used to build your Siebel Application, along with schema definitions. The repository merge process preserves prior customizations by comparing Prior Customer Repository (the repository from your previous release), the New Siebel Repository (the unaltered repository from the current Siebel release), Prior Siebel Repository (unaltered Siebel repository from the previous release) and merges the content into a fourth repository called the New Customer Repository.

For information on all tasks leading up to and including the Repository merge, see [Chapter 15, "Performing the Siebel Repository Merge."](#)

Phase 5: Perform Postmerge Tasks

Following a successful Repository merge, you run the postmerge utilities to make revisions to objects in the New Customer Repository. The utilities are useful for identifying problems with previous UI customizations. For more information on the Postmerge Utilities, see ["Running the Siebel Postmerge Utilities" on page 270.](#)

Additionally, you will review the Siebel user interface to make sure no postmerge issue have arisen. Also, you will review objects that were deleted from the Siebel Repository, review objects made obsolete by the new release, and so on. For more information, see ["Reviewing the Siebel User Interface" on page 273](#) and [Chapter 17, "Siebel Postmerge Development Tasks."](#)

Phase 6: Perform Postupgrade Tasks

At this stage of the upgrade process, you reapply previous schema customizations and perform other verification and validation efforts on your database environment. For more information, see [Chapter 18, "Postupgrade Tasks for the Siebel Database and File System."](#)

Additionally, you will install new license keys for your Siebel Business applications, activate and deploy workflows, and other tasks related to getting your Siebel Business Applications up and running. For more information, see [Chapter 19, "Postupgrade Tasks for Siebel Applications."](#)

Phase 7: Prepare for Production Upgrade

This is the final phase of the development environment upgrade. At this phase, you create a new SRF file, and regenerate the repository definition files if necessary.

For more information creating a new SRF file, see *Developing and Deploying Siebel Business Applications*.

For information on the repository definition files, see ["Regenerating the Siebel Repository Definition Files" on page 259.](#)

4

How to Perform a Siebel Database Upgrade

This chapter provides process information on how to upgrade your Siebel Database. This chapter includes the following topics:

- ["Roadmap for Performing a Siebel Database Upgrade" on page 25](#)
- ["Process of Planning a Siebel Database Upgrade" on page 27](#)
- ["Process of Upgrading a Siebel Development Environment" on page 28](#)
- ["Process of Upgrading a Siebel Production Test Environment" on page 35](#)
- ["Process of Tuning Siebel Upgrade Performance" on page 40](#)
- ["Process of Upgrading a Siebel Production Environment" on page 44](#)

Related Topics

[Chapter 2, "About Siebel Database Upgrade Topics"](#)

[Chapter 3, "Simplified Overview of Siebel Database Upgrade Process"](#)

[Chapter 5, "How the Siebel Database Upgrade Works"](#)

Roadmap for Performing a Siebel Database Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Use one of the roadmaps which follow to guide you through the steps for upgrading your Siebel Database.

Each roadmap consists of a group of processes. Each process consists of a numbered list of tasks. After you complete the tasks in a process, go on to the next process in the roadmap. When you have completed all the processes in the roadmap, the upgrade is complete.

Depending on your database and installed release, you may not need to complete all the tasks in a process. Before starting a task, check the applicability information at the beginning of the task and verify the task applies to your upgrade.

Choose one of the following roadmaps to perform the upgrade:

- ["Roadmap for Upgrading from Siebel 6.x" on page 26](#)
- ["Roadmap for Upgrading from Siebel 7.x on IBM DB2 UDB for z/OS" on page 26](#)
- ["Roadmap for Upgrading from Siebel 7.x If You Have a Development Environment" on page 26](#)
- ["Roadmap for Upgrading from Siebel 7.x Without a Development Environment" on page 26](#)

Roadmap for Upgrading from Siebel 6.x

If you are upgrading from Siebel 6.x, you must first upgrade to Siebel 7.7. Then, you upgrade from Siebel 7.7 to the current release. To upgrade from Siebel 6.x to Siebel 7.7, see the latest Siebel 7.7 version of the *Siebel Database Upgrade Guide*.

Direct upgrades from Siebel 6.2.1 of Siebel Financial Services to the current release of Siebel Industry Applications on IBM z/OS platforms are supported. To perform this upgrade, see *Siebel Database Upgrade Guide for DB2 UDB for z/OS*.

Roadmap for Upgrading from Siebel 7.x on IBM DB2 UDB for z/OS

Upgrading from Siebel 7.x on IBM DB2 UDB for z/OS is described in *Siebel Database Upgrade Guide for DB2 UDB for z/OS*.

The *Siebel Database Upgrade Guide for DB2 UDB for z/OS* contains steps that refer you to the present guide (*Siebel Database Upgrade Guide*) to perform certain tasks. A statement at the beginning of each of these tasks indicates that the task applies to IBM z/OS platforms. For example:

Platforms: Windows, UNIX, IBM z/OS.

Do not perform a task unless IBM z/OS is listed as an applicable platform.

After you have completed a task, refer to the *Siebel Database Upgrade Guide for DB2 UDB for z/OS* to perform the next upgrade step.

Roadmap for Upgrading from Siebel 7.x If You Have a Development Environment

If you are upgrading and have a development environment, complete the processes in this roadmap in the order shown:

- 1 "Process of Planning a Siebel Database Upgrade" on page 27.
- 2 "Process of Upgrading a Siebel Development Environment" on page 28.
- 3 "Process of Upgrading a Siebel Production Test Environment" on page 35.
- 4 "Process of Tuning Siebel Upgrade Performance" on page 40.
- 5 "Process of Upgrading a Siebel Production Environment" on page 44.

Roadmap for Upgrading from Siebel 7.x Without a Development Environment

If you are upgrading and do not have a development environment, complete the processes in this roadmap in the order shown.

Upgrading without a development environment means the following are true:

- You are running an uncustomized, out-of-the box version of Siebel applications.
- You have not used Siebel Tools to create or modify any objects or logical schema definitions in the Siebel Repository.
- You have not modified the physical schema in the Siebel Database.

If your upgrade meets these criteria, complete the following processes in the order shown:

- 1 [“Process of Planning a Siebel Database Upgrade” on page 27.](#)
- 2 [“Preparing for a No-Development-Environment Siebel Upgrade” on page 235.](#)
- 3 Perform a production test environment upgrade. See [“Process of Upgrading a Siebel Production Test Environment” on page 35.](#)
- 4 [“Process of Tuning Siebel Upgrade Performance” on page 40.](#)
- 5 Upgrade your production environment. See [“Process of Upgrading a Siebel Production Environment” on page 44.](#)

Process of Planning a Siebel Database Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows and UNIX only. This topic does not apply to IBM z/OS.

This process is part of a roadmap. See [“Roadmap for Performing a Siebel Database Upgrade” on page 25.](#)

To plan the upgrade, read the following:

- 1 [Chapter 5, “How the Siebel Database Upgrade Works.”](#)
- 2 [Chapter 6, “Siebel Database and UI Upgrade Planning.”](#)
- 3 If you are planning to migrate to Unicode, see [“Upgrade Planning for Siebel Unicode Support” on page 135.](#)
- 4 [Chapter 7, “Application Planning for a Siebel Upgrade.”](#)
- 5 [Appendix A, “Siebel Marketing Upgrade Reference.”](#)
- 6 [Appendix B, “Tables Modified or Seeded During a Siebel Upgrade.”](#)

Remote Upgrade Planning

See the following resources to deploy the upgrade to remote users on Siebel Mobile Web Clients:

- Use Siebel Packager to create language packs for remote installation. See *Going Live with Siebel Business Applications*.
- Use Siebel Anywhere to create installation kits for deployment. See *Siebel Anywhere Administration Guide*.
- Generate a Siebel Remote database template. See *Siebel Remote and Replication Manager Administration Guide*.
- Set up database extraction for Mobile Web Clients. See *Siebel Remote and Replication Manager Administration Guide*.

Process of Upgrading a Siebel Development Environment

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows and UNIX only. This topic does not apply to IBM z/OS.

This process is part of a roadmap. See [“Roadmap for Performing a Siebel Database Upgrade” on page 25](#).

This topic lists the steps required to upgrade a Siebel development environment to the current release. Print this topic and use it as a checklist for doing the upgrade.

The topic is divided into sections, each containing a list of numbered steps. Complete each section in the order shown.

CAUTION: Do not use this topic to perform a development environment upgrade on IBM z/OS. Instead, refer to *Siebel Database Upgrade Guide for DB2 UDB for z/OS*.

Search for Bulletins and Alerts on OracleMetalink 3

- 1 Check OracleMetalink 3 for recently published Bulletins and Alerts regarding the upgrade. For more information, see [“About Siebel Upgrade Planning Resources” on page 121](#).

Upgrade the Servers

Verify you have identified all the maintenance releases, Fix Packs, and quick-fix patches required for the upgrade. These requirements are located in the applicable *Siebel Maintenance Release Guide* on OracleMetalink 3.

CAUTION: Do not install a new Siebel Database as part of upgrading the Siebel Enterprise.

To perform the following steps, see the *Siebel Installation Guide* for the operating system you are using.

- 2 Upgrade the Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE).
For information upgrading these Siebel Enterprise components, see *Siebel Installation Guide* for the operating system you are using.
- 3 Install the Siebel Database Server files on the Siebel Server you will use to perform the upgrade. You only need to install the database server files for the database type that you are upgrading.
- 4 Install language packs for your currently deployed languages and any new languages.
- 5 If you have customized the configuration of Enterprise components, such as Siebel Servers, you can run a script to migrate configuration parameters to the upgraded Siebel Enterprise. See *Going Live with Siebel Business Applications*.

Upgrade Third-Party Software

- 6 Upgrade third-party software as required due to dependencies on Oracle's Siebel software or other installed software. For example, you may need to upgrade the following software:
 - Actuate Server (Siebel Reports Server).
 - Operating system software. Some database upgrades require newer versions of AIX or Windows.

Upgrade the RDBMS

- 7 If required, upgrade the RDBMS version. Refer to the vendor's documentation to perform the upgrade:
 - For information on supported RDBMS systems, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.
 - For information on how upgrading your RDBMS affects the upgrade process, see ["About Upgrading Your RDBMS in the Siebel Environment"](#) on page 127.

Pre-Upgrade Tasks for the Siebel Database

These steps apply to all database types.

- 8 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 9 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 10 Stop Siebel Servers, but make sure the Siebel Gateway Name Server is running.
NOTE: The Database Configuration Wizard will not proceed unless the Gateway Name Server is running.
- 11 Perform the tasks in [Chapter 8, "Basic Database Preparations for a Siebel Upgrade."](#)
- 12 ["Migrating Siebel Repository Objects to the Standard UI"](#) on page 265.

Pre-Upgrade Tasks for IBM DB2

- 13 Perform the tasks in [Chapter 9, "Preparing an IBM DB2 Database for a Siebel Upgrade."](#)
- 14 Apply runstats to your Tools tables, reorganize any that are out of cluster, or that are clustered less than 90%, then execute runstats on those tables again. This will improve performance and reduce the number of tasks that will need to be done after the upgrade is complete.

Pre-Upgrade Tasks for Oracle Database

- 15 Perform the tasks in [Chapter 10, "Preparing an Oracle Database for a Siebel Upgrade."](#)
- 16 Run the Oracle Analyze command on the Siebel Database. Highly fragmented indexes can cause the upgrade to fail.

Pre-Upgrade Tasks for MS SQL Server

- 17 Perform the tasks in [Chapter 11, “Preparing a Microsoft SQL Server Database for a Siebel Upgrade.”](#)
- 18 Run Microsoft SQL Server statistics. This will improve upgrade performance.

Pre-Upgrade Tasks for Application Data

- 19 Perform the tasks in Bulletin 477519.1 (Doc ID) on Oracle*Metalink* 3. This document was previously published as Siebel Technical Note 521. The following table shows the applicability of the tasks in the Bulletin.

Item	Environment
Location and amendment of S_VIEW_WTMPL_IT.ITEM_NUM records	Development, production test, production
Update multi-value group (MVG) and association list applets	Development only
Update unnamed controls in applet web template items	Development only

- 20 Perform the tasks in the Oracle*Metalink* 3 Bulletin listed in the following table.

Title	Environment
477254.1 (Doc ID). This document was previously published as Siebel Technical Note 481.1	Development only

- 21 Perform the tasks in [Chapter 12, “Preparing Siebel Application Data for Upgrade.”](#)

Preparing Developers for the Upgrade

- 22 Back up the development database.
- 23 Verify that all developers have checked in their projects and that all projects are unlocked.
- 24 Verify that all developers have disconnected from the database. The only open account must be the account that is performing the upgrade.
- 25 Install the new Siebel Tools and license keys on development workstations.

NOTE: Keep at least one copy of the previous version of Siebel Tools. You will need it to perform repository operations before the repository merge.

Perform all remaining tasks with the new Siebel Tools, unless stated otherwise.

Copy the r752.dat Ancestor Repository File

26 7.5.2 upgrades only (HOR and SIA):

After you install the Siebel Database Configuration Utilities software, but before you upgrade to the Siebel 8.0 release, you must copy the r752.dat ancestor repository file located in the tools\bin directory to one of the directories listed as follows:

- **Windows:** Copy the r752.dat ancestor repository file from the tools\bin directory to the DBSRVR_ROOT\common directory.
- **UNIX:** FTP the r752.dat ancestor repository file from the tools\bin directory on the Windows machine to the DBSRVR_ROOT/common directory on the UNIX machine.

Upgrade Siebel Database Schema (upgrep)

27 [“Renaming the Siebel Repository” on page 213.](#)

28 (Optional) [“Changing the Siebel Database Configuration Utilities Language” on page 214.](#)

29 Run the Database Configuration Wizard:

- [“Preparing to Run the Siebel Database Configuration Wizard” on page 214.](#)
- [“Running the Siebel Database Configuration Wizard Under Windows” on page 218.](#)
- [“Running the Siebel Database Configuration Wizard Under UNIX” on page 221.](#)

Choose the following settings:

- **Upgrade Options:** Upgrade Siebel Database Schema (upgrep)
- **Environment Type:** Development

30 Edit generated SQL files as required by Bulletins or Alerts on Oracle *Metalink* 3, and by the applicable version of *Siebel Release Notes* on Oracle *MetaLink* 3.

31 Resume the upgrade: [“Starting the Siebel Upgrade Wizard” on page 224.](#)

32 Review the upgrade logs and resolve errors:

- [“Summarizing Siebel Log Files Using the Logparse Utility” on page 239.](#)
- [“Reviewing Siebel Upgrade Log Files for Errors” on page 241.](#)

33 If the upgrade contains unacceptable errors, do the following:

- a Restore the backup of the database.
- b Correct the errors.
- c If errors occurred because you entered incorrect information in the Database Configuration Wizard, see [“Regenerating SQL Files for a Siebel Upgrade” on page 233.](#)
- d Rerun the Database Configuration Wizard.

34 Back up the upgraded database.

Prepare for Repository Merge

- 35 Set the Upgrade Ancestor property for copied objects. See [“Configuring Siebel Repository Objects to Inherit Upgrade Behavior”](#) on page 249.
- 36 See [“Configuring Siebel Tools for the Repository Merge”](#) on page 250.
- 37 (IBM DB2 and Microsoft SQL Only) Run database statistics on the Siebel Database. If upgrading from Siebel 7.0.x or 7.5.x, run statistics specifically on S_SYM_STR and S_SYM_STR_INTL.
Running statistics on the Siebel Database improves merge performance.

Perform Repository Merge

Use the upgraded version of Siebel Tools to perform these tasks.

- 38 [“Performing a Siebel Repository Merge”](#) on page 252.
- 39 [“Reviewing the Siebel Repository Merge Log Files”](#) on page 255.
- 40 If the repository merge contains unacceptable errors, do the following:
 - a Restore the backup of the upgraded database.
 - b Correct the errors.
 - c Rerun the repository merge.
- 41 [“Running the Siebel Postmerge Utilities”](#) on page 270.
- 42 [“Generating Siebel EIM Temporary Columns”](#) on page 257.
- 43 (Optional) Configure case insensitive queries in the New Customer Repository. See [“Running the Siebel Case Insensitivity Wizard”](#) on page 228.
- 44 Reset Siebel Tools configuration. See [“Configuring Siebel Tools for the Repository Merge”](#) on page 250.
- 45 Back up the Siebel Database.

Upgrade Custom Database Schema (upgphys)

- 46 Run the Database Configuration Wizard:
 - [“Preparing to Run the Siebel Database Configuration Wizard”](#) on page 214.
 - [“Running the Siebel Database Configuration Wizard Under Windows”](#) on page 218.
 - [“Running the Siebel Database Configuration Wizard Under UNIX”](#) on page 221.Choose the following settings:
 - **Upgrade Options:** Upgrade Custom Database Schema (upgphys)
 - **Environment Type:** Development
- 47 Run the Upgrade Wizard: [“Starting the Siebel Upgrade Wizard”](#) on page 224.
- 48 Review the upgrade log files and resolve errors:

- [“Summarizing Siebel Log Files Using the Logparse Utility”](#) on page 239.
 - [“Reviewing Siebel Upgrade Log Files for Errors”](#) on page 241.
- 49** If the upgrade contains unacceptable errors, do the following:
- a** Restore the backup of the database you made after the repository merge.
 - b** Correct the errors.
 - c** Repeat the steps beginning after the backup.
- 50** [“Manually Archiving Siebel Upgrade Log Files”](#) on page 244.
- 51** Back up the upgraded database.

Initialize and Extract Developers’ Local Databases

- 52** Install the new release of Siebel Tools on developers’ machines.
- 53** Initialize and extract the local database on development machines. For more information, see *Developing and Deploying Siebel Business Applications*.

Review the User Interface

- 54** [“Reviewing Siebel Repository Object Property Conflicts”](#) on page 258
- 55** Review the following topics on UI upgrade before proceeding:
- [“About Inheriting Upgrade Behavior in a Siebel Upgrade”](#) on page 96
 - [“About the Siebel Postmerge Utilities”](#) on page 98
 - For ICL upgrades: [“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option”](#) on page 102
- 56** If you performed an ICL upgrade and chose Label on Top, revise the cascading style sheet. See [“Setting Label Alignment for Siebel Text Fields”](#) on page 262.
- 57** Create a new SRF file to assist in UI testing. For information on creating an SRF file, see *Developing and Deploying Siebel Business Applications*.
- 58** If you customized style sheet or Web template files in the previous release, reimplement those customizations in the new release as desired.
- Carefully review the UI in the new release before reimplementing customizations to these files.
- 59** Perform the tasks in [Chapter 16, “Reviewing the Siebel User Interface.”](#)
- 60** The postmerge utilities do not convert certain types of flow-based applets to grid-based applets. For example, they do not convert custom form applets to grid-based applets. See the “Editing Applet Layout” portion of *Configuring Siebel Business Applications* and convert remaining flow-based applets as desired.

Postmerge Development Tasks

- 61** Perform the tasks in [Chapter 17, “Siebel Postmerge Development Tasks.”](#)

- 62 Resolve any business component and join conflicts.
- 63 ["Deleting Unneeded Siebel Repository Files"](#) on page 265.

Postupgrade Tasks for Database and File System

- 64 Perform the tasks in [Chapter 18, "Postupgrade Tasks for the Siebel Database and File System."](#)
- 65 Reset upgrade-specific database and database server parameters back to their recommended settings for production. See *Siebel Installation Guide* for the operating system you are using for recommended parameter settings.
- 66 If you exported data from interface tables before the upgrade, review the database and import the data as desired. See *Siebel Enterprise Integration Manager Administration Guide*.
- 67 ["Upgrade Planning for Siebel RC2 or AES Encryption"](#) on page 137.
- 68 Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide*.
- 69 Run database statistics.

NOTE: The upgrade is complete. The remaining topics deal with configuration and validation tasks.

Postupgrade Tasks for Applications Configuration

- 70 ["Installing New Siebel License Keys During an Upgrade"](#) on page 236.
- 71 If applicable, review the results of the Person and Organization merge. Make configuration changes as required. To determine applicability or obtain more information, see ["About the Siebel Party Model"](#) on page 130.
- 72 Perform the tasks in [Chapter 19, "Postupgrade Tasks for Siebel Applications."](#)
- 73 Verify the function of interfaces in integrated applications.
- 74 Activate and deploy workflows.
- 75 If you have set up integrations for transferring data to or from third-party applications using Siebel EAI, verify the integrations are configured correctly. For information on EAI, see *Overview: Siebel Enterprise Application Integration*.
- 76 If you have used EIM to set up batch processing jobs, verify EIM is configured correctly. For information on EIM, see *Siebel Enterprise Integration Manager Administration Guide*.

System Tests

- 77 Use available test data to perform unit testing. Validate the operation of the application in the following areas:
 - User interface
 - Data interfaces
 - Integrity of migrated data

- Workflow function

Prepare for Transition to Production Test Environment

- 78** Create a new SRF file. For information on creating an SRF file, see *Developing and Deploying Siebel Business Applications*.
- 79** If you revised the repository after running upgphys, you must regenerate the repository definition files. See [“Regenerating the Siebel Repository Definition Files” on page 259](#).

Process of Upgrading a Siebel Production Test Environment

Upgrades from: All Supported Siebel releases.

Environments: Production test environment only. Does not apply to production environment.

Platforms: Windows and UNIX only. This topic does not apply to IBM z/OS.

This process is part of a roadmap. See [“Roadmap for Performing a Siebel Database Upgrade” on page 25](#).

This topic lists the steps required to upgrade a production test environment to the current release. Print this topic and use it as a checklist for doing the upgrade.

The topic is divided into sections, each containing a list of numbered steps. Complete each section in the order shown.

CAUTION: Do not use this topic to perform a production test environment upgrade on IBM z/OS. Instead, refer to *Siebel Database Upgrade Guide for DB2 UDB for z/OS*.

Search for Bulletins and Alerts on OracleMetalink 3

- 1** Check OracleMetalink 3 for recently published Bulletins and Alerts regarding the upgrade. For more information, see [“About Siebel Upgrade Planning Resources” on page 121](#).

Upgrade the Servers

Verify you have identified all the maintenance releases, Fix Packs, and quick-fix patches required for the upgrade. These requirements are located in the applicable *Siebel Maintenance Release Guide* on OracleMetalink 3

CAUTION: Do not install a new Siebel Database as part of upgrading the Siebel Enterprise.

To perform the following steps, see the *Siebel Installation Guide* for the operating system you are using.

- 2** Upgrade the Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE). For information upgrading these Siebel Enterprise components, see *Siebel Installation Guide* for the operating system you are using.

- 3 Install the Siebel Database Server files on the Siebel Server you will use to perform the upgrade. You only need to install the database server files for the database type that you are upgrading.
NOTE: You need only to install a new Siebel Server software if you are using a different machine from that on which you performed the Production Test Upgrade. If you are using the same Siebel Server installation as for the Production Test Upgrade, the existing files can be leveraged instead.
- 4 Install language packs for your currently deployed languages and any new languages.
- 5 If you have customized the configuration of Enterprise components, such as Siebel Servers, you can run a script to migrate configuration parameters to the upgraded Siebel Enterprise. See *Going Live with Siebel Business Applications*.

Upgrade Third-Party Software

- 6 Upgrade third-party software as required due to dependencies on Oracle's Siebel software or other installed software. For example, you may need to upgrade the following software:
 - Actuate Server (Siebel Reports Server).
 - Operating system software. Some database upgrades require newer versions of AIX or Windows.

Upgrade the RDBMS

- 7 If required, upgrade the RDBMS version. Refer to the vendor's documentation to perform the upgrade:
 - For information on supported RDBMS systems, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.
 - For information on how upgrading your RDBMS affects the upgrade process, see ["About Upgrading Your RDBMS in the Siebel Environment"](#) on page 127.

Pre-Upgrade Tasks for the Siebel Database

These steps apply to all database types.

- 8 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 9 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 10 Perform the tasks in [Chapter 8, "Basic Database Preparations for a Siebel Upgrade."](#)

Pre-Upgrade Tasks for IBM DB2

- 11 Perform the tasks in [Chapter 9, "Preparing an IBM DB2 Database for a Siebel Upgrade."](#)

- 12 Make sure that runstats is current in your environment and that tables with clustering indexes are not out of cluster. Reorganize any tables that are out of cluster or that are clustered less than 90% in advance of the upgrade if possible.

Table maintenance should be aggressively pursued prior to the upgrade to ensure that your system is in optimal condition. The same general consideration should also be kept in mind with regard to repository tables.

Pre-Upgrade Tasks for Oracle Databases

- 13 Perform the tasks in [Chapter 10, "Preparing an Oracle Database for a Siebel Upgrade."](#)
- 14 Run the Oracle Analyze command on the Siebel Database. Highly fragmented indexes can cause the upgrade to fail.

Pre-Upgrade Tasks for Microsoft SQL Server

- 15 Perform the tasks in [Chapter 11, "Preparing a Microsoft SQL Server Database for a Siebel Upgrade."](#)
- 16 Run Microsoft SQL Server statistics. This will improve upgrade performance.

Pre-Upgrade Tasks for Application Data

- 17 Perform the tasks in Bulletin 477519.1 (Doc ID) on Oracle*Metalink* 3. This document was previously published as Siebel Technical Note 521. The following table shows the applicability of the tasks in the Bulletin.

Item	Environment
Location and amendment of S_VIEW_WTMPL_IT.ITEM_NUM records	Development, production test, production
Update multi-value group (MVG) and association list applets	Development only
Update unnamed controls in applet web template items	Development only

- 18 Perform the tasks in [Chapter 12, "Preparing Siebel Application Data for Upgrade."](#)

Prepare the Siebel Database for Upgrade

- 19 Refer to Bulletin 477772.1 (Doc ID) on Oracle*Metalink* 3. This document was previously published as Siebel Technical Note 586, and includes instructions on how to set up the Siebel Database and Siebel Servers in the production test environment.
- 20 Stop the Siebel Servers and the Siebel Gateway Name Server.

21 Close all database connections. The only database connection must be the account performing the upgrade.

22 Copy application files to the environment:

- Custom SRF file.
- Reports files.
- Custom Web templates and style sheets. See [“Copying UI Files to a New Siebel Environment” on page 263](#).

23 Copy the custrep.dat and schema.ddl files to the environment. See [“Moving the Siebel Repository Files” on page 227](#)

If you have made changes to the Siebel Repository since performing the development upgphys, you must regenerate the schema.ddl and custrep.dat files. See [“Regenerating the Siebel Repository Definition Files” on page 259](#).

24 Verify the production test database is either a copy of the current production database or has the same topology and amount of data.

This is important for effective upgrade tuning before performing the production upgrade.

25 Back up the production test environment database. (If you backed up the database as part of an RDBMS upgrade, ignore this step.)

To do upgrade tuning, you will restore this database and perform test-upgrades on it.

Run the Upgrade Utilities in Prepare for Production Mode

26 (Optional) [“Changing the Siebel Database Configuration Utilities Language” on page 214](#).

27 Run the Database Configuration Wizard:

- [“Preparing to Run the Siebel Database Configuration Wizard” on page 214](#).
- [“Running the Siebel Database Configuration Wizard Under Windows” on page 218](#).
- [“Running the Siebel Database Configuration Wizard Under UNIX” on page 221](#).

Choose the following settings:

- **Upgrade Options:** Prepare for Production Upgrade
- **Environment Type:** Production Test

28 Review the following topic and perform the procedure if applicable: [“Identifying and Dropping Obsolete Indexes for a Siebel Upgrade” on page 234](#).

Upgrade the Siebel Database (upgrep and upgphys)

29 Run the Database Configuration Wizard:

- [“Preparing to Run the Siebel Database Configuration Wizard” on page 214](#).
- [“Running the Siebel Database Configuration Wizard Under Windows” on page 218](#).

- [“Running the Siebel Database Configuration Wizard Under UNIX” on page 221.](#)

Choose the following settings:

- **Upgrade Options:** Upgrade Siebel Database Schema (upgrep + upgphys)
- **Environment Type:** Production Test

- 30 Edit generated SQL files as required by Bulletins or Alerts on *OracleMetalink 3, Siebel Release Notes* on *OracleMetaLink 3* or other publications.
- 31 Resume the upgrade: [“Starting the Siebel Upgrade Wizard” on page 224.](#)
- 32 Review the upgrade logs and resolve errors:
 - [“Summarizing Siebel Log Files Using the Logparse Utility” on page 239.](#)
 - [“Reviewing Siebel Upgrade Log Files for Errors” on page 241.](#)
- 33 If the upgrade contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c If errors occurred because you entered incorrect information in the Database Configuration Wizard, see [“Regenerating SQL Files for a Siebel Upgrade” on page 233.](#)
 - d Rerun the Database Configuration Wizard.
- 34 [“Manually Archiving Siebel Upgrade Log Files” on page 244.](#)
- 35 [“Installing New Siebel License Keys During an Upgrade” on page 236.](#)
- 36 Back up the upgraded database.

Postupgrade Tasks for Database and File System

- 37 Perform the tasks in [Chapter 18, “Postupgrade Tasks for the Siebel Database and File System.”](#)
- 38 Reset upgrade-specific database and database server parameters back to their recommended settings for production. See *Siebel Installation Guide* for the operating system you are using for recommended parameter settings.
- 39 If you exported data from interface tables before the upgrade, review the database and import the data as desired.
- 40 [“Upgrade Planning for Siebel RC2 or AES Encryption” on page 137.](#)
- 41 Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide*.
- 42 Run database statistics.

NOTE: The upgrade is complete. The remaining topics deal with configuration and validation tasks.

Postupgrade Tasks for Applications Configuration

- 43 Perform the tasks in [Chapter 19, “Postupgrade Tasks for Siebel Applications.”](#)

- 44 Verify the function of interfaces in integrated applications.
- 45 Activate and deploy workflows.
- 46 If you have set up integrations for transferring data to or from third-party applications using Siebel EAI, verify the integrations are configured correctly. For information on EAI, see *Overview: Siebel Enterprise Application Integration*.
- 47 If you have used EIM to set up batch processing jobs, verify EIM is configured correctly. For information on EIM, see *Siebel Enterprise Integration Manager Administration Guide*.
- 48 If you customized style sheet or Web template files in the previous release, reimplement those customizations in the new release as desired.
Carefully review the UI in the new release before reimplementing customizations to these files.

System Tests

- 49 Use available test data to perform unit testing. Validate the operation of the application in the following areas:
 - User interface
 - Data interfaces
 - Integrity of migrated data
 - Workflow function

Process of Tuning Siebel Upgrade Performance

Upgrades from: All Supported Siebel releases.

Environments: Production test environment only. Does not apply to production environment.

Platforms: Windows and UNIX only. This topic does not apply to IBM z/OS.

NOTE: This process is optional.

This process is part of a roadmap. See ["Roadmap for Performing a Siebel Database Upgrade" on page 25](#).

Use this process to run test upgrades in the production test environment so you can tune upgrade performance. Improving upgrade performance reduces downtime when you perform the production environment upgrade. The steps in this process cover standard performance tuning. To implement more advanced tuning, including high-availability tuning, contact your Oracle sales representative for Oracle Advanced Customer Services.

Also use this process to test the additive schema changes feature. This feature allows you to perform part of the upgrade on the production database without taking it offline. This reduces the downtime required to upgrade the production database. Test the additive schema changes feature to verify that it does not adversely affect the operation of the application.

Perform this process in the production test environment. Do not perform this process in the production environment.

Review the following upgrade planning and performance tuning resources before performing this process:

- [“About Siebel Additive Schema Changes Mode” on page 70](#)
- **Bulletin 478308.1 (Doc ID) on OracleMetalink** 3. This document was previously published as Siebel Technical Note 616. This Bulletin describes the strategies for minimizing production environment downtime during an upgrade. The steps that follow are intended primarily for use with the Baseline best practices described in Bulletin 478308.1 (Doc ID).
- [Chapter 20, “Tuning the Siebel Upgrade Files.”](#) This chapter provides detailed information on how to use Upgrade Tuner.

UNIX users. You must have a Siebel Server installed on a Windows host to run the Upgrade Tuner. To obtain a Windows version of Siebel Server, contact your account manager or Oracle Global Customer Support.

CAUTION: Do not use this topic to tune upgrade performance on IBM z/OS. Instead, refer to *Siebel Database Upgrade Guide for DB2 UDB for z/OS*.

Set Up the Target Database

- 1 Back up and remove the upgraded production test database.
This preserves seed data and metadata changes you have made that must be migrated to the production environment. This database is called the **production-test final database**.
- 2 In the production test environment, install a recent backup of your production database.
This database has not been upgraded and is called the **target database**. You will use it to perform test upgrades as part of tuning upgrade performance.
- 3 Define an ODBC connection to the target database.
- 4 Verify that the target database and RDBMS server are configured for optimum upgrade performance:
 - [Chapter 9, “Preparing an IBM DB2 Database for a Siebel Upgrade.”](#)
 - [Chapter 10, “Preparing an Oracle Database for a Siebel Upgrade.”](#)
 - [Chapter 11, “Preparing a Microsoft SQL Server Database for a Siebel Upgrade.”](#)
- 5 [“Preparing Siebel Tables and Views for Upgrade” on page 163.](#)
- 6 Run statistics on the target database. This optimizes query performance.
- 7 **UNIX users.** [“Securing AIX Memory Allocation Segment Space for the Siebel Database” on page 166.](#)
- 8 Perform the tasks in [Chapter 12, “Preparing Siebel Application Data for Upgrade.”](#)

Testing Additive Schema Changes

If you do not plan to run additive schema changes on the production database in the production environment, you do not need to perform the steps in this topic.

Perform these steps on the target database.

- 9 Review [“About Siebel Additive Schema Changes Mode”](#) on page 70.
- 10 (Optional) [“Changing the Siebel Database Configuration Utilities Language”](#) on page 214.
- 11 In the production test environment, run the Database Configuration Wizard:
 - [“Preparing to Run the Siebel Database Configuration Wizard”](#) on page 214.
 - [“Running the Siebel Database Configuration Wizard Under Windows”](#) on page 218.
 - [“Running the Siebel Database Configuration Wizard Under UNIX”](#) on page 221.
- 12 Choose Apply Additive Schema Changes mode.

The Apply Additive Schema Changes mode identifies changes that can be made to the target database without taking it offline.

Windows. You will not be prompted whether you want to run the Upgrade Wizard. Instead, the Upgrade Wizard starts automatically, and creates the `schema.additive.sql` script. The Upgrade Wizard does not run the script against the target database.
- 13 **UNIX.** Run the Siebel Upgrade Wizard.

See [“Starting the Siebel Upgrade Wizard”](#) on page 224

Specify `master_additive_gen.ucf` as the input file.

The Upgrade Wizard generates the `schema.additive.sql` script. It does not run the script against the target database.
- 14 Run the SQL script against the target database.

See [“Applying Siebel Additive Schema Changes”](#) on page 232.
- 15 Thoroughly review the operation of all applications. Verify that applying the additive schema changes has not adversely affected the operation of all applications.
- 16 If applying additive schema changes adversely affects the operation of the applications, do the following:
 - a For help diagnosing the cause of the problem and help in taking corrective action create a service request (SR) on Oracle*Metalink* 3. Alternatively, you can phone Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers remain the same and are listed on Oracle*Metalink* 3.
 - b Restore the target database from backup before proceeding with further upgrade tuning.
 - c Do not use the additive schema changes feature during your production upgrade.
- 17 If applications function normally, continue with upgrade tuning. Consider using the additive schema changes feature during your production upgrade.

Upgrading the Target Database (upgrep and upgphys)

18 (Optional) [“Changing the Siebel Database Configuration Utilities Language” on page 214.](#)

19 In the production test environment, run the Database Configuration Wizard:

- [“Preparing to Run the Siebel Database Configuration Wizard” on page 214.](#)
- [“Running the Siebel Database Configuration Wizard Under Windows” on page 218.](#)
- [“Running the Siebel Database Configuration Wizard Under UNIX” on page 221.](#)

Choose the following settings:

- **Upgrade Options:** Upgrade Siebel Database Schema (upgrep + upgphys)
- **Environment Type:** Production

20 In the production test environment, run Siebel Upgrade Wizard.

See [“Starting the Siebel Upgrade Wizard” on page 224.](#)

Note the time required to upgrade the database.

21 Review the upgrade log files for errors:

- [“Summarizing Siebel Log Files Using the Logparse Utility” on page 239.](#)
- [“Reviewing Siebel Upgrade Log Files for Errors” on page 241.](#)

22 If the upgrade contains errors that prevented completion or adversely affected performance, correct the errors and rerun the upgrade.

23 [“Manually Archiving Siebel Upgrade Log Files” on page 244.](#)

Tuning the Upgrade Files

24 Evaluate upgrade performance, particularly the time required to complete the upgrade.

25 Do one of the following:

- If the time required to complete the upgrade is acceptable, no further tuning is needed. Perform the steps in [“Restore the Production-Test Final Database”](#).
- If the time required to complete the upgrade is too long, perform the remaining steps in this topic to continue tuning upgrade performance.
- If the time required to complete the upgrade is too long and you cannot tune further, contact Expert Services to apply advanced tuning.

26 Carefully review target database and RDBMS server configuration. Adjust as needed to further improve upgrade performance.

27 Run Upgrade Tuner to tune the upgrade files:

- **UNIX users.** [“Transferring UNIX Files for Use by Siebel Upgrade Tuner” on page 346.](#)
- [“Starting and Stopping Siebel Upgrade Tuner” on page 338.](#)
- [“Managing Parallel Threads Using Siebel Upgrade Tuner” on page 340.](#)

- [“Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner” on page 343.](#)

To roll back changes you have made in previous Upgrade Tuner sessions, see [“Rolling Back Siebel Upgrade Tuner Changes” on page 349.](#)

Restore the Target Database

Perform these steps if you have made changes to the upgrade environment or to the upgrade files and want to run the upgrade again to verify performance improvement.

- 28** In the production test environment, restore the target database from backup.

This returns the target database to its non-upgraded state so that you can perform another test upgrade.

- 29** In the production test environment, apply additive schema changes again as desired.

- 30** Perform another test upgrade and evaluate upgrade performance.

- 31** Repeat the tuning process and perform test-upgrades until upgrade performance is acceptable.

Restore the Production-Test Final Database

Perform these steps only if you have completed tuning upgrade performance.

- 32** In the production test environment, delete and remove the target database.

- 33** In the production test environment, restore the production-test final database from backup.

Process of Upgrading a Siebel Production Environment

Upgrades from: All Supported Siebel releases.

Environments: Production environment.

Platforms: Windows and UNIX only. This topic does not apply to IBM z/OS.

To perform this process, you must be able to execute ODBC commands on the production database from within the production test environment. For more information, see [“About the Siebel Database Upgrade Process” on page 58.](#) If you cannot do this, contact Technical Support.

This process is part of a roadmap. See [“Roadmap for Performing a Siebel Database Upgrade” on page 25.](#)

This topic lists the tasks required to transition your production test environment to production. Print this topic and use it as a checklist for doing the upgrade. It is divided into sections, each containing numbered steps. Complete the steps in the order shown.

CAUTION: Do not use this topic to perform a production upgrade on IBM z/OS. Instead, see *Siebel Database Upgrade Guide for DB2 UDB for z/OS.*

Search for Bulletins and Alerts on Oracle*Metalink* 3

- 1 Check Oracle*Metalink* 3 for recently published Bulletins and Alerts regarding the upgrade. For more information, see [“About Siebel Upgrade Planning Resources” on page 121](#).

Upgrade the Servers

Verify you have identified all the maintenance releases, Fix Packs, and quick-fix patches required for the upgrade. These requirements are located in the applicable *Siebel Maintenance Release Guide* on Oracle*Metalink* 3.

CAUTION: Do not install a new Siebel Database as part of upgrading the Siebel Enterprise.

To perform the following steps, see the *Siebel Installation Guide* for the operating system you are using.

- 2 Upgrade the Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE).
For information upgrading these Siebel Enterprise components, see *Siebel Installation Guide* for the operating system you are using.
- 3 Install the Siebel Database Server files on the Siebel Server you will use to perform the upgrade. You only need to install the database server files for the database type that you are upgrading.
- 4 Install the language packs for your currently deployed languages and any new languages.
- 5 If you have customized the configuration of Enterprise components, such as Siebel Servers, you can run a script to migrate configuration parameters to the upgraded Siebel Enterprise. See *Going Live with Siebel Business Applications*.

Upgrade Third-Party Software

- 6 Upgrade third-party software as required due to dependencies on Oracle’s Siebel software or other installed software. For example, you may need to upgrade the following software:
 - Actuate Server (Siebel Reports Server).
 - Operating system software. Some database upgrades require newer versions of AIX or Windows.

Apply Additive Schema Changes

These steps are optional. Do not perform these steps unless you have thoroughly tested the additive schema changes feature in the production test environment and have verified that it will not adversely affect your applications.

Perform these steps on the production database. You do not need to take the production database offline. You can perform these steps anytime prior to running the production upprep.

After applying additive schema changes there is no need to run database statistics. The schema changes do not affect the performance of your release prior to the upgrade.

- 7 Review [“About Siebel Additive Schema Changes Mode” on page 70](#).

- 8 Verify you have a recent backup of the production database and that you are logging transactions.

If you encounter problems applying additive schema changes, you might need to restore the production database.

- 9 (Optional) [“Changing the Siebel Database Configuration Utilities Language” on page 214.](#)

- 10 In the production environment, start the Database Configuration Wizard:

- [“Preparing to Run the Siebel Database Configuration Wizard” on page 214.](#)
- [“Running the Siebel Database Configuration Wizard Under Windows” on page 218.](#)
- [“Running the Siebel Database Configuration Wizard Under UNIX” on page 221.](#)

- 11 Choose Apply Additive Schema Changes mode.

The Apply Additive Schema Changes mode identifies changes that can be made to the production database without taking it offline.

Windows. You will not be prompted whether you want to run the Upgrade Wizard. Instead, the Upgrade Wizard starts automatically, and creates the `schema.additive.sql` script. The Upgrade Wizard does not run the script against the production database.

- 12 UNIX. Run the Siebel Upgrade Wizard.

See [“Starting the Siebel Upgrade Wizard” on page 224.](#)

Specify `master_additive_gen.ucf` as the input file.

The Upgrade Wizard generates the `schema.additive.sql` script. It does not run the script against the production database.

- 13 Run the SQL script against the production database.

See [“Applying Siebel Additive Schema Changes” on page 232.](#)

- 14 Thoroughly review the operation of all applications. Verify that applying additive schema changes has not adversely affected applications function.

- 15 If applying additive schema changes adversely affects the operation of the application, do the following:

- a Contact Oracle Global Support for help in diagnosing and correcting the problem.
- b If required, restore the production database from backup.

Upgrade the RDBMS

- 16 If required, upgrade the RDBMS version. Refer to the vendor’s documentation to perform the upgrade:

- For information on supported RDBMS systems, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.
- For information on how upgrading your RDBMS affects the upgrade process, see [“About Upgrading Your RDBMS in the Siebel Environment” on page 127.](#)

Pre-Upgrade Tasks for the Siebel Database

These steps apply to all database types.

- 17 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 18 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 19 Stop the Siebel Servers and the Siebel Gateway Name Server.
- 20 Verify that the Siebel Gateway Name Server is running.
- 21 Stop all other Siebel Servers.
- 22 Close all database connections. The only open connection must be the account performing the upgrade.
- 23 Perform the tasks in [Chapter 8, “Basic Database Preparations for a Siebel Upgrade.”](#)

Pre-Upgrade Tasks for IBM DB2

- 24 Perform the tasks in [Chapter 9, “Preparing an IBM DB2 Database for a Siebel Upgrade.”](#)
- 25 Make sure that runstats is current in your environment and that tables with clustering indexes are not out of cluster. Reorganize any tables that are out of cluster in advance of the upgrade if possible.

Pre-Upgrade Tasks for Oracle Database

- 26 Perform the tasks in [Chapter 10, “Preparing an Oracle Database for a Siebel Upgrade.”](#)
- 27 Run the Oracle Analyze command on the Siebel Database. Highly fragmented indexes can cause the upgrade to fail.

Pre-Upgrade Tasks for Microsoft SQL Server

- 28 Perform the tasks in [Chapter 11, “Preparing a Microsoft SQL Server Database for a Siebel Upgrade.”](#)
- 29 Run Microsoft SQL Server statistics. This will improve upgrade performance.

Pre-Upgrade Tasks for Application Data

30 Perform the tasks in Bulletin 477519.1 (Doc ID) on Oracle *Metalink* 3. This document was previously published as Siebel Technical Note 521. The following table shows the applicability of the tasks in the Bulletin.

Item	Environment
Location and amendment of S_VIEW_WTMPL_IT.ITEM_NUM records	Development, production test, production
Update multi-value group (MVG) and association list applets	Development only
Update unnamed controls in applet web template items	Development only

31 Perform the tasks in [Chapter 12, “Preparing Siebel Application Data for Upgrade.”](#)

Pre-Upgrade Tasks for the UI

32 Copy application files to the environment:

- Custom SRF file.
- Reports files.
- Custom Web templates and style sheets. See [“Copying UI Files to a New Siebel Environment” on page 263.](#)

Upgrade the Siebel Database (upgrep and upgphys)

You do not run the upgrep and upgphys portions of the upgrade in the production environment. Instead, the numbered steps in this topic implement the following process:

- Run the Database Configuration Wizard in the production test environment.
- In the Utilities, enter the information for the production environment instead of the production test environment. For example, you enter the ODBC connection for the production environment.
This information configures the driver file to run against the production database rather than the production test database. It also configures the driver file to use the upgrade SQL files you generated for the production test upgrade.
- Run the Upgrade Wizard. The Upgrade Wizard uses the SQL files in the production test environment to upgrade the database in the production environment.

This approach has several advantages:

- You do not have to generate upgrade SQL files in the production environment, and then manually transfer the customizations to them from the production test environment.

- You do not lose any changes to the SQL files that were made by Siebel Upgrade Tuner in the production test environment.
- You do not have to run the Database Configuration Wizard in Prepare for Production mode again.
- With some exceptions, you do not have to perform again the database-related configuration tasks required by *Siebel Release Notes* on Oracle*MetaLink* 3 or Alerts on Oracle*Metalink* 3.

If your network configuration prevents creating an ODBC connection to your production database from inside your production test environment, contact Oracle Global Customer Support for assistance in completing the production upgrade.

- 33** Verify you have a current backup of the production environment database.
- 34** On the Siebel Server you used to upgrade the production test environment, create an ODBC to connect to the production environment database.
- 35** Navigate to DBSRVR_ROOT\common (UNIX: DBSRVR_ROOT/common) and verify that the file sqlgen.usg exists.

This file contains a record of when the SQL generator was run. When you run the Database Configuration Wizard, if this file exists, no SQL commands are generated.

CAUTION: If this file does not exist, do not run the Database Configuration Wizard. It will overwrite the SQL files used to upgrade your production test database. Contact Oracle Global Customer Support for guidance on proceeding with the upgrade.

You do not have to run the Database Configuration Wizard in Prepare for Production Mode.

- 36** (Optional) [“Changing the Siebel Database Configuration Utilities Language”](#) on page 214.

- 37** Run the Database Configuration Wizard:

- [“Preparing to Run the Siebel Database Configuration Wizard”](#) on page 214.
- [“Running the Siebel Database Configuration Wizard Under Windows”](#) on page 218.
- [“Running the Siebel Database Configuration Wizard Under UNIX”](#) on page 221.

a Choose the following settings when you run the utility:

- Upgrade Options:** Upgrade Siebel Database Schema (upgrep + upgphys)
- Environment Type:** Production

b Enter the information for the production environment instead of the production test environment.

c Enter the name of the ODBC for connecting to the production database.

d When prompted whether you want to run the Siebel Upgrade Wizard, answer No and exit.

This updates the master UCF file with the production environment configuration. When you run the Siebel Upgrade Wizard, the SQL commands will be executed on the production environment database.

- 38** In the production test environment, verify that the SQL scripts for performing the upgrade were not overwritten. You can do this by checking the modification times. If the scripts were overwritten, do not continue. Instead, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

39 Perform the production database upgrade. See [“Starting the Siebel Upgrade Wizard” on page 224](#).

The Siebel Upgrade Wizard uses the SQL commands generated for the production test environment to upgrade the production environment. If you used Upgrade Tuner to revise the SQL commands, these changes are included.

40 Review the upgrade logs and resolve errors:

- [“Summarizing Siebel Log Files Using the Logparse Utility” on page 239](#).
- [“Reviewing Siebel Upgrade Log Files for Errors” on page 241](#).

41 If the upgrade contains unacceptable errors, do the following:

- a** Restore the backup of the database.
- b** Correct the errors.
- c** Rerun the Siebel Upgrade Wizard.

42 [“Manually Archiving Siebel Upgrade Log Files” on page 244](#).

43 [“Installing New Siebel License Keys During an Upgrade” on page 236](#).

44 Use the Siebel Application Deployment Manager to migrate administrative data such as LOVs and responsibilities from production test to production. See *Going Live with Siebel Business Applications*.

45 Back up the upgraded production environment database.

Postupgrade Tasks for Database and File System

46 Perform the tasks in [Chapter 18, “Postupgrade Tasks for the Siebel Database and File System.”](#)

47 Reset upgrade-specific database and database server parameters back to their recommended settings for production. See *Siebel Installation Guide* for the operating system you are using for recommended parameter settings.

48 If you exported data from interface tables before the upgrade, review the database and import the data as desired.

49 [“Upgrade Planning for Siebel RC2 or AES Encryption” on page 137](#).

50 Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide*.

51 Run database statistics.

NOTE: The upgrade is complete. The remaining topics deal with configuration and validation tasks.

Postupgrade Tasks for Applications Configuration

52 If applicable, review the results of the Person and Organization merge. Make configuration changes as required. To determine applicability or obtain more information, see [“About the Siebel Party Model” on page 130](#).

53 Perform the tasks in [Chapter 19, “Postupgrade Tasks for Siebel Applications.”](#)

- 54 Verify the function of interfaces in integrated applications.
- 55 Activate and deploy workflows.
- 56 If you have set up integrations for transferring data to or from third-party applications using Siebel EAI, verify the integrations are configured correctly. For information on EAI, see *Overview: Siebel Enterprise Application Integration*.
- 57 If you have used EIM to set up batch processing jobs, verify EIM is configured correctly. For information on EIM, see *Siebel Enterprise Integration Manager Administration Guide*.
- 58 If you customized the style sheet or Web template files in the previous release, reimplement those customizations in the new release as desired.

System Tests

- 59 Use available test data to perform unit testing. Validate the operation of the application in the following areas:
 - User interface
 - Data interfaces
 - Integrity of migrated data
 - Workflow function

5

How the Siebel Database Upgrade Works

This chapter provides an overview of your Siebel Database upgrade. This chapter includes the following topics:

- ["Upgrade Flow Orientation" on page 54](#)
- ["About Supported Siebel Upgrade Paths" on page 54](#)
- ["About Siebel Upgrade Environments" on page 55](#)
- ["About the Siebel Database Upgrade Process" on page 58](#)
- ["About the Siebel Database Configuration Utilities" on page 62](#)
- ["About the Siebel Upgrade Wizard and Driver Files" on page 68](#)
- ["About Siebel Additive Schema Changes Mode" on page 70](#)
- ["About the Siebel Database Upgrade Log Files" on page 72](#)
- ["About the Siebel Case Insensitivity Wizard" on page 75](#)
- ["Example of a Siebel Development Upgrade Flow" on page 84](#)
- ["About the Siebel Repository Merge" on page 91](#)
- ["About Inheriting Upgrade Behavior in a Siebel Upgrade" on page 96](#)
- ["About the Siebel Postmerge Utilities" on page 98](#)
- ["About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option" on page 102](#)
- ["About the Database Configuration Utilities" on page 112](#)
- ["About Tuning Siebel Production Upgrade Files" on page 112](#)

Related Topics

[Chapter 2, "About Siebel Database Upgrade Topics"](#)

[Chapter 3, "Simplified Overview of Siebel Database Upgrade Process"](#)

[Chapter 4, "How to Perform a Siebel Database Upgrade"](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase One of the Siebel Database upgrade flow, as shown in [Figure 2](#). At this phase of the upgrade you are gathering information and upgrading your infrastructure.

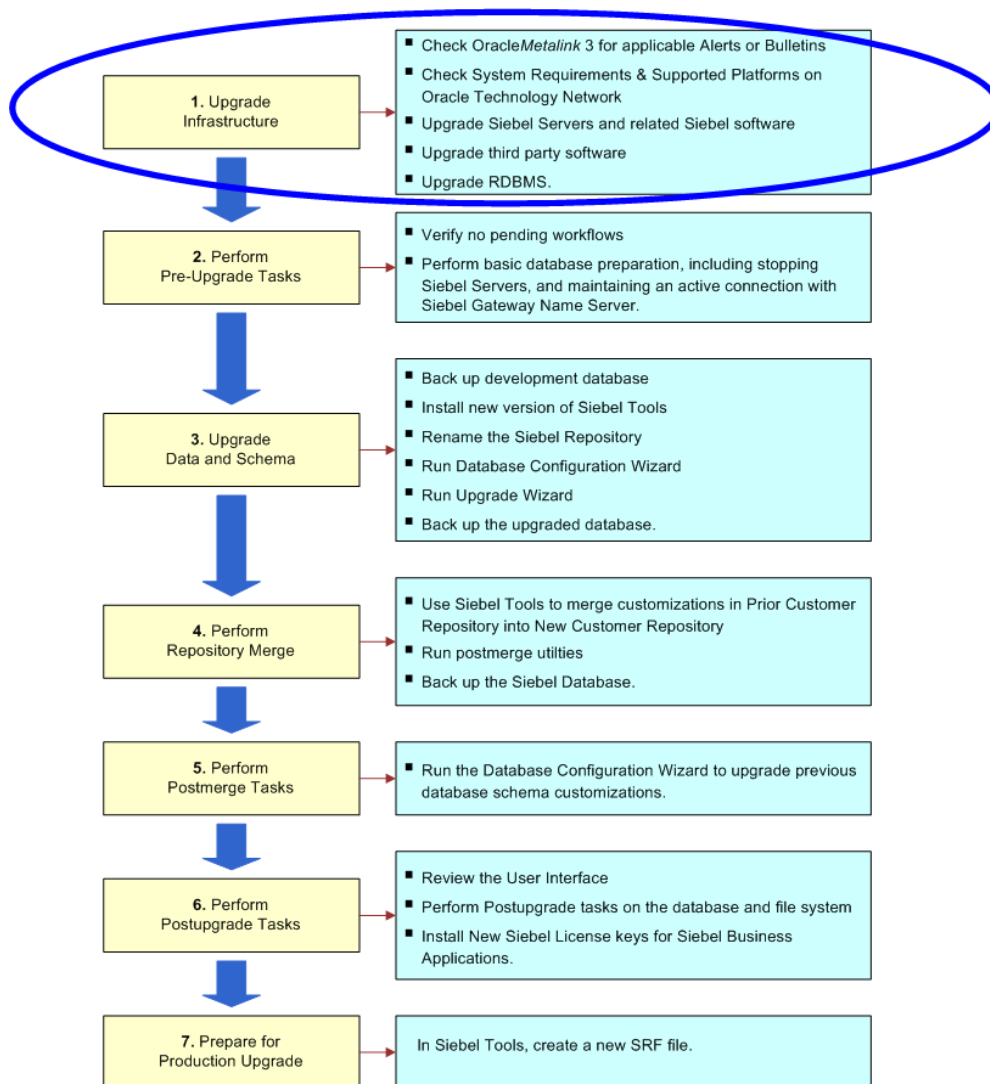


Figure 2. Phase One of Upgrade Flow

About Supported Siebel Upgrade Paths

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Supported upgrade paths are described in *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

The *Siebel Database Upgrade Guide* does not cover the following specific upgrade paths or infrastructure changes. Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

- Direct upgrades from Siebel 6.x to releases after Siebel 7.7 are not supported, except where noted. You must first upgrade from Siebel 6.x to Siebel 7.7.

However, direct upgrades from Siebel 6.2.1 of Siebel Financial Services to Siebel 7.8 or 8.0 of Siebel Industry Applications on IBM z/OS platforms are supported. To perform this upgrade, see *Siebel Database Upgrade Guide for DB2 UDB for z/OS*.

NOTE:

To upgrade from Siebel 6.x to Siebel 7.7, see the latest Siebel 7.7 version of the *Siebel Database Upgrade Guide* on the *Siebel Bookshelf*.

- Changing database platform type during an upgrade, for example changing from Oracle8i to IBM DB2.
- Changing operating system type during an upgrade, for example changing from Windows to UNIX.
- Migrating to Unicode.
- Migrating from Oracle's Siebel Industry Solutions applications or Siebel Financial Services to Siebel Business Applications.

If your enterprise uses SAP and you have implemented the Siebel Enterprise Application Integration (EAI) product, see *Siebel Connector for SAP R/3* for upgrade instructions.

- Upgrading from one base language to another. To achieve similar results, upgrade your existing base language and install the language pack for the desired language.

Related Topics

["About Siebel Upgrade Environments" on page 55](#)

About Siebel Upgrade Environments

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

The *Siebel Database Upgrade Guide* describes how to upgrade three environments:

- Development environment
- Production test environment
- Production environment

Development Environment

A development environment contains the following Oracle modules:

- Siebel Server
- Web server and Siebel Web Server Extension
- Siebel Gateway Name Server
- Siebel Database Server files installed on a Siebel Server
- RDBMS server and Siebel Database
- Siebel Tools installed on workstations running a supported Windows environment. This includes the local database running on developers' Mobile Web Clients.
- Siebel applications and test data required to verify the basic function of newly compiled SRF files.

The development environment is where developers use Siebel Tools to customize Siebel applications. The development environment upgrade merges these customizations with the new release. The merged repository and schema definitions become inputs to the production test environment upgrade and production upgrade.

Production Test Environment

The production test environment includes the following Oracle modules:

- Siebel Enterprise, including at least one Siebel Server and an RDBMS server and Siebel Database
- Web server with a Siebel Web Server Extension installed
- Siebel Gateway Name Server
- Siebel Database Server files installed on a Siebel Server
- All the Siebel applications currently installed in your production environment
- A copy of the Siebel Database installed in your production environment

You perform the following processes in the production test environment:

- Test the upgraded release to validate its function and performance before deploying it to users.
- Tune the upgrade process to minimize the time required to perform your production upgrade.

Oracle provides an upgrade tuning application that analyzes how the upgrade scripts interacted with the production test environment database. The Upgrade Tuner enables you to tune how the scripts will execute against the Siebel Database in your production environment. Tuning the scripts can significantly reduce the time required to complete the production upgrade. For this reason, the production test environment database must contain the same data volume and topography as your production database.

Production Environment

The production environment is your live business environment, where your internal and external users interact with applications and generate actual business data. The production environment includes all your Siebel Enterprises worldwide.

The upgrade process assumes all production environment databases are completely separate from the development environment and production test environment databases.

Oracle provides these tools to help you transition from production test to production:

- **Siebel Application Deployment Manager (ADM).** This module migrates administrative data such as lists of values (LOVs) from the production test environment to the production environment. For details, see *Siebel Application Deployment Manager Guide*.
- **Siebel Packager.** This application creates installation packages for use by Siebel Anywhere. For details, see *Going Live with Siebel Business Applications*.
- **Siebel Anywhere.** This application builds distribution kits for remote users. For details, see *Siebel Anywhere Administration Guide*.

Mapping Your Environments

You may have more or fewer environments than those described above. [Table 5](#) gives recommendations for mapping your environments to the ones used in the *Siebel Database Upgrade Guide*.

Table 5. Mapping Your Environments to Upgrade Processes

Environment Description	Recommended Upgrade
The environment has the following characteristics: <ul style="list-style-type: none"> ■ It is used primarily for development with Siebel Tools. ■ The Siebel Database is a subset of your production database. ■ The environment is not used for tech support or training. Developers are usually installed as Mobile Web Clients. 	Development environment upgrade.
The environment has the following characteristics: <ul style="list-style-type: none"> ■ It is intended for testing customizations before deploying them. ■ It is where you tune your upgrade SQL files to minimize production upgrade time. ■ There may be multiple upstream environments in addition to the production test environment. For example, these could include environments used by a product management group, Technical Support, and Quality Assurance. 	Production test environment upgrade.
The environment is used for live business transactions by both local and remote users.	Production environment upgrade.

Related Topics

[“About Supported Siebel Upgrade Paths” on page 54](#)

About the Siebel Database Upgrade Process

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Upgrading to a new release involves two aspects:

- The order in which to upgrade your environments
- The flow of the upgrade process within each environment

Environment Upgrade Order

If you have a development environment, you must upgrade it first, before upgrading other environments. This merges your customizations with the new release. A merged repository file and database schema file are created and become inputs to the production test environment upgrade and production upgrade.

If you do not have a development environment or have not customized your repository, no repository merge is required. You can use the repository and schema definition files included in the new release to upgrade your production test environment and production environment.

Flow of the Upgrade Within an Environment

The basic flow of the upgrade process is shown in Figure 3. This flow applies to development environment and production test environment upgrades. You also use this flow to upgrade your production environment, with the addition of several deployment steps.

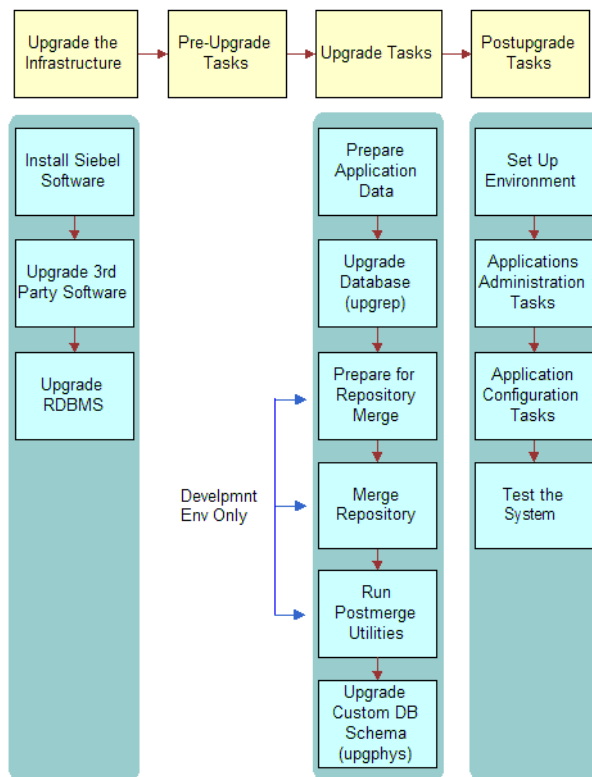


Figure 3. Flow of the Upgrade Process

Upgrade the Infrastructure

The first phase is to upgrade your hardware and software to meet system and implementation requirements, which includes upgrading the Siebel Enterprise to the new release. This action upgrades the Siebel Servers and provides the programs, scripts, input files, and other files required to merge the repository and upgrade the Siebel Database. For information on how to upgrade the infrastructure, see *Siebel Installation Guide* for the operating system you are using.

Perform Pre-Upgrade Tasks

This phase prepares the Siebel Database for upgrade and includes such tasks as closing the database connections, clearing pending workflow tasks, and disabling customized triggers.

Perform Upgrade Tasks (Development Environment)

This phase merges your customizations into the new release. This phase also upgrades the development environment database and includes these tasks:

- **Prepare application data.** These tasks prepare test data for migration.
- **Upgrade database (upgrep).** You run the Database Configuration Wizard in upgrep mode. They perform a basic upgrade of the Siebel Database schema and load repositories to prepare for the repository merge. Both upgrep and upgphys modes are run as a single step in the Database Configuration Utility.
- **Merge repository.** You use Siebel Tools to merge your existing repository with the repository in the new release. Postmerge utilities upgrade form applets and verify that applets and views are configured correctly.
- **Run postmerge utilities.** You use Siebel Tools to run a set of utilities that examine the merged repository. The utilities analyze your customizations and apply changes to them as needed to conform to the user interface in the new release.
- **Upgrade database (upgphys).** You run the Database Configuration Utilities in upgphys mode. They further upgrade the Siebel Database with changes resulting from the repository merge and complete the database upgrade.

The Database Configuration Utilities also generate the customer repository definition file and logical schema definition file that are used as input to the production test environment and production upgrades.

NOTE: During the synchronization process, custom columns in the Siebel Schema that are not in the Siebel Repository are not deleted but custom indexes in the Siebel Schema that are not in the Siebel Repository are deleted.

Perform Upgrade Tasks (Production Test Environment)

This phase upgrades a production test environment Siebel Database to the new release and includes the following tasks:

- **Prepare application data.** These tasks are about preparing application data for migration.
- **Upgrade database (upgrep).** You run the Database Configuration Utilities in upgrep mode. They perform a basic upgrade of the Siebel Database schema:
 - You run the utility in Prepare for Production mode before running it in upgrep mode. The Prepare for Production mode reviews the upgraded development environment database schema and creates input files. The upgrep mode uses these files to make schema changes to the Siebel Database.
 - The upgrep mode imports the repository and schema definition files from the development environment upgrade. It uses these files to upgrade the Siebel Database.
 - The upgphys portion runs automatically. It makes several administrative changes to table data, including updating the schema version in S_APP_VER. It does not make schema changes.

Perform Upgrade Tasks (Production Environment)

This phase upgrades a production environment Siebel Database to the new release and includes the following tasks:

- **Additive Schema Changes.** This step is optional and is run in the production environment. You can run this mode of the Database Configuration Utilities without taking the database offline. This mode makes schema changes that do not affect the operation of the application. This reduces the amount of time the production database must be offline to perform the upgrade. Testing Additive Schema Changes is also included in the process checklist for tuning the upgrade. This allows you to verify Additive Schema Changes in the production test environment before running it in the production environment.
- **Prepare application data.** These tasks are about preparing application data in the production database for migration.
- **Upgrade database (upgrep).** In the production test environment, you start the Database Server Configuration Utilities in upgrep mode, and enter configuration information for the production environment. This includes an ODBC definition for connecting to the production Siebel Database.

This step causes the wizard to use the SQL for upgrading the production test database to upgrade the production database.

The SQL generated for the production test database upgrade is preserved, and no new SQL is generated. This SQL has been tuned using the Upgrade Tuner and has been revised by the Prepare for Production mode. You do not have to repeat these two steps before upgrading the production database.

You do not have to run the Database Configuration Utilities in upgphys mode. The upgphys steps are included in the production environment upgrep.

Perform Postupgrade Tasks

This phase is where you set up the environment, configure applications, and test the system as follows:

- **Set up the environment.** These tasks set up the postupgrade environment, which includes extracting the developers' databases and running database statistics.
- **Application administration.** These tasks set up the applications and include such tasks as setting up user access and visibility of views and screens.
- **Application configuration.** These tasks prepare applications for testing, including data migration for specific applications.
- **Test the system.** These tasks test the system. For development environment upgrades, you perform basic unit tests to verify the operation of the application. For production test environment upgrades, you perform a full suite of regression and stress tests to verify the system is ready for production.

Related Topics

["About Siebel Additive Schema Changes Mode" on page 70](#)

[“About the Siebel Database Configuration Utilities” on page 62](#)

[“About the Siebel Upgrade Wizard and Driver Files” on page 68](#)

[“About the Siebel Repository Merge” on page 91](#)

[“About the Database Configuration Utilities” on page 112](#)

About the Siebel Database Configuration Utilities

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

The Database Configuration Utilities are part of the Siebel Database Configuration Wizard. The Database Server Configuration Utilities interactively gather the information required to perform the following operations:

- **Install the Siebel Database.** These wizards set up the Siebel Database in the RDBMS as part of a first-time installation of Siebel applications. They can also add a language to the Siebel Database Server installation.
- **Upgrade the Siebel Database.** These wizards upgrade the Siebel Database to a new release in a development environment, production test environment, or production environment.
- **Import or export a Siebel repository.** These wizards move entire repositories between database environments with the same schema definition. A wizard is also provided that adds installed languages to a Tools repository.
- **Apply Additive Schema Changes.** As part of upgrading to a new Release, this wizard generates a SQL script. The script contains schema changes that you can make to the production database without taking it offline. This reduces the production database downtime required for the upgrade.
- **Migrate a Siebel repository.** This wizard migrates repositories between databases and synchronizes the target database schema with the migrated repository schema definition.
- **Run database utilities.** This group of wizards perform the following functions:
 - Synchronize a database schema with the Siebel Repository schema definition.
 - Add a license key to a Siebel Database (MSSQL only).
 - Convert existing time-keeping management to UTC.
 - Convert existing Lists of Values (LOVs) to Multilingual Lists of Values (MLOVs).

Upgrade the Siebel Database

When you run the Database Configuration Wizard > Upgrade the Database, the wizard prompts you for the upgrade environment (development or production) and the upgrade phase (upgrep or upgphys). The Wizard then prompts you for the information it needs to perform the upgrade.

After collecting and confirming the information, the wizard creates an upgrade configuration file and calls a driver that uses the file to set up the SQL scripts required to upgrade your database.

After you run the Database Configuration Wizard, you run the Upgrade Wizard. The Upgrade Wizard opens a state file containing the steps for the upgrade. These steps execute the SQL scripts created after you ran the Database Configuration Wizard.

To upgrade a development environment, production test environment, and production environment, you must run the Database Configuration Wizard (and Upgrade Wizard) several times, as shown in [Table 6](#).

Table 6. Database Configuration Utilities Modes

Upgrade Step	Select This Environment Type	Select This Upgrade Option
Development env. upgrep	Development	upgrep
Development env. upgphys	Development	upgphys
Prod. test env. prepare for production	Production	Prepare for Production
Prod. test env. upgrep and upgphys	Production	upgrep and upgphys
Apply Additive Schema Changes	Production	Apply Additive Schema Changes
Production env. upgrep and upgphys	Production	upgrep and upgphys

Figure 4 shows how the Database Configuration Wizard (and Upgrade Wizard) work together with the Siebel Tools repository merge to upgrade your environments.

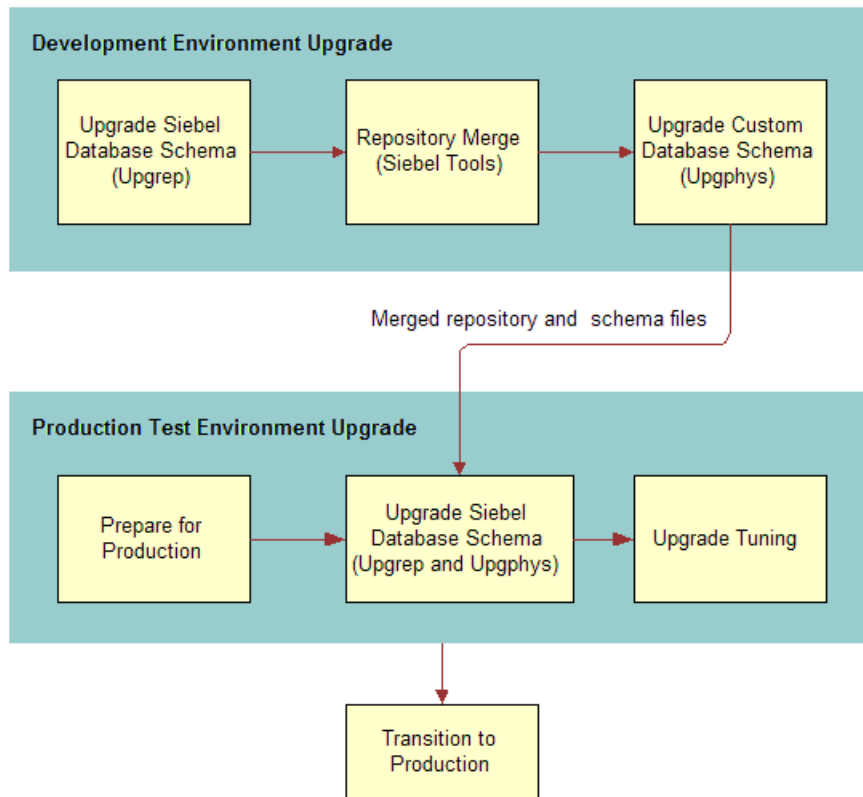


Figure 4. Upgrade Flow

Development Environment Upgrades

For a development environment upgrade, you run the utility twice, once in each of the following modes:

- **upgrep.** This mode makes the following changes:
 - Removes interface tables and database triggers
 - Populates columns that must change from NULL to NOT NULL
 - Creates new tables. Merges existing tables.
 - Prepares for index creation. Verifies that there are no unique key violations.
 - Creates indexes
 - Imports seed data

- Imports the version x.x Prior Customer Repository, New Siebel Repository, and New Customer Repository
- Makes modifications to repository objects to prepare for the repository merge
- Updates primary children foreign key references
- Performs miscellaneous file actions
- **upgphys**. This mode makes the following changes:
 - Synchronizes the Siebel Database schema to the logical schema definition in the merged repository
 - Deduplicates intersection tables
 - Exports repository object definitions to a file, `custrep.dat`
 - Exports the logical schema definition to a file, `schema.ddl`These two files are used as input to the production test environment and production environment upgrades.
- Renames the New Customer Repository to Siebel Repository
- Updates the schema version in `S_APP_VER`

Production Test Environment Upgrades

For a production test environment upgrade, you run the utility three times:

■ Prepare for production.

This mode does the following in the production test database:

- Examines the development environment database and generates SQL that deduplicates intersection tables and sets up support for database aggregation.

In the production test environment, you must define an ODBC connection to the development environment database before performing this upgrade step.

- Compares the repository schema and the physical database schema. Generates a file that lists indexes present in the physical schema that are not present in the repository schema. You can decide whether to remove these indexes.

■ upgrep. This mode makes the following changes:

- Removes interface tables and database triggers
- Populates columns that must change from NULL to NOT NULL
- Uses the `custrep.dat` and `schema.ddl` files from the development environment upgrade to create new tables and merge existing tables
- Prepares for index creation. Verifies that there are no unique key violations.
- Creates indexes
- Imports seed data
- Updates primary children foreign key references

How the Siebel Database Upgrade Works ■

- Performs miscellaneous file actions
- The upgphys portion runs automatically. It makes several administrative changes to table data, including updating the schema version in S_APP_VER. It does not make schema changes.
- **Upgrade Tuning.** These steps are optional. You can use the Logparse utility to tune the SQL generated for the upgrade. Logparse identifies SQL commands that are not used, or that can be run in parallel. This reduces the time required to perform the production database upgrade.

You can also test the Additive Schema Changes script to verify that it does not adversely affect the operation of the application. Additive Schema Changes allows you to run part of the schema upgrade without taking the database offline. This reduces production database downtime.

Production Environment Upgrades

After you have completed testing applications and have tuned the upgrade SQL commands, you perform the production upgrade. The production upgrade uses the SQL commands generated in the production test environment. In the production test environment, you perform the following steps to upgrade your production environment Siebel Database:

- Define an ODBC connection to the production environment Siebel Database
- **Apply Additive Schema Changes.** This step is optional and is run in the production environment. You can run this mode of the Database Configuration Utilities without taking the database offline. This mode makes schema changes that do not affect the operation of the application. This reduces the amount of time the production database must be offline to perform the upgrade.
- **Prepare for Production.** This step is not required. You ran it as part of the production test environment upgrade. The required changes have already been made to the upgrade SQL commands.
- **upgrep.** Run the Database Configuration Utilities in the production test environment. Enter information for the production environment (not the production test environment), including the new ODBC definition.

The Database Configuration Utilities update the upgrade configuration file with production environment information. A lock file that was set when you ran the utility previously, prevents new SQL from being generated. This preserves the SQL you have revised and tuned.

When you run the Upgrade Wizard, it reads the production environment information from the configuration file and uses the production test environment SQL commands to upgrade the production environment Siebel Database.

The upgrep step makes the same changes as when it ran in the production test environment. This includes automatically running the production upgphys portion of the upgrade.

How the Upgrade Configuration File and SQL Files Are Created

When you run the Database Server Configuration Utilities, the following actions are performed:

- Collect configuration information

- Create a master upgrade configuration file (UCF). This file maps the information you entered in the Database Server Configuration Utilities to environment variables.
- Forward the information to an SQL generator that creates or populates SQL files with the required commands. The SQL generator extracts these commands from an intermediate XML file containing all the SQL commands required for an upgrade.
- Prompt you to start the Upgrade Wizard.

In some cases, you will need to modify the generated SQL files as required by Bulletins, Alerts, or *Siebel Release Notes* on *OracleMetaLink* 3 before you run the Upgrade Wizard. To do this, answer No when prompted to run the Upgrade Wizard. Then, edit the SQL files and manually launch the Upgrade Wizard.

How to Locate Master Configuration Files

Master upgrade configuration files are stored in the following location:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$(SIEBEL_ROOT)/bin`

Master upgrade configuration files use the following naming convention:

`master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`

where:

- *UPGRADEOPTION* Indicates the upgrade process you are performing
 - *upgrep* Indicates Siebel Database schema upgrade
 - *upgphys* Indicates Custom database schema upgrade indicates
 - *prepare_for_production_upgrade* Indicates Prepare for production upgrade
- *ENVIRONMENT* Indicates the environment that you are upgrading
 - Development environment upgrades *dev*
 - Production environment upgrades *prod*
- *VERSION* Indicates the version from which you are upgrading

For example, if you are upgrading from Release 7.0.4, the master configuration file generated from the development environment upgrep is as follows:

`master_upgrep_dev_704.ucf`

The master configuration file generated from the Prepare for Production mode is as follows:

`master_prepare_for_production_upgrade.ucf`

Related Topics

[“About Siebel Additive Schema Changes Mode” on page 70](#)

[“About the Siebel Upgrade Wizard and Driver Files” on page 68](#)

About the Siebel Upgrade Wizard and Driver Files

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

The Upgrade Wizard performs the upgrade. It uses the information collected by the Database Server Configuration Utilities to execute commands and SQL scripts. These commands and SQL scripts are contained in driver files. Driver files are in ASCII text format and are organized into steps. The Upgrade Wizard reads the steps in the driver files and performs the commands contained in each step.

In a driver file, steps are separated by a blank line, and each step begins with a File Execute Entry number. The key part of each step is the command or script to be executed. When an SQL script is specified, you can review the script and see what changes it will make to the Siebel Database before running the Upgrade Wizard.

Driver files are provided for each of the major upgrade operations. Examples of driver files are as follows:

- driver_upgrep_dev_80.ucf
- driver_upgphys_dev_80.ucf
- driver_upgrep_prod_80.ucf

Here is an excerpt from a driver file that controls a development environment upgrep from Siebel CRM 8.x on Oracle Database. The excerpt contains three steps:

```
[Sql File Entry 0]
Type = DbSql
File Name = rename_existing_repositories.sql
Use Table Owner = 1
Use System Admin = 0
IgnoreAllDDLerrors = 0
IgnoreAllDMLerrors = 0
Argument 0 = $SiebelVersion
Title = Verify Repository Name
Title Message Num = 0
Estimated Disk Space = 0
Backup Db = 0
Parallelizable Item = 0
Prompt User For Status = 0
```

[File Execute Entry 1]

Type = FileExecute

File Name = \$SiebelRoot\bin\odbcsql

Check Return Code = 1

Return Code Compliance = 0

16 Bit App = 0

Command Line = /s "\$ODBCDataSource" /u \$TableOwner /p \$TablePassword /separator / /
a /c rem /l \$SiebelLogDir/dropif-db.log \$DbsrvrRoot/\$DatabasePlatformform/dropif-db.sql
/v

Number of 10 Second Wait Loops = 2000

Return Code = 0

Title = Drop interface tables

Title Message Num = 0

Estimated Disk Space = 0

Backup Db = 0

Parallelizable Item = 0

Prompt User For Status = 0

[File Execute Entry 2]

Type = FileExecute

File Name = \$SiebelRoot\bin\odbcsql

Check Return Code = 1

Return Code Compliance = 0

16 Bit App = 0

Command Line = /s "\$ODBCDataSource" /u \$TableOwner /p \$TablePassword /separator / /
a /c rem /l \$SiebelLogDir/trigdrop-db.log \$DbsrvrRoot/\$DatabasePlatformform/trigdrop-
db.sql /v

Number of 10 Second Wait Loops = 2000

Return Code = 0

Title = Drop triggers

Title Message Num = 0

Estimated Disk Space = 0

Backup Db = 0

Parallelizable Item = 0

Prompt User For Status = 0

How To Locate Upgrade Driver Files and SQL Scripts

Driver files are stored in the following location:

Windows: *DBSRVR_ROOT\PLATFORM\UPGRADE\VERSION*

UNIX: *DBSRVR_ROOT/bin/PLATFORM/UPGRADE/VERSION*

where:

- *PLATFORM* indicates the database type, for example ORACLE
- *VERSION* indicates the version from which you are upgrading, for example V8.0

For example, if you are upgrading from Siebel 7.7, the driver file for the development environment upgrep is as follows:

`driver_upgrep_dev_80.ucf`

Most of the SQL scripts generated or populated for the upgrade are stored in the same directory as the driver file. The remaining SQL scripts are stored in the *PLATFORM* directory.

Related Topics

[“About the Siebel Database Upgrade Process” on page 58](#)

[“About the Siebel Database Configuration Utilities” on page 62](#)

[“Example of a Siebel Development Upgrade Flow” on page 84](#)

[“About the Siebel Repository Merge” on page 91](#)

About Siebel Additive Schema Changes Mode

Upgrades from: All Supported Siebel releases.

Environments: Production test, production.

All databases. This topic applies to all supported databases.

Windows and UNIX only. This topic does not apply to IBM z/OS platforms.

As of Siebel CRM 8.0, Apply Additive Schema Changes mode is part of the Database Server Configuration Utilities. The purpose of Additive Schema Changes is to reduce the downtime required for the production database upgrade.

Additive Schema Changes generates an SQL script. You run the script on the live production database before you take it offline to perform the production upgrep. The script performs certain upgrade steps that are normally part of the production upgrep. This reduces the number of steps the upgrep must perform while the database is offline, and thus reduces database downtime.

Running the SQL script is optional. If you do not run it, the regular production upgrep performs the steps in the SQL script.

Types of Changes

Additive Schema Changes make the following types of schema changes to support the new release. These changes do not adversely affect data integrity or database normalization:

- Creating new tables.
- Adding columns to an existing table. The column must either be specified as null, or if the column is not null, it must have a specified default value.
- Creating non-unique indexes on new tables.
- Increasing column sizes for numeric or varchar columns. The column must not be the basis for a picklist. Also, the resultant cumulative row size must not be larger than the data page size.
- Revising a not-null column to null.
- Revising data type from char to varchar.

Implementation of Additive Schema Changes

You generate the Additive Schema Changes script by starting the Database Server Configuration Utilities and choosing Apply Additive Schema Changes. A wizard prompts you for environment information similar to other upgrade modes.

After you review and confirm your entries, the wizard populates the `platform\driver_additive_gen.ucf file`, where *platform* is the database type (DB2, MSSQL, Oracle).

The Upgrade Wizard, then performs the following steps:

- Reads the `master_additive_gen.ucf` file to obtain environment information and the name of the driver file.
- Reads the `driver_additive_gen.ucf` file and executes the `ddlmp2` command it contains.
- The `ddlmp2` command performs the following steps:
 - Reads the tables and indexes from the `schema.ddl` file.
 - Connects to the specified database and reads the existing schema and indexes.
 - Identifies the upgrade schema changes that can be made with the database online.
 - Writes these upgrade changes to the `schema.additive.sql` script.
 - Writes all actions to `schema.additive.log`.

The Upgrade Wizard does not execute `schema.additive.sql` against the database.

How To Use Additive Schema Changes

Use the following workflow to implement Additive Schema Changes:

- 1 In the production test environment, complete all development and upgrade testing. The `schema.ddl` file must be in its final form.
- 2 As part of running the Tuning Upgrade Performance process checklist, restore the production test environment database to its original, unupgraded state.
- 3 Run the Database Server Configuration Utilities in Apply Additive Schema Changes mode, and generate the `schema.additive.sql` script.
- 4 **Windows.** You will not be prompted whether you want to run the Upgrade Wizard. Instead, the Upgrade Wizard starts automatically, and creates the `schema.additive.sql` script. The Upgrade Wizard does not run the script against the database.
- 5 **UNIX.** After the Database Server Configuration Utilities exit, run the Upgrade Wizard to generate the `schema.additive.sql` script. The Upgrade Wizard does not run the script against the database.
- 6 Review the `schema.additive.log` file. Verify that the script was created successfully.
- 7 Review the `schema.additive.sql` script. Verify that it will not make changes that conflict with customizations. Edit the script as required.
- 8 Use an SQL editor to apply the `schema.additive.sql` script to the database.
- 9 Verify that applications function normally. If necessary, revise `schema.additive.sql` and reapply it to the database.
- 10 Complete the remainder of the Tuning Upgrade Performance process checklist. See [“Process of Tuning Siebel Upgrade Performance” on page 40](#).
- 11 When you are ready to upgrade the production database, run the `schema.additive.sql` script against the production database before doing the production upprep.
- 12 There is no need to then run database statistics. The schema changes are not useful for optimizing applications function before the upgrade.

The steps in this workflow have been incorporated into the process checklists you use to perform both upgrade tuning and the production environment upgrade.

Related Topics

[“About the Siebel Database Upgrade Process” on page 58](#)

[“About the Siebel Database Configuration Utilities” on page 62](#)

About the Siebel Database Upgrade Log Files

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows and UNIX only. Does not apply to IBM z/OS.

The Upgrade Wizard writes log files that provide detailed information on the upgrade processes, for example development-environment upprep and upgphys. These log files provide detailed information on the upgrade steps, and they also list all errors. The Logparse utility analyzes and summarizes these log files.

Upgrade Wizard Log Files

The Upgrade Wizard writes the log files for a process to the following directory:

Windows: `SI EBEL_ROOT\log\PROCESS`

UNIX: `$SI EBEL_ROOT/log/PROCESS`

where `PROCESS` defaults to `UPGRADEOPTION_ENVIRONMENT_VERSION`.

For example, the log files for a Siebel 8.0 development environment upprep appear in the following locations:

Windows: `SI EBEL_ROOT\log\upprep_dev_80`

UNIX: `$SI EBEL_ROOT/log/upprep_dev_80`

You can select a different directory for the log files in the Log Output Directory screen in the Database Server Configuration Utilities.

Each subdirectory in the log directory contains the following files:

- **Output.** This subdirectory contains the Upgrade Wizard log files.
 - `upgwi z.log` (`srvrupgwi z.log` on UNIX). The Upgrade Wizard log for the process. This log contains detailed information on how the Upgrade Wizard ran. If you run the process more than once, the log name increments, for example `upgwi z2.log` or `srvrupgwi z2.log`.
 - Windows only. Log files with the name `upgwi zn(nnn)` can be ignored. For example, `upgwi z1(001).log`, `upgwi z1(002).log`, and so on.
 - Windows only. Log files with the name `sw_cfg_xxx` can be ignored. (These are located in the log directory.)
 - UNIX only. Log files with the name `srvrupgwi zn_nnn` can be ignored. For example, `srvrupgwi z1_01.log`, `srvrupgwi z1_02.log`, and so on.
 - Log files with the name `si ebel .log` can be ignored.

TIP: Each upgrade process, for example, a development upprep, has a driver file located in `\dbsrvr\platform\upgrade\version`. An example of a driver file: `driver_upprep_dev_77.ucf`. The Upgrade Wizard performs the steps in the driver file. If a step calls a utility, such as `ddlmp` or `dataimp`, the `/l` option in the command syntax specifies the name of the log file. Unless you modified the output location for the log, `$Siebel LogDir` is `\siebsrvr\log\process`.

- **State.** This subdirectory contains the state.log file.

- **state.log.** The state.log file lists each step in the upgrade process and whether the step completed successfully.

The output and state log file directories are automatically archived on subsequent runs of a process that completes successfully. (The names of subsequent log directories are appended with `_1`, `_2`, and so on.) To preserve disk space, periodically delete or save log directories to another location.

- **Summary.** This subdirectory is generated by the Logparse utility. The Logparse utility also generates a report that summarizes the upgrade, `summary.html`. The `summary.html` file contains links to the html pages in the summary subdirectory.
 - **Summary.html.** This file is created by the Logparse utility. It summarizes the log files in the `output` directory. Use this file to verify that all upgrade steps completed successfully and to determine if there were any errors.
 - **Summary.txt.** Text file created by the Logparse utility. This file contains the same information as `summary.html`.
- NOTE:** If a browser is not installed on the machine you are using, review `summary.txt` instead of `summary.html`.
- **Summary.xml.** File produced by the Logparse utility during production upgrades and used by the Siebel Upgrade Tuner to parallelize table creation and index creation and to inactivate SQLs that affect no rows. (Development upgrades do not produce a `summary.xml` file.)
 - **Upgtuner_ftp_get.txt. (UNIX Only).** File produced by the Logparse utility during production upgrades. You can use this file to transfer the upgrade scripts from your UNIX machine to a Windows temporary directory to perform upgrade tuning. (Development upgrades do not produce `upgtuner_ftp_get.txt`.)
 - **Upgtuner_ftp_put.txt. (UNIX Only).** File produced by the Logparse utility during production upgrades. After you tune your production upgrade scripts, you can use this file to transfer the upgrade scripts from the temporary directory on your Windows machine back to your UNIX environment. (Development upgrades do not produce `upgtuner_ftp_put.txt`.)

The Logparse Utility

For each upgrade process, the Logparse utility analyzes the log files in `si_ebsrvr\log\process`. It then summarizes the log files and provides several reports. The Logparse reports provide the following information:

Parameters report

The Parameters report lists the total time required for the upgrade process. It also lists environment variables and input files used by the upgrade process. Use this report to verify that the upgrade was set up correctly.

Step/Error Summary

The Step/Error Summary lists each step in the upgrade process and whether it completed successfully. It also provides a link to a detailed step summary for each step.

The detailed step summary lists database errors returned by the Upgrade Wizard, SQL scripts, and errors returned by the upgrade utilities called by the driver files, such as ddlimp, and dataimp.

For Oracle Database and IBM DB2, the detailed step summary lists only unacceptable errors that must be corrected. The summary does not list benign errors.

For MSSQL, the detailed step summary lists all errors, including benign errors. An errors file (errors.xls and errors.rtf) is provided that lists benign errors for each database type. When you find an error in the summary.html file, you must compare it with the error file to determine if it is benign.

Performance report

The performance report provides a list of the longest running SQL scripts and a list of SQL scripts returning 0 rows. The lists include a link to each script. Upgrade Tuner uses this information as input to help you tune upgrade performance.

The report also provides two lists of DDL commands executed during the upgrade process. The DDL Summary lists the tables that were created, altered, or deleted by the ddlimp utility. The Native DDL Summary lists tables that were created, altered, or deleted by SQL scripts and includes a link to each script.

About the Siebel Case Insensitivity Wizard

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Databases: All databases.

For Siebel CRM 8.0, query features are enhanced to provide indexes that directly support case and accent insensitive (CIAI) queries on eligible text columns. The CIAI Wizard configures specified columns for CIAI queries by defining CIAI columns and CIAI indexes in the repository. The Wizard also sets the Default Insensitivity property for these columns to DB Case & Accent.

The purpose of the enhanced CIAI features is to improve query effectiveness and performance. Running the CIAI Wizard is optional.

You run the CIAI Wizard during the development environment upgrade, after the upgrep and before the repository merge.

Overview of What the CIAI Wizard Does

The CIAI Wizard performs the following functions in the repository to configure columns to support CIAI queries. No columns or indexes are created in the Siebel Database until you synchronize the repository to the Siebel Database. The columns you want to configure for CIAI queries are called base columns:

How the Siebel Database Upgrade Works ■

- Validates the syntax of all records if an input file is used.
- Validates that all specified tables and columns are eligible for CIAI configuration.
- For each eligible base column, defines a new CIAI column. The CIAI column contains the data in the base column converted to uppercase.
- If you select the Single or Copy All index strategy, the Wizard defines an index on the CIAI column.
- If you select the Copy All index strategy, defines a copy of all indexes that have the base column as a key. The new indexes reference the CIAI column instead of the base column.
- Sets the Default Insensitivity property for the base column to DB Case & Accent.
- Sets flags and performs other configuration operations in the repository to support CIAI queries.

The CIAI Wizard can also be run in a special mode to set the Default Insensitivity property on columns that do not have any indexes defined.

The main purpose of the CIAI query enhancements is to provide indexes that can be used for case insensitive searches. The database does not have to perform table scans to locate records. This enhancement allows the database to perform case insensitive searches more quickly.

For example, in S_CONTACT, you configure the column LAST_NAME for CIAI queries. The CIAI Wizard defines a column called LAST_NAME_CI. When you query for the name Smith, the Object Manager creates a query similar to the following (IBM DB2):

```
SELECT column list FROM S_CONTACT
WHERE LAST_NAME_CI indicates SMITH
```

The database then uses the CIAI index on LAST_NAME_CI to locate the records.

For columns where Force Case is set, the data in the database is all in the same case. No CIAI columns or indexes are needed. The Object Manager uses the indexes already defined on the base column to retrieve records.

How CIAI Columns and Indexes Are Implemented in the Database

In a development environment upgrade, you apply the repository schema definition to the physical database. How CIAI columns and indexes in the repository are implemented in the physical database depends on the database type. For example, on Microsoft SQL, CIAI columns are implemented as calculated columns. On Oracle Database, CIAI columns are implemented as function-based indexes. On IBM DB2, CIAI columns are implemented as schema columns.

CIAI Wizard Modes

You can run the CIAI Wizard in one of two modes:

- **Configure specified columns.** You can specify the columns you want to configure for CIAI queries by using an input file or by selecting the files manually:

- **Use an input file.** The Wizard reads the input file and configures the columns in the file for CIAI queries. Using an input file allows you to control which configuration options the Wizard uses. Oracle provides recommended input files. These files contain columns frequently used in searches. Use an input file if you want to enable large numbers of columns or to use methods or index strategies other than the defaults.
- **Select columns manually.** You can use the Tools Object Explorer to manually select and configure specific columns for CIAI queries. The Wizard uses configuration defaults to configure these columns. If you want to modify the configuration options, you can export the configuration strings to a text file, edit them, and run the Wizard using the edited file as an input file.
- **Specify how queries are built for columns without indexes.** The CIAI Wizard defines CIAI columns and indexes only on columns that already have indexes defined. However, for columns without indexes that meet all other eligibility criteria, you can run the CIAI Wizard in a special mode to change the Default Insensitivity property from None to DB Case & Accent. In queries, the column values are then converted to uppercase before being compared. This allows searches to be both case and accent insensitive.

For example, in S_CONTACT, assume the column LAST_NAME has no indexes defined on it. You run the CIAI Wizard to set Default Insensitivity to DB Case & Accent. When you query for the name Smith, the Object Manager uses a query similar to the following (IBM DB2):

```
SELECT column list FROM S_CONTACT
WHERE UPPER(LAST_NAME) LI KE UPPER(Smi th)
```

Input File Format

For configuring eligible table columns, the CIAI Wizard can accept a comma-delimited csv file as input. Each line in the file is one record that defines one column to be configured for CIAI queries.

The record format is as follows:

```
<Table Name>, <Column Name>, <Method>, <Index Strategy>, <Operation>
```

The fields in the record are explained in the following example of a record:

```
S_CONTACT, EMAIL_ADDR, Database, Copy All, On
```

Only the first two items in the record are mandatory (Table Name and Column Name). For records where items are omitted, the CIAI Wizard inserts the default value for the items.

If you omit an item from a record, you must still provide the item's delimiting commas. Here is a record with the index strategy item omitted:

```
S_CONTACT, EMAIL_ADDR, Database, , On
```

The CIAI Wizard does not provide special handling for denormalized columns. To configure CIAI queries on denormalized columns, you must include them in an input file.

Oracle provides a recommended input file for Siebel Business Applications and for Siebel Industry Applications. The input files have a .csv extension and are located in the following directory:

Windows: `Tools\Install_dir\objects`

Eligibility Criteria

The CIAI Wizard verifies that all records in an input file or in columns that you manually select meet the following eligibility criteria:

- The table and column must exist in the repository.
- The column must be active and belong to the specified table.
- The table type must be supported for CIAI configuration.
- The column functional type must be supported for CIAI configuration.
- The column physical type must be supported for CIAI configuration.
- The column must have one or more indexes already defined on it. If no indexes are defined on the column, but the column is otherwise eligible, the Wizard accepts the column, but it does not create a CIAI column or any CIAI indexes for the column. The Wizard sets the Default Insensitivity to DB Case & Accent.

Table Name—Supported Table Types

Table 7 lists the table types that can be configured for CIAI queries.

Table 7. Supported Table Types

Table Type	Can Be Configured for CIAI Queries?
Data (Intersection)	Yes
Data (Private)	Yes
Data (Public)	Yes
Database View	No
Dictionary	No
Extension	Yes
Extension (Siebel)	Yes
External	No
Interface	No
Log	No
Repository	No

Column Name—Supported Column Functional Types

Table 8 lists the column functional types that can be configured for CIAI queries.

Table 8. Supported Functional Column Types

Column Functional Type	Can Be Configured for CIAI Queries?
Data (Private)	Yes
Data (Public)	Yes
Denormalized	Yes
Extension	Yes
External	No
IFMGR: Exists	No
IFMGR: FKey	No
IFMGR: Status	No
IFMGR: ROW_ID	No
System	No

Column Name—Supported Physical Column Types

Table 9 lists the physical column types that can be configured for CIAI queries.

Table 9. Supported Physical Column Types

DB Value	Physical Type	
	Maps To:	Can Be Configured for CIAI Queries?
C	Char	Yes
D	Date	No
N	Number	No
S	Time Stamp	No
T	Date Time	No
U	UTC Date Time	No
V	Varchar	Yes
X	Text	Yes
L	CLOB	Yes

Text and CLOB are accepted by the Wizard as valid. However, the Wizard does not create a CIAI column or CIAI indexes for these two types. The Wizard sets Default Insensitivity to DB Case & Accent.

Method and Index Strategy

The CIAI Wizard method and index strategy determine how the Wizard configures CIAI queries for a column.

There are two methods:

- **Force Case.** The Force Case method does not create a CIAI column or indexes. Use the Force Case method for columns where the Force Case property is set.

The Force Case property causes column data to be set to the same case (FirstUpper, lower, upper) before being written to the database. Since all the data in the base column are in the same case, the base column and its indexes can be used for case insensitive queries. A CIAI column and indexes are not needed.

The CIAI Wizard considers Force Case to be set for a column when Force Case indicates FirstUpper, Lower, or Upper. The CIAI Wizard does not consider Force Case to be set when Force Case indicates none or is null.

- **Database.** The Database method defines a CIAI column for the base column and uses an *index strategy* to create indexes. Index strategies determine how indexes are defined for the CIAI column, as described in [Table 10](#).

Table 10. Index Strategies Used by the Database Method

Index Strategy	Description
None	The Wizard defines no new columns or indexes. The Wizard sets the Default Insensitivity to DB Case & Accent.
Single	The Wizard defines a new CIAI column and defines a single index on it. For every index the base column participates in, the Wizard does not create another index that references the CIAI column. The Wizard sets the Default Insensitivity to DB Case & Accent.
Copy All	The Wizard defines a new CIAI column and a CIAI index for the column. In addition, for every index the base column participates in, the Wizard defines a copy of that index. The copy references the CIAI column instead of the base column. The Wizard sets the Default Insensitivity to DB Case & Accent.

For indexes with multiple columns as keys, for the first key where the column becomes CIAI enabled, the Wizard defines a copy of the index. In the copy, the key references the CIAI column instead of base column. For each additional key that is CIAI enabled, the Wizard deletes the index copy in the repository and redefines it so that keys reference the additional CIAI columns.

For example:

- The S_CONTACT table has Base Column A with Index A that has two columns as keys, LAST_NAME and FST_NAME.
- You then select the column LAST_NAME for CIAI queries and specify the Copy All index strategy.
- The CIAI Wizard defines the column LAST_NAME_CI and a CIAI index for the new column.
- The Wizard also defines Index B for Base Column A by copying Index A and specifying LAST_NAME_CI and FST_NAME as keys.
- You then select the column FST_NAME for CIAI queries.
- As part of configuring FST_NAME for CIAI queries, the Wizard does the following in the repository:
 - Defines a new column FST_NAME_CI.
 - Deletes Index B on Base Column A and redefines it with the keys LAST_NAME_CI and FST_NAME_CI.

Operation—Inactivating CIAI Configuration

The Operation field in the configuration syntax controls whether the columns and indexes created by the CIAI Wizard are active. The available settings are On and Off. The default is On.

Use the Operation field to deactivate the CIAI configuration of columns. When you run the CIAI Wizard against a column and specify Operation indicates Off, the wizard does the following:

- Inactivates the CIAI column created on the base column.
- Inactivates the CIAI index created for the CIAI column.
- For indexes that the base column participates in, sets the related CIAI indexes to inactive.

Note that the CIAI Wizard does not delete CIAI columns or CIAI indexes in the repository.

You cannot use Operation indicates Off to set the Default Insensitivity property on a column from DB Case & Accent to None.

CIAI Wizard Defaults

The only required inputs for the CIAI Wizard are Table Name and Column Name. If you omit the other fields, the CIAI Wizard uses the default settings.

Methods

The CIAI Wizard uses the following default settings for the methods:

- In the repository, if the Force Case property is set on the column, then the Wizard uses the Force Case method.
- If the Force Case property is not set, then the Wizard uses the Database method.

Index Strategy

The Wizard uses the following defaults for index strategy:

- If the method is Force Case, the Wizard sets the index strategy to None.
- If the method is Database, and the base column does not have indexes defined on it, the Wizard sets the index strategy to None.
- If the method is Database, and the base column has indexes defined on it, the Wizard uses the Copy All index strategy.

If the Wizard uses None as an index strategy, then the Wizard does not define new columns or indexes. It sets Default Insensitivity to DB Case & Accent.

This setting means that the Wizard has the following default behaviors:

- When the Force Case property is set on a column, the Wizard does not define columns or indexes.
- If the column has no indexes defined on it, the Wizard does not define columns or indexes.

Note that in both cases, the Wizard accepts the column as eligible but does not define columns or indexes. These default behaviors define the implicit eligibility requirements.

Operation

If you omit Operation, the Wizard sets Operation to On, regardless of the method, or index strategy.

Running the CIAI Wizard Multiple Times

You can run the CIAI Wizard multiple times. Typically, you would do this to tune the CIAI configuration of columns. If you have already run the CIAI Wizard once to configure columns, you may run it again to perform additional tasks. [Table 11](#) lists operations that you can perform when running the CIAI Wizard again:

Table 11. Running the CIAI Wizard Multiple Times

Task	How to Run the CIAI Wizard
Configure new columns to support CIAI queries.	You can use either an input file, or you can select files manually.
Inactivate CIAI configuration for specified columns.	Run the CIAI Wizard using an input file that specifies the desired columns. For each column, specify Operation indicates Off.
Run the CIAI Wizard to change the Default Insensitivity property from None to DB Case & Accent for eligible columns without indexes.	Start the CIAI Wizard from the Tools > Utilities > Case Insensitivity menu and select the "Enable for all unindexed columns" mode.
For columns already configured, change method from Force Case to Database.	You can use either an input file, or you can select files manually.

Table 11. Running the CIAI Wizard Multiple Times

Task	How to Run the CIAI Wizard
For columns already configured, change method from Database to Force Case.	Run the CIAI Wizard using an input file that specifies the desired columns. For each column, specify Operation indicates Off. This deactivates the CIAI column and CIAI indexes. Verify that the base columns have the Force Case property set.
For columns already configured, change index strategy from Single to Copy All.	Run the CIAI Wizard with an input file that specifies this change.
For columns already configured, change index strategy from Copy All to Single.	<p>Run the CIAI Wizard using an input file that specifies the desired columns. For each column, specify Operation indicates Off. This deactivates the CIAI column and CIAI indexes.</p> <p>Then run the CIAI Wizard on the same base columns with method indicates Database and index Strategy indicates Single. This activates the index on the CIAI columns.</p>

Column and Index Naming Convention

The CIAI Wizard uses the same naming conventions as the EIM Wizard when creating CIAI column names and index names. The naming conventions used by the Wizard are fixed and cannot be overridden.

Column Names

The Wizard defines CIAI column names by appending `_CI` to the parent column name. For example, if the parent column is `LAST_NAME`, the CIAI column would be `LAST_NAME_CI`.

Index Names

The Wizard defines CIAI index names by adding a string to the base table name:

base table Name_C#

where:

is an integer incremented starting at 1 as needed to create a unique name.

An example of a CIAI index created on table `S_CON_ADDR` is `S_CON_ADDR_C1`.

Truncating Names

The default length for column names and index names is 30 characters. You can limit the length of names to 18 characters (to conform to the requirements of the DB2 database), prior to running the CIAI, by selecting the "Limit schema object names to 18 characters" option in Tools > View > Options > Database.

If CIAI column names or index names exceed the maximum length, the Wizard truncates the column base name or the table base name (for indexes) using the following strategy:

- Deletes underscores one at a time, beginning with the left-most
- Deletes vowels one at a time, beginning with the right-most
- Deletes characters (letters, numbers, and so on) one at a time, beginning with the right-most.

The Wizard does not truncate prefixes or postfixes.

Column and Index Name Uniqueness

After truncating a column or index name, the name may not be unique. When this occurs, the Wizard modifies the truncated column or index name by truncating the right most character in the base column name or base table name. The Wizard replaces the truncated character with an integer, starting at 1.

If this modification does not create a unique name, the Wizard increments the integer. If the integer becomes two digits or larger, the Wizard truncates the name to make room for the digits. This process maintains the overall string length.

Error Reporting

The CIAI Wizard reports errors in a pane in the Wizard. The error listing provides enough information so that you can identify the column and the cause of the problem.

You then have two options:

- You can correct the errors and rerun the Wizard.
- You can ignore the errors.

When the Wizard configures columns, it skips all columns that have errors.

When the Wizard reports errors, you can export them to a text file for reference. Errors usually fall into one of the following categories:

- **Input file syntax errors.** These include punctuation and use of improper configuration options.
- **Table and column eligibility problems.** These errors occur when you choose tables and columns of types that the Wizard does not support.
- **Project not locked.** You must lock the tables you want to configure before running the Wizard. The Wizard displays the list of projects that must be locked.

Example of a Siebel Development Upgrade Flow

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

This topic presents the flow of steps in part of a typical development environment upgrade. The steps are extracted from an actual driver file. To perform an upgrade, the Upgrade Wizard reads the steps in a driver file and performs the commands the steps contain.

The driver file type used in this example is as follows:

- Upgrade: Siebel Industry Application (SIA) 7.7 to Siebel SIA 7.8
- Environment: development
- Upgrade mode: upgrep
- Database: Oracle
- Multilingual: No

Table 12 lists the steps in the driver file. Key steps are in bold.

The Script or Input File column in the table lists the SQL file or input file that is executed in each step. The Comment column provides a brief explanation of what the SQL file or input file does.

The SQL files used for an upgrade and the contents of the SQL files vary depending on the upgrade mode and database.

Table 12. Example of Steps in a Development Environment Upgrep

Step	Script or Input File	Comment
Determine collation sequence of database.	Not applicable	Determines database sort order.
Verify repository name	rename_existing_repositories.sql	Renames Siebel Repository to Prior Customer Repository.
Remove interface tables	dropif-db.sql	Removes all EIM tables.
Remove database triggers	trigdrop-db.sql	Removes all dynamically created triggers.
Remove database-level functions and procedures	drop_db_func_proc.sql	Removes the exchange rate function exrate.
Prepare for table creation	pret.sql	Removes specified tables. Performs DDL operations such as adding columns to tables. Performs DML operations such as revising date formats.
Create temporary tables for SIA	<ul style="list-style-type: none"> ■ ddlimp utility ■ ddl_temp_sia.ctf as input 	The input file specifies the structure of the tables to be created or updated. These tables are used to perform data migration and other DML changes.

Table 12. Example of Steps in a Development Environment Upgrep

Step	Script or Input File	Comment
Prepare for table creation for SIA	pret_sia.sql	Drops specified tables. Performs DDL operations such as adding columns to tables. Performs DML operations such as revising date formats.
Create and update tables	<ul style="list-style-type: none"> ■ ddlimp utility ■ ddl.ctl as input 	The ddl.ctl file specifies the structure of tables to be created or updated.
Create temporary tables for stored procedures	<ul style="list-style-type: none"> ■ ddlimp utility ■ ddlsptbl.ctl as input 	The input file specifies the structure of temporary tables to be created or updated.
Household data migration for FINS	Household_Mig_Fins.sql	Creates and populates specified temporary tables. Then migrates data to them and performs DML operations. Migrates data back to primary tables. Drops temporary tables.
Prepare for index creation	preschem.sql	Performs DML operations. Moves data between tables. Changes data in existing fields based on specified conditions.
Prepare for index creation for SIA	preschem_sia.sql	Same as preschem.sql.
Create indexes	<ul style="list-style-type: none"> ■ ddlimp utility ■ ddl.ctl as input 	The input file specifies the structure of indexes to be created.
Delete old license key	delappkey.sql	Deletes the Siebel license key from S_APP_KEY.

Table 12. Example of Steps in a Development Environment Upgprep

Step	Script or Input File	Comment
Import seed data	<ul style="list-style-type: none"> ■ dataimp utility ■ seedupg0.inp as input ■ seedupg1.inp as input ■ seedupg_locale.inp as input 	<p>Prior to importing seed data, dataimp deletes existing seed data.</p> <p>The seedupg* files contain filters that dataimp uses to prevent deleting seed data you have modified or seed data meeting specified criteria.</p> <p>Unmodified seed data has a last update date (LAST_UPD) of 1980-01-01. Dataimp does not delete records where LAST_UPD is later than this date.</p>
Upgrade data after seed data import	upg_data_afterseed.sql	For customers who have not converted to UTC time, sets the UTC value in S_SYS_PREF to False. For customers who have converted to UTC time, the script takes no action.
Upgrade data after seed data import SIA	upg_data_afterseed_sia.sql	Empty.
Set system preference for codepage for DB	set_codepage.sql	Sets the database codepage in S_SYS_PREF.
Set system preference for Unicode codepage for DB	set_unicode.sql	Sets the Unicode codepage to UTF-8 in S_SYS_PREF
Update version component information	upd_upgcomp.sql	<p>Updates S_UPG_COMP with the product release level. Also updates location of srf file.</p> <p>S_UPG_COMP stores version information for application executables.</p>
Run Oracle-specific DDL commands	ddlora.sql	Creates Oracle-specific DDL information, such as default storage parameters for docking objects, repository objects, and seed objects.

Table 12. Example of Steps in a Development Environment Upgrop

Step	Script or Input File	Comment
Import common ancestor repository	<ul style="list-style-type: none"> ■ repimexp utility ■ Standard Siebel Repository as input 	Imports the Standard Siebel Repository into S_REPOSITORY. For example, if you are upgrading from Siebel 7.5.3, this command imports the standard Siebel 7.5.3 repository.
Rename EIM columns/indexes	rmv_anc_eim_proc_col_ind.sql	<p>Removes EIM processing columns and indexes from the Prior Customer Repository and the common ancestor repository (Standard Siebel Repository). This prevents the repository merge from preserving EIM columns incorrectly.</p> <p>The merge will preserve only those Siebel EIM columns shipped with the new Release.</p>
Update Siebel Database version	<ul style="list-style-type: none"> ■ update_ver.sql ■ seeduver.sql 	The update_ver.sql script creates a temporary table, S_APP_VER_TEMP, which contains new version information for the database schema. The seeduver.sql script updates S_APP_VER with this information.
Import new Siebel Repository	<ul style="list-style-type: none"> ■ repimexp utility ■ Imports New Siebel Repository 	<p>Imports the New Siebel Repository into S_REPOSITORY.</p> <p>Revises schema version information in S_APP_VER.</p>
Import New Customer Repository	<ul style="list-style-type: none"> ■ repimexp utility ■ Imports New Customer Repository 	<p>Imports the New Customer Repository into S_REPOSITORY.</p> <p>Revises schema version information in S_APP_VER.</p>
Restore database version	restore_ver.sql	Uses S_APP_VER_TEMP to update the schema version information in S_APP_VER. Drops S_APP_VER_TEMP.

Table 12. Example of Steps in a Development Environment Upgrade

Step	Script or Input File	Comment
Upgrade repository data SIA	repos_upgrade_sia.sql	Empty.
Upgrade repository data	repos_upgrade.sql	Makes specific repository-related changes to repository records and to other tables.
Set repository workflow domains to MLOV	set_multilingual.sql	Empty.

Table 12. Example of Steps in a Development Environment Upgrep

Step	Script or Input File	Comment
Install SQL packages	<ul style="list-style-type: none"> ■ seeduver.sql ■ ifstrg.sql ■ ifindxstrg.sql ■ pkgseq.sql ■ pkgldel.sql ■ trgreset.sql ■ ddlseq.sql ■ pkgvis.sql ■ delete_dock_rules.sql 	<ul style="list-style-type: none"> ■ seeduver.sql. Verifies that versions are set correctly in S_APP_VER. ■ ifstrg.sql. Sets storage parameters for EIM tables. ■ ifindxstrg.sql. Sets storage parameters for EIM table indexes. ■ pkgseq.sql. Adds a suffix to row IDs in the S_SEQUENCE table. Ensures that row IDs are unique. ■ pkgldel.sql. Defines s_txn_log_del_proc. Procedure periodically deletes transactions from S_DOCK_TXN_LOG. Also deletes rows from S_DOCK_TXN SET. Prevents need for large rollback segment. ■ trgreset.sql. Ensures that denormalized rows in S_TERR have correct values. ■ ddlseq.sql. Sets sequence numbers for specified tables. ■ pkgvis.sql. Creates function that modifies how Oracle optimizer does visibility check. ■ delete_dock_rules.sql. Deletes Prior Customer Repository routing rules from S_DOC_VIS_RULE that meet specified conditions. Attempts to preserve rules added using Docking Wizard.

Table 12. Example of Steps in a Development Environment Upgreq

Step	Script or Input File	Comment
Create database-level functions and procedures	db_func_proc.sql	Creates or replaces the currency exchange rate function: exrate.
Upgrade ISS data	upg_iss.sql	Updates data stored in tables that support Oracle's Siebel Product Configurator.
Set primary children in data tables	<ul style="list-style-type: none"> ■ gen_primary1.sql ■ gen_primary2.sql ■ gen_primary3.sql ■ gen_primary4.sql 	<ul style="list-style-type: none"> ■ gen_primary1.sql1. Sets primary child for S_DOC_QUOTE ■ gen_primary2.sql. Empty. ■ gen_primary3.sql. Sets primary child for S_LOY_PROMO ■ gen_primary4.sql. Empty

Related Topics

["About the Siebel Database Configuration Utilities" on page 62](#)

["About the Siebel Upgrade Wizard and Driver Files" on page 68](#)

About the Siebel Repository Merge

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Repositories are located in tables in the Siebel Database. These tables store the definitions of the objects used to build Siebel Applications. These tables also store Siebel schema definitions. When you display objects in the Siebel Tools Object List Explorer, you are displaying records from these tables.

The repository tables contain two types of records:

- Object definitions used to create Siebel applications, such as business components, applets, controls, and the relationships between them.
- Definitions of the tables in the Siebel Database (metadata). These records define the logical schema of the Siebel Database. Later in the upgrade process, you will synchronize the physical schema of the Siebel Database with the logical schema as defined by these records.

Repository records include a repository ID that identifies the repository to which the record belongs. The repository ID forms part of the user index for records and allows several repositories to be stored in the same set of tables.

The Siebel Repository is the deployed, active Tools repository. You use this repository to customize applications and create SRF files for deployment.

The development upprep adds three additional repositories to the Siebel Database, as shown in [Table 13](#). These repositories are required for the repository merge.

Table 13. Development Database Repositories

Repository Name	Added During Upgrade?	Siebel Tools Name	Description
Prior Customer Repository	No	Prior Customized	This is your current Siebel Repository. It contains any changes you have made. You renamed it Prior Customer Repository before doing the initial database upgrade.
Prior V7.x Siebel Repository	Yes	Prior Standard	The standard Siebel Repository for your installed release (the one you are upgrading from). Is called the common ancestor repository in the upgrade SQL scripts.
New Siebel Repository	Yes	New Standard	The Siebel Repository for the release to which you are upgrading. This repository defines the new release.
New Customer Repository	Yes	New Customized	A second copy of the New Siebel Repository. Your customizations from the Prior Customer Repository are merged into this repository. After the upgrade, this becomes the Siebel Repository.

What Happens During a Repository Merge?

The repository merge process identifies differences between the repository in your old release (Prior Customer Repository) and the repository in the new release (New Siebel Repository). The merge process then merges these differences into the New Customer Repository. The merge process searches for the following types of objects in the Prior Customer Repository: customer-created, customer-deleted, and customer-modified.

Customer-Created Objects

Customer-Created objects are top-level objects, such as screens, applets, and business components, that you have created in the Prior Customer Repository. If an object is present in the Prior Customer Repository but not the Prior Siebel Repository, it is customer-created. The merge process copies customer-created objects intact to the New Customer Repository.

Customer-Deleted Objects

Customer-Deleted objects are top-level objects you have deleted in the Prior Customer Repository. If an object is absent in the Prior Customer Repository but present in the Prior Siebel Repository, it is customer-deleted.

If you delete a top-level object in the Prior Customer Repository and it is present in the New Customer Repository, the merge process does not delete the object from the New Customer Repository. After the merge, you must review these objects and remove them as desired.

If you delete a top-level object from the Prior Customer Repository, and it is obsolete (inactive) in the New Customer Repository, the object will be present and inactive in the New Customer Repository.

Customer-Modified Objects

A customer-modified object has the following characteristics:

- It is a top-level object, such as screen, applet, or business component.
- The object exists in the Prior Siebel Repository and the Prior Customer Repository. (The object is not customer-created or customer-deleted)
- The properties of the object in the Prior Customer Repository and Prior Siebel Repository are not the same. (You have modified the object).

If the object properties are also different between the Prior Siebel Repository and New Siebel Repository, the object has changed in the new release, and a conflict exists. The merge process logs the conflict and resolves it. After the merge, you must review how these conflicts were resolved and change the resolutions as desired.

For customer-modified objects where no conflict exists, the merge process copies the modifications to the object to the New Customer Repository.

If you modify a top-level object by deleting any of its child objects, the merge process does not delete these child objects in the New Customer Repository. After the merge, you must review child objects and remove them as desired.

Some of the child objects of Applet are an exception. If you delete the following child objects of Applet from the Prior Customer Repository, the merge process deletes these objects from the applet in the New Customer Repository:

- Control
- List Column
- Page Tab
- Chart

How the Siebel Database Upgrade Works ■

- Applet Web Template Item
- View Web Template Item

For example, if you delete a button from Applet-A in the Prior Customer Repository. The merge process deletes this button from Applet-A in the New Customer Repository.

Table 14 shows how a regular merge handles both customer-modified and customer-created objects. The columns list the status of a repository object:

- **Standard.** The object appears in the Prior Standard Repository, in the New Siebel Repository, and is not customer-modified.
- **Deleted.** You have deleted the object from the Prior Customer Repository (customer-deleted).
- **Customized.** You have modified the object in the Prior Customer Repository (customer-modified).
- **Revised.** The object has changed in the new release (New Siebel Repository).
- **New.** You have created the object in the Prior Customer Repository (customer-created), or the object is new in the new release (New Siebel Repository).
- **Inactive.** The object is present in the New Siebel Repository and New Customer Repository but is inactive and not used in the new release. The object is obsolete.

The first three columns list the status of the object in the three repositories that the merge process compares during the merge. The last column, Merged New Customer Repository, lists the status of all top-level repository object types after the repository merge is complete and the postmerge utilities have run.

Table 14. How the Merge Handles Repository Objects

Prior Standard Repository (PSR)	Prior Customer Repository (PCR)	New Siebel Repository (NSR)	Merged New Customer Repository
Standard	Standard	Standard	Standard
Standard	Standard	Revised	Revised
Standard	Standard	Standard/ Inactive	Standard/Inactive
Standard	Customized	Customized	Customized
Standard	Customized	Customized/ Inactive	Customized/Inactive
Standard	Customized	Revised (conflict)	Revised (conflict)
Not applicable	New	New	New
Not applicable	Not applicable	New	New
Standard	Deleted	Standard	Standard

Table 14. How the Merge Handles Repository Objects

Prior Standard Repository (PSR)	Prior Customer Repository (PCR)	New Siebel Repository (NSR)	Merged New Customer Repository
Standard	Deleted		
Standard	Deleted		

Upgrade Ancestor Property

You can link repository objects together so that one object inherits the upgrade behavior of another. You do this by specifying an upgrade ancestor for an object. You can specify upgrade ancestors for the following object types:

- Applets
- Business Components
- Integration Objects
- Reports

For example, you create Applet-B by copying Oracle standard Applet-A. In Applet-B you specify Applet-A as the upgrade ancestor. In the New Siebel Repository, Applet-A has a new button. Because Applet-B is a descendant of Applet-A, the merge process adds the new button to both Applet-A and Applet-B.

If you had not specified Applet-A as an upgrade ancestor of Applet-B, Applet-B would not have received the new button.

Postmerge Utilities

The postmerge utilities run after the repository merge completes. The utilities make changes to objects in the New Customer Repository. The purpose of the postmerge utilities is to reduce the configuration changes required as part of reviewing applications and UI after the merge.

Incorporate Custom Layout

If you are upgrading from Siebel 7.x or later, you can choose Incorporate Custom Layout (ICL) when performing a repository merge. ICL is intended to reduce postmerge UI reconfiguration workload for customer-modified UI objects.

An ICL merge is a regular merge that contains two extra steps. The first step occurs at the end of the merge and applies special handling to screens, views, and applets. The second step is part of the postmerge utilities and provides specialized web template files similar to the release you are upgrading from.

Related Topics

[“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 102](#)

[“About Inheriting Upgrade Behavior in a Siebel Upgrade” on page 96](#)

[“About the Siebel Postmerge Utilities” on page 98](#)

[“Reviewing Siebel Repository Object Property Conflicts” on page 258](#)

About Inheriting Upgrade Behavior in a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

You can link objects together so that one object inherits the upgrade behavior of another. You do this by specifying an upgrade ancestor for an object.

Use standard objects as upgrade ancestors. A **standard object** is an uncustomized repository object provided by Oracle.

A common customization strategy is to create new objects by making a copy of a standard object and then modifying the copy, called the **descendant**.

For descendants that are the following object types, you can specify an upgrade ancestor:

- Applets
- Business Components
- Integration Objects
- Reports

You specify the upgrade ancestor in the descendant's Upgrade Ancestor field in Siebel Tools. During the repository merge, the descendant is upgraded in the same way as the upgrade ancestor. (Upgrade ancestors of applets are ignored in an ICL merge.)

For example, you create Applet-B by copying standard Applet-A. In Applet-B you specify Applet-A as the upgrade ancestor. In the New Siebel Repository, Applet-A has a new button.

After a regular merge, both Applet-A and Applet-B will have the new button.

After an ICL merge, Applet-B will have the new button, but Applet-A will not:

- During the regular merge, Applet-B receives the new button because it is a descendant of Applet-A.
- During the ICL merge portion, the web template configuration of Applet-A is copied from the Prior Customer Repository to the New Customer Repository. This web template configuration does not contain the new button. This removes the new button from Applet-A.
- Since the ICL merge portion ignores the Upgrade Ancestor property, this change is not propagated to Applet-B. Applet-B retains the new button.

Limitations on the Upgrade Ancestor Property

The Upgrade Ancestor property is considered only during repository merges as part of application upgrades under these conditions:

- If an upgrade ancestor is inactive in the New Siebel Repository, it is obsolete, and its upgrade behavior is not propagated to descendants.
If an ancestor object is obsolete in the New Siebel Repository, descendants are not also obsolete.
If an upgrade ancestor is not present in the New Siebel Repository, error messages display during the repository merge and are written to the merge log file. These errors are acceptable and do not mean the merge has failed.
- The Upgrade Ancestor property is not considered during repository imports. However, imported objects can specify an upgrade ancestor. When the next application upgrade is done, the Upgrade Ancestor property is taken into account.
- The setting of the Upgrade Ancestor property is not considered when applying application patches. If the upgrade ancestor is modified by the patch, descendants are not modified.
- Use caution when specifying upgrade ancestors. For regular merges, setting the Upgrade Ancestor property on applets propagates merge problems from standard-object applets to descendant applets.
- Specifying an upgrade ancestor for objects slows the repository merge.

Upgrade Ancestor Picklist

When you click in the Upgrade Ancestor field, a picklist displays. The following criteria are used to populate the picklist:

Applets

- Table is the same as the current applet buscomp
- Class is the same as the current applet class
- Upgrade Ancestor is null
- Applet is a standard object

Reports

- Buscomp is the same as the current report buscomp
- Class is the same as the current report class
- Upgrade Ancestor is null
- Report is a standard object

Business Components

- Bus Comp is the same as the current business component
- Class is the same as the current business component

- Upgrade Ancestor is null
- Business component is a standard object

Integration Objects

- Base Object Type is the same as the current Base Object Type
- Business Object is the same as the current business object
- Upgrade Ancestor is null
- Integration object is a standard object

Propagating Changes to Objects After the Merge

If you do not select an upgrade ancestor for an object, changes to the upgrade ancestor are not propagated to the descendant during the repository merge.

You can manually propagate changes to descendants after the merge by using the Siebel Tools object comparison and synchronization features. These features enable you to compare any two objects and propagate differences to one or both of the objects. For more information, see *Configuring Siebel Business Applications*.

Related Topics

[“About the Siebel Repository Merge” on page 91](#)

[“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 102](#)

About the Siebel Postmerge Utilities

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

You run the postmerge utilities after the repository merge completes.

The postmerge utilities make revisions to objects in the New Customer Repository. The utilities are intended primarily to identify UI configuration problems for upgrades prior to Siebel 7.7. (At Siebel 7.7, there were significant changes to UI navigation.) The utilities are also useful for identifying problems with UI customizations for upgrades after Siebel 7.7.

The postmerge utilities are implemented as a framework. Each utility in the framework is a business service and is listed in `reputi l i ty. xml` :

Windows: `SIEBEL_TOOLS_INSTALL_DIR\reppatch\reputi l i ty. xml`

Where

`SIEBEL_TOOLS_INSTALL_DIR` is the directory where Siebel Tools is installed.

The postmerge utilities are as follows:

- **CSSGridRepPatch**. Converts customized applet controls to grid-based layout.
- **CSSUI NavUpgradeReposPrep**. Converts customized screens to the Siebel 7.7 UI navigation scheme.
- **CSSMVGUpgradePatch77**. Enables the shuttle feature in customized MVGs.
- **CSSUI UpgradeReports**. Lists unresolved problems in converting customized screens and views to the Siebel 7.7 UI navigation scheme.
- **CSSWebTemplatePatch**. For ICL upgrades, the utility copies Web templates into Siebel Tools that support the release you are upgrading from.
- **CSSWFRepPatch**. For workflows, this utility changes step references from row-id references to name references for child objects of steps.

CSSGridRepPatch

As of Siebel 7.7, the layout of form applets is grid-based, rather than flow-based. Applet Form Web templates have several new properties: Grid Property, Row Span, and Col Span. In addition, item identifier syntax changes to xxyyy, where xx is the grid row and yyy is the grid column where the control is located.

Regular merge. The utility identifies both customer-created and customer-modified applets and adjusts the properties of controls in them as follows:

- If you have modified the location of an existing control, the utility restores the control to its original location. If you have modified other properties of the control, these changes are preserved. You must move the control and label to the desired location.
- If you have added a new control, it does not display after the merge. To display the control, the utility assigns an item identifier that places the control in the lower left portion of the applet. The utility creates a Grid Property, Row Span, and Col Span property for the item. The utility also creates a Label control. You must move the new control and label to the desired location.
- The utility does not change property settings for controls that are unmodified.

ICL merge. (Applies only to applets where Upgrade Behavior is not set to Admin.) For upgrades from releases prior to Siebel 7.7, form applets copied-in from the Prior Customer Repository remain flow-based and display normally. The utility does not convert these applets to grid-based. New and modified controls display in the correct locations. For upgrades from Siebel 7.7 or later, the utility makes no changes.

The utility generates output to the `reputility.log` section called Invalid Applet Web Template Item Mapping Cleanup.

CSSUI NavUpgradeReposPrep

At Siebel 7.7, a declarative model for associating views with screens was introduced. The relationship between views and screens for all levels of navigation must be explicitly declared. Several new object properties for screens and views are introduced to support this.

How the Siebel Database Upgrade Works ■

The new navigation scheme is applied to all screens and views for both regular and ICL merges. The utility runs after the repository merge in all upgrades. However, it is intended primarily for upgrades from releases prior to Siebel 7.7.

After a regular or ICL merge, screens and views display as follows:

- Regular merge:
 - Unmodified screens display normally.
 - The views in customer-created screens do not display.
 - Views added to existing screens do not display.
- ICL Merge:
 - Unmodified screens with Upgrade Behavior property set to Admin display normally.
 - Views in unmodified screens with Upgrade Behavior set to Preserve or NULL do not display.
 - Views in customer-created screens do not display.
 - Views added to existing screens do not display, regardless of Upgrade Behavior setting.

The utility scans all screen view definitions looking for orphaned views. It groups orphaned views under existing Categories. If no Category exists, the utility creates one. This causes the orphaned views to display.

For a regular merge, you must review customer-created and customer-modified screens and views to verify that views are correctly associated with screens.

For an ICL merge, you must also review these screens and views. In addition, you must review all screens and views where Upgrade Behavior is set to Preserve or NULL.

The utility generates output to the `reputility.log` section called User Interface Navigation Upgrade.

CSSMVGUpgradePatch77

As of Siebel 7.7, MVG applets with an M:M relationship to the underlying business component are configured as shuttle applets by default. The utility scans these MVG applets in the New Customer Repository and reconfigures them to display as shuttle applets:

- **Regular merge.** After the merge, customer-created and customer-modified MVGs are not shuttle-enabled. The utility reconfigures these MVGs so they display as shuttle applets.
- **ICL merge.** After an ICL merge, the utility reconfigures MVGs as follows:
 - Customer-created and customer-modified MVGs are not shuttle-enabled. The utility reconfigures these MVGs so they display as shuttle applets.
 - Unmodified MVGs that have Upgrade Behavior set to Preserve or NULL are copied to the New Customer Repository during the merge and replace MVGs that are shuttle-enabled. The copied-in MVGs may not be configured as shuttle applets. The utility reconfigures the copied-in MVGs so they display as shuttle applets.
 - Unmodified MVGs that have Upgrade Behavior set to Admin are not deactivated in the New Customer Repository and so display as shuttle applets.

The utility generates output to the `reputility.log` section called `Multi Value Group Shuttle Applet Upgrade`.

CSSUI UpgradeReports

This utility makes no changes to the New Customer Repository. Instead, it scans UI objects in the repository and lists problems that could not be resolved by the `CSSUINavUpgradeReposPrep` utility.

The `CSSUIUpgradeReports` utility writes the report to `reputility.log`. You must manually correct problems listed in the report. The report is located in the `POST MERGE USER INTERFACE REPORTING UTILITY` section of the log.

The report has the following sections:

- Issue 1: Rich Text Control (RTC) that needs to have User Properties Reconfigured
- Issue 2: New Aggregate Category Records that should be renamed
- Issue 3: Views that need an applet in View Web Template Item Id 1
- Issue 4: Chart Views Needing Migration to Aggregate Type
- Issue 5: Explorer Views Needing Migration to Aggregate Type
- Issue 6: Categories where parent applets are missing drilldowns to a Detail View

CSSWebTemplatePatch

This utility runs after an ICL merge. It does not run after a regular merge. The utility performs the following steps in the installation directory of the Siebel Tools that you used to perform the merge:

- The utility moves the Web template files for the new release from `webtempl` to `\temp\webtempl`.
- The utility copies Web template files from a subdirectory of `reppatch\web_templates` to `\webtempl`.

These Web template files support the UI of the release you are upgrading from. They are very similar but not identical to the Web template files included in that release.

For example, if you selected the 7.5.3 and “Label on Top” ICL options, the utility copies Web template files from the `753` and `TopLabel` subdirectories of `reppatch`.

The utility generates output to the `reputility.log` section called `Web Template PCL Patch Rule`.

CSSWFRepPatch

This utility changes workflow step references from row-id references to name references for child objects of steps. This completes the process of migrating workflows to the repository at Siebel 7.7. This utility primarily affects upgrades from releases prior to Siebel 7.7.

This utility does not write to the `reputility.log`.

How the Postmerge Utilities Work the Upgrade Behavior Property

The postmerge utilities ignore the Upgrade Behavior property. They make changes to UI objects based on the object's characteristics rather than Upgrade Behavior setting.

Related Topics

["About the Siebel Repository Merge" on page 91](#)

["About Inheriting Upgrade Behavior in a Siebel Upgrade" on page 96](#)

["About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option" on page 102](#)

About the Siebel Incorporate Custom Layout (ICL) Upgrade Option

Upgrades from: Siebel 7.x using ICL.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

In each new release, there are typically several types of changes to existing UI objects:

- Screen layouts are revised
- View layouts are revised
- Controls are added or removed in applets
- Controls are moved in applets

If you have modified existing screens, views, and applets, the changes in the new release can require significant UI layout reconfiguration after the repository merge. The purpose of an ICL merge is to reduce postmerge UI layout reconfiguration.

Overview of Incorporate Custom Layout

If you are upgrading from Siebel 7.x or later, you can choose Incorporate Custom Layout (ICL) when performing a repository merge. An ICL merge handles customer-modified screens, views, and applets differently than other repository objects.

A regular merge identifies changes you have made to objects in the Prior Customer Repository (customer-modified objects) and merges those changes into the counterpart objects in the New Customer Repository. For example, if you added a control to Applet-A in the Prior Customer Repository, a regular merge adds the control to Applet-A in the New Customer Repository.

An ICL merge does not merge changes into counterpart objects in the New Customer Repository. Instead, an ICL merge process replaces the configuration of screens, applets, and views in the New Customer Repository with those from the Prior Customer Repository.

For example, you have added a control to Applet-A in the Prior Customer Repository. The merge process deactivates the Web template configuration for Applet-A in the New Customer Repository and replaces it with the Web template configuration for Applet-A from the Prior Customer Repository. The Web template configuration from the Prior Customer Repository contains the new control along with the correct layout of the control.

In addition, after the merge, the postmerge utilities copy-in new Web template files that are similar to the release you are upgrading from. Web template files provide page containers that control screen, view, and applet layouts. The copied-in Web template files further ensure that the UI in the new release has layouts similar to the UI in the old release.

An ICL merge adds two additional steps to the regular merge process. These two steps are called *ICL handling*:

- At the end of the merge process, Screen, View, and Applet object types along with specific child objects, are copied to the New Customer Repository from the Prior Customer Repository. They replace their counterpart objects in the New Customer Repository.
- When the postmerge utilities run, the Web template files for the new release are copied out and saved. New Web template files similar to those of the release you are upgrading from are copied-in.

Customer-created screens, views, and applets do not receive ICL handling. These UI objects are copied to the New Customer Repository intact. This is true even if a customer-created applet specifies an ICL object applet as an upgrade ancestor.

Who Should Use ICL?

Consider using ICL if it can significantly reduce your postmerge UI cleanup workload. ICL merges are intended for customer-modified UI objects. A *customer-modified UI object* is a standard screen, view, or applet you have modified and saved without changing its name. Screens, views, and applets that you create by copying them from a standard object and renaming them are *customer-created UI objects*.

Use the following factors to assess whether to use ICL:

- ICL can be used only for upgrades from Siebel 7.x onward.
- If your UI customizations consist mostly of customer-modified objects, then ICL is a good choice. ICL is intended to preserve the layout and contents of customer-modified screens, views, and applets.
- If your customizations consist primarily of customer-created UI objects that specify an upgrade ancestor, ICL is not recommended. This is because ICL ignores the Upgrade Ancestor property. For UI objects, if the configuration of an upgrade ancestor changes in the new release, the changes are not propagated to the descendant.
- In each release, ICL handling cannot be applied to screens, views, and applets that have Upgrade Behavior set to Admin in Siebel Tools. If you have large numbers of customer-modified screens, views and applets that are excluded from ICL handling, ICL may not be a good choice.

- You can select ICL for only every other upgrade. Choose ICL for upgrades where the most customizations occur.

At the upgrade after an ICL upgrade, you must bring UI layouts forward to those of the installed release before performing the repository merge.

UI Objects That ICL Affects

To receive ICL handling, a UI object must meet all the following criteria. Objects that meet all the criteria are called *ICL objects*:

The object must be one of the following top-level object types. For each top-level object type, the child objects that ICL affects are also listed:

- Screen
 - Screen View and all child objects
- View
 - View Web Template and all child objects
- Applet
 - Applet Web Template and all child objects
 - Drilldown and all child objects
- Screens, Views, and Applets, must be a standard object or a customer-modified object. (Customer-Created screens, views, and applets receive regular merge handling.)
- The object must be *preservable*. A screen, view, or applet is preservable if its Upgrade Behavior property is not set to Admin.

In summary, both customer-modified and standard unmodified screens views, and applets that are preservable receive ICL handling.

[Table 15](#) compares a regular merge and an ICL merge. The columns list the status of a repository object:

- **Standard.** The object appears in the Prior Standard Repository, in the New Siebel Repository, and is not customer-modified.
- **Deleted.** You have deleted the object from the Prior Customer Repository (customer-deleted).
- **Customized.** You have modified the object in the Prior Customer Repository (customer-modified).
- **Revised.** The object has changed in the new release (New Siebel Repository).
- **New.** You have created the object in the Prior Customer Repository (customer-created), or the object is new in the new release (New Siebel Repository).
- **Inactive.** The object is present in the New Siebel Repository and New Customer Repository but is inactive and not used in the new release. The object is obsolete.

The first three columns list the status of the object in the three repositories that the merge process compares during the merge. The last two columns list the status of the object after the merge is complete and the postmerge utilities have run.

The Merged New Customer Repository column lists the status of all repository object types after the repository merge.

The ICL Merged New Customer Repository column lists the status of screen, view, and applet ICL objects after an ICL merge.

Exceptions listed in [Table 15](#) are described in a section following the table.

Table 15. How the Merge Handles Repository Objects

Prior Standard Repository (PSR)	Prior Customer Repository (PCR)	New Siebel Repository (NSR)	Merged New Customer Repository	ICL Merged New Customer Repository
Standard	Standard	Standard	Standard	Standard
Standard	Standard	Revised	Revised	Standard (from PCR) See 1 in Table Exceptions
Standard	Standard	Standard/ Inactive	Standard/ Inactive	Standard/ Inactive
Standard	Customized	Standard	Customized	Customized
Standard	Customized	Standard/ Inactive	Customized/ Inactive	Customized/ Inactive
Standard	Customized	Revised	Revised (conflict) See 2 in Table Exceptions	Customized
Not applicable	New	Not applicable	New	New
Not applicable	Not applicable	New	New	New
Standard	Deleted	Standard	Standard	Standard
Standard	Deleted	Standard/ Inactive	Standard/ Inactive	Standard/ Inactive
Standard	Deleted	Revised	Revised	Revised

Table Exceptions

Exceptions listed in [Table 15](#) are as follows:

- In an ICL merge, UI navigation changes at Siebel 7.7 are implemented in screens copied-in from the Prior Customer Repository.
- A conflict exists when an object is different in all three repositories. The merge process typically resolves the conflict in favor of the New Siebel Repository. After the merge, you can review conflicts and change the resolution.

Some UI Objects Are Not Eligible for ICL Handling

Certain screens, views, and applets that are required to administer applications and support the standard operation of the application are never eligible for ICL handling. These include UI objects for Server Administration and Remote Administration.

In addition, if the screens, views, and applets in an application have changed significantly in a new release, they are also excluded from ICL handling. ICL handling for these UI objects is not practical, because their layout and functionality have been greatly revised.

Application areas excluded from ICL handling in recent releases are as follows:

- **Siebel 7.7:** Applets and views for Siebel Employee Relationship Management (ERM) and Siebel Marketing.
- **Siebel 7.8:** Applets and views in the Quotes and Orders screens and related customer management applets and views.
- **Siebel 8.0:** Most application home page screens.

Upgrade Behavior Property

As of Siebel 7.7, an object property called Upgrade Behavior determines whether a UI object is preservable:

- If this property is set to Preserve then the UI object is eligible for ICL handling.
- If this property is set to Non-Preservable, then the UI object is not eligible for ICL handling for upgrades to the currently shipping release. At a future release the setting may be changed to Preserve.
- If this property is set to ADMIN, then the UI object is never eligible for ICL handling.

The Upgrade Behavior property is defined on Screen, View, and Applet.

At each release, Oracle sets the value of Upgrade Behavior for UI objects. Do not change these values.

ICL Upgrade Path

As of Siebel CRM 8.0, an object property called ICL Upgrade Path controls the releases for which Upgrade Behavior indicates Preserve is effective. ICL Upgrade Path restricts eligibility for ICL handling to specific releases.

You select ICL Upgrade Path from an LOV. Values include Pre-7.5.3, 7.5.3, 7.7, and 8.0. Like the Upgrade Behavior property, ICL Upgrade Path is defined on Screen, View, and Applet.

If Upgrade Behavior is set to Preserve, and ICL Upgrade Path is set to Release X, these settings have the following effects:

- For upgrades from releases prior to Siebel X, the object is not eligible for ICL handling. Customizations will not be preserved.
- For upgrades from Siebel X and later, the Preserve status is effective, and the object is eligible for ICL handling. Customizations will be preserved.

For example:

- At Siebel 7.7, Applet A was redesigned.
- The changes were so great that preserving customizations to Applet A when upgrading from a previous release was not practical. So Applet A must be made ineligible for ICL handling for upgrades from releases prior to Siebel 7.7.
- There were no significant changes to Applet A at Siebel 7.8. or Siebel 8.0. So Applet A is eligible for ICL handling for upgrades from Siebel 7.7 and later.
- The following settings create this upgrade behavior for Applet A: Upgrade Behavior indicates Preserve, ICL Upgrade Path indicates 7.7.

Note that there are two ways to make an object ineligible for ICL handling in the currently shipping release:

- Set the Upgrade Behavior property to Non-Preservable.
- Set the Upgrade Behavior property to Preserve, and set ICL Upgrade Path to the currently shipping release.

At each release, Oracle sets the value of ICL Upgrade Path for UI objects. Do not change these values.

How ICL Affects the Overall UI

In general, an ICL merge preserves the UI layouts of the Siebel Release you are upgrading from but also applies any significant UI navigation changes in the new Siebel Release.

Screens

The association of views with screens is preserved. For upgrades prior to Siebel 7.7, a new navigation scheme is introduced. When reviewing the UI after the merge, verify that all views can be displayed.

Views

- The association of applets with views is preserved.
- The layout of views is preserved.
- System pages such as Help, About View, About Record, and Technical Support are not preserved.

Applets

The Web templates of standard and customer-modified applets in the Prior Customer Repository are preserved:

- When a Web template is used for an applet. Exception: for upgrades prior to Siebel 7.7, the postmerge utilities provide a new Web template for MVG applets that the postmerge utilities have converted to shuttle-enabled.
- All Web template item properties are preserved:
 - All control types are preserved.
 - The controls that are visible are preserved.

How the Siebel Database Upgrade Works ■

- No new controls from the New Siebel Repository are added or deleted. Exception: for upgrades prior to Siebel 7.7, MVG applets receive new controls as part of being shuttle-enabled.
- The location of controls in applets is preserved.
- Applet mode (whether an applet displays in More mode or Less mode) is preserved.
- Exception: for upgrades prior to Siebel 7.7, many flow-based form applets are converted to grid-based.
- Applet Drilldowns and their child objects are preserved:
 - Drilldown properties are preserved.
 - The columns that have drilldowns is preserved
 - The destination views of drilldowns are preserved

The following features are not preserved:

- Applet-level and application-level menus.
- Properties of controls or list columns. For example, row height, column width, caption, and pop-up icons are not preserved.
- As of Siebel 7.7, Pick and Association applets support in-line queries. This feature is implemented whether or not you choose ICL.

If you are upgrading from a Siebel 7.x release prior to Siebel 7.7, ICL has the following specific effects on applet controls and layout:

- Siebel 7.7 added two buttons in attachment applets: New File and New URL. An ICL merge does not add these buttons to applets.
- As of Siebel 7.7, MVG dialog boxes display only an OK button to close them. After an ICL merge, MVG dialog applets will have both OK and Cancel buttons.
- Siebel 7.7 added two new UI features, Query Assistant and Quick Print. An ICL merge does not add these buttons to applets. You must add these buttons as desired after completing the upgrade.
- In Siebel 7.7, applets have three default buttons: New, Delete, and Query. If you select an ICL merge, applets will keep their existing default buttons.
- In Siebel 7.7, the Reset button was removed from applets. If you select an ICL merge, applets keep the Reset button.

Navigation

Changes to navigation in the release you are upgrading to are implemented. This may affect screen, view, and applet layouts.

For example, in Siebel 7.7, a new navigation scheme was introduced. If you are upgrading from a release prior to Siebel 7.7, this navigation scheme is implemented during the upgrade regardless of whether you choose ICL. In parent list views, view tabs do not display in a row near the middle of the view as they did in 7.0x and 7.5x. You must select a record in the parent list applet to display the view tabs.

How ICL Handles Deleted and Obsolete Objects

An ICL merge treats deleted UI objects in the same way as a regular merge. If you deleted screens, views, or applets in the Prior Customer Repository, and they are present in the New Siebel Repository they will be present in the New Customer Repository after the merge. You will need to manually verify and remove these as desired.

In a regular merge, if an object is obsolete (inactive) in the New Siebel Repository, the object is inactive in the New Customer Repository after the merge. This is also true for an ICL merge. In addition, an ICL merge replaces the following object types with their counterparts from the Prior Customer Repository. The replacement object types are then set to inactive:

- Applet Toggle
- Control and child objects
- Drilldown and child objects
- List and child objects
- Chart and child objects
- Tree and child objects

Any other UI object types that are obsolete in the New Customer Repository are handled the same way as a regular merge.

ICL and Customer-Created UI Objects

If you created screens, view, or applets in the Prior Customer Repository, they are handled as follows:

- The ICL merge step ignores the Upgrade Ancestor property. For example, New Applet specifies Standard Applet as an upgrade ancestor. New Applet does not receive ICL handling. This means that New applet and all its child objects are copied intact to the New Customer Repository. The child objects do not replace UI objects in the New Customer Repository.
- For an ICL merge, the postmerge utilities copy new Web template files to the Tools installation directory. These files replace those in the new release and will be used to render customer-created objects.

ICL and the Upgrade Ancestor Property

The ICL merge step ignores the Upgrade Ancestor Property. Customer-created applets do not receive ICL handling. For example, you create Applet-B by copying standard Applet-A. In Applet-B you specify Applet-A as the upgrade ancestor. In the New Siebel Repository, Applet-A has a new button. After an ICL merge, Applet-B displays the new button, but Applet-A does not:

- During the regular merge, Applet-B receives the new button, because it is a descendant of Applet-A.
- During the ICL merge portion, the Web template configuration of Applet-A is copied from the Prior Customer Repository to the New Customer Repository. This Web template configuration does not contain the new button. This removes the new button from Applet-A.

How the Siebel Database Upgrade Works ■

- Because the ICL merge portion ignores the Upgrade Ancestor property, this change is not propagated to Applet-B. Applet-B retains the new button.

In a regular merge, both Applet-A and Applet-B display the new button.

On an ICL merge, the regular merge portion propagates upgrade handling of non-ICL-object types (Business Component, and so on) to descendants normally. For this reason, using the Upgrade Ancestor Property with an ICL merge is recommended.

How an ICL Merge Works

An ICL merge is a regular merge that contains two additional steps:

- The first step occurs at the end of the merge.
- The second step is part of the postmerge utilities, which run after the merge is complete.

Additional Merge Step

At the end of the regular merge portion, the ICL merge step runs and makes changes to ICL objects in the New Customer Repository:

- ICL objects from the new release, are deactivated, and -UPG is appended to their name. These objects are read-only and cannot be deleted.
- ICL objects from the Prior Customer Repository are copied to the New Customer Repository and are verified as active.

The ICL merge step writes status and issues to the merge log files.

Postmerge Utilities and Grid-Based Applets

The second step occurs after the ICL merge step completes and the merge is finished. The postmerge utilities run and make additional ICL changes:

- At Siebel 7.7, many form applets in the New Siebel Repository that were formerly flow-based are grid-based. After an ICL merge, flow-based form applets from the Prior Customer Repository will replace some of these grid-based applets in the New Customer Repository, particularly for upgrades from releases prior to Siebel 7.7. The postmerge utilities will convert these replacement flow-based applets to grid-based.
- For flow-based form-applets that are converted to grid-based, you can choose to put field labels on top or on the left when setting up the merge. The UI standard as of Siebel 7.7 is to place field labels on the left.

Postmerge Utilities and Web Template Files

The postmerge utilities copy-in new Web template files. These Web template files are similar to those in the release you are upgrading from. They provide page containers and supporting Web templates that preserve screen, view, and applet layouts. They also implement navigation changes. If you have edited the Web template files in the release you are upgrading from, you must reimplement your customizations in the new Web template files.

The postmerge utilities make the following changes to Web template files in the Siebel Tools installation directory:

- The utilities move the Web template files for the new release from `webtempl` to `\temp\webtempl`.
- The utilities copy Web template files from a subdirectory of `repatch\web_templates` to `\webtempl`.

These Web template files support the UI layout of the release you are upgrading from. They are similar but not identical to the Web template files included in that release. For example, if you selected the 7.5.3 and Label on Top ICL options, the postmerge utilities copy Web template files from the `753` and `TopLabel` subdirectories of `repatch`.

The postmerge utilities write status and issues to the postmerge utilities log.

ICL and HTML Style Sheets

The postmerge utilities do not copy-in a new version of the HTML style sheets, `main.css` and `printmain.css`. The style sheets for the new release are used to render all UI objects, including those copied-in from the Prior Customer Repository. If you customized the style sheets in the release you are upgrading from, you must reimplement your customizations in the style sheets in the new release.

What Happens at the Next Siebel Release?

You can perform an ICL merge for only every other upgrade. For example, if you performed an ICL merge when upgrading from Siebel 7.x to Siebel 7.7, performing an ICL merge when upgrading to the next release is not recommended.

At the next release before you run the repository merge, you must run a Siebel Tools utility that restores repository objects to the standard UI. The utility makes the following repository changes:

- Removes the ICL objects that were copied-in from the Prior Customer Repository.
- Activates the ICL objects with the `-UPG` suffix and removes the suffix. These `-UPG` objects are from the Prior Customer Repository (the repository in the release you are upgrading from). This reverses the effect of ICL and upgrades your UI to the current release.

A Tools utility provides a method for defining a filter to identify UI objects that you have modified since the prior upgrade. If an object has been modified since the prior upgrade, it is not replaced by its corresponding `-UPG` object. You must manually reconfigure these customer-modified objects after the merge as needed.

Related Topics

[“About the Siebel Repository Merge” on page 91](#)

[“About Inheriting Upgrade Behavior in a Siebel Upgrade” on page 96](#)

[“About the Siebel Postmerge Utilities” on page 98](#)

[“Reviewing Siebel Repository Object Property Conflicts” on page 258](#)

About the Database Configuration Utilities

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

The executables for doing an upgrade reside on the Siebel Server. You can use any upgraded Siebel Server to perform an upgrade of the Siebel Database.

The Siebel Database Server is not a server. It does not participate in processing business data like a Web server or Siebel Server. Instead, the Siebel Database Server is a set of files and scripts that provide inputs to the upgrade executables.

For best performance, install the Siebel Database Server files on the Siebel Server that you will use to perform the upgrade. You must manually select the Siebel Database Server for installation; it is not automatically installed when you install a Siebel Server.

The Siebel Database Server files will be installed at the same directory level as the Siebel Server in a directory called `dbsrvr`. For example, if the Siebel Server is installed in `C:\sba80\si_ebsrvr` (Windows), then the Siebel Database Server will be installed in `C:\sba80\dbsrvr`. You need to install only one Siebel Database Server.

To edit and execute Siebel Database Server procedures and maintenance scripts, you must have READ-WRITE access to the Siebel Server bin directories in `SIEBEL_ROOT` (Windows), `$SIEBEL_ROOT` (UNIX). You must also have READ-WRITE access to the Siebel Server Log directories and upgrade directory.

About Tuning Siebel Production Upgrade Files

Upgrades from: All Supported Siebel releases.

Environments: Production test environment only. Does not apply to production environment.

Platforms: Windows and UNIX only.

Databases: All databases.

The Upgrade Tuner allows you to tune the upgrade files generated by the Siebel Database Server Configuration Utilities. Tuning the production upgrep files can significantly reduce database downtime when performing a production environment upgrade.

Upgrade Tuner displays the following information and provides the following options:

- **Table creation times.** You can place tables with long creation times in parallel threads.
- **Index creation times.** You can assign index creation to run in parallel threads.

- **Zero-Row SQL commands.** You can review and inactivate SQL statements that do not affect any table rows.

The scripts used to upgrade your Siebel Database are generic. They update your Siebel Database to support all Siebel applications functionality. You can use this option to eliminate SQL statements that are not needed for your applications.

Operating System and RDBMS Support

Upgrade Tuner supports parallel threads for table and index creation for the combinations of operating system and RDBMS show in [Table 16](#).

Table 16. Upgrade Tuner Support for Parallel Table and Index Threads

Operating System	Oracle Database	IBM DB2	Microsoft SQL
Windows	Yes	No	No
UNIX	Yes	No	No

Upgrade Tuner is not supported for IBM z/OS. Upgrade Tuner supports zero-row SQL command deactivation for the combinations of operating system and RDBMS shown in [Table 17](#).

Table 17. Upgrade Tuner Support for Zero-Row SQL Deactivation

Operating System	Oracle Database	IBM DB2	Microsoft SQL
Windows	Yes	Yes	Yes
UNIX	Yes	Yes	Yes
z/OS	Not applicable	No	Not applicable

For information on which versions and releases of the operating systems and RDBMS products that Oracle supports for Siebel products, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

Upgrade Tuner is part of the Siebel Server and runs only under Microsoft Windows. Upgrade Tuner does not run under UNIX. To tune UNIX production upgrade files, you must copy them to a Windows platform, tune them, and move them back to the UNIX platform. Scripts are provided to move the files.

If you are a UNIX user and do not have a Siebel Server for Windows, contact your account manager or Oracle Global Customer Support to obtain one.

When to Use Upgrade Tuner

Use Upgrade Tuner in the production test environment to tune the upgrade files that perform the production upgrep. There is no need to tune the upgrade files that perform the production upgphys. You also do not need to tune the upgrade files that perform the development environment upgrep or upgphys.

Upgrade Tuner Modes

When you start Upgrade Tuner it displays four tabs:

- Process Information
- Parallelize Table Creation
- Parallelize Index Creation
- Deactivate 0-Row SQLs

Process Information Tab

This page displays the information sources that Upgrade Tuner is using. These sources include the Logparse summary.xml file, master .ucf file, and driver .ucf file.

You cannot edit the information on this page. Upgrade Tuner obtains the information by reading the summary.xml file.

Parallelize Table Creation and Parallelize Index Creation Tabs

The Parallelize Table Creation page and the Parallelize Index Creation page both have the same layout. These pages allow you to do the following:

- **Parallelize Table Creation tab.** This page displays the time required to create tables and allows you to assign table creation to parallel threads. Adding a table to a parallel thread does not add index creation for that table to the thread. Table and index creation are handled as separate steps during the upgrade.
- **Parallelize Index Creation tab.** This page displays the time required to create table indexes and allows you to assign index creation to parallel threads.

Creating parallel threads improves the upgrade performance by reducing the total time to create tables and indexes. You can create up to nine parallel threads. Each thread can have a maximum of ten tables or indexes. Tables or indexes not assigned to a parallel thread remain in the serial thread.

Deactivate 0-Row SQLs

This page allows you to activate or deactivate the SQL statements that do not affect any table rows and therefore any data. This capability improves the upgrade performance by eliminating SQL statements that may not apply to your data. This page lists only the SQL files that are executed natively by the RDBMS. It does not list SQL files that are executed using odbcsql.

Files Required to Run Upgrade Tuner

Upgrade Tuner requires the following files. The location of the files is the same on both Windows and UNIX hosts. [Table 18](#) displays these files.

Table 18. Files Required to Run Upgrade Tuner.

File	Location on a Windows Host
summary.xml	<i>SIEBEL_ROOT\log\upgprep_prod_VERSION</i> For example, <i>SIEBEL_ROOT\log\upgprep_prod_77\summary.xml</i>
master_upgprep_prod_VERSION.ucf	<i>SIEBEL_ROOT\bin\</i> For example, <i>SIEBEL_ROOT\bin\master_upgprep_prod_77.ucf</i>
schema*.ddl	<i>DBSRVR_ROOT\DBPLATFORM</i> For example, <i>DBSRVR_ROOT\oracl e\schema.ddl</i> , <i>schema_i1.ddl</i> , <i>schema_t1.ddl</i>
driver_upgprep_prod_VERSION.ucf	<i>DBSRVR_ROOT\DBPLATFORM\upgrade\VERSION</i> For example, <i>DBSRVR_ROOT\oracl e\upgrade\7_7\driver_upgprep_prod_77.ucf</i>
*.sql	<i>DBSRVR_ROOT\DBPLATFORM\upgrade\VERSION</i> For example, <i>DBSRVR_ROOT\oracl e\upgrade\7_7\pret.sql</i> , <i>preschm.sql</i>

How Upgrade Tuner Modifies Files

When you save your changes, Upgrade Tuner modifies the upgrade files. These are the files the Upgrade Wizard uses to upgrade the database. These files are as follows:

- [“Driver Configuration File” on page 115](#)
- [“Schema.ddl File” on page 116](#)
- [“SQL Files” on page 117](#)

Driver Configuration File

When you add or remove parallel threads and save your changes, Upgrade Tuner modifies the driver configuration file, for example *driver_upgprep_80.ucf*. The driver configuration file is a text file that contains a series of steps. The steps specify the commands that control the production upgprep. When you run the Upgrade Wizard to upgrade your database, it executes the steps in the driver configuration file.

The following actions are examples of steps that can appear in the driver configuration file:

- Making schema changes using the *ddlmp* utility and *schema.ddl*
- Making schema and data changes by executing SQL scripts

How the Siebel Database Upgrade Works ■

Upgrade Tuner manages the driver configuration file as follows:

- When you save your changes after the first session, Upgrade Tuner makes a copy of the file and appends `.orig` to the file name. It then modifies the file. For example, Upgrade Tuner copies `driver_upgprep_<release version>.ucf` to `driver_upgprep_<release version>.ucf.orig`. It then makes changes to `driver_upgprep_<release version>.ucf`.
- When you save your changes after the second session, Upgrade Tuner makes a copy of the file and appends `.old` to the file name. It then modifies the driver file. For example, Upgrade Tuner copies `driver_upgprep_<release version>.ucf` to `driver_upgprep_<release version>.ucf.old`. It then makes changes to `driver_upgprep_<release version>.ucf`.
- When you save your changes after the third session and all following sessions, Upgrade Tuner saves the driver file to `.old` again and then updates the driver file.

This file management strategy preserves the previous set of revisions to the file. It also preserves the original version of the file.

Schema.ddl File

When you run the Upgrade Wizard after the production upgrep it reads the driver file. The driver file contains steps that call the `ddlmp` utility. This utility uses `schema.ddl` as input to upgrade your database schema.

In the Parallelize Table Creation and Parallelize Index Creation pages, Upgrade Tuner displays the creation times for the tables and indexes in the `schema.ddl` file.

When you create parallel threads, Upgrade Tuner creates thread-files that have the same format as `schema.ddl` but contain only the table or index creation steps in the thread. Upgrade Tuner then adds steps to the driver file. These steps call the `ddlmp` utility, and specify the thread-files as input.

Upgrade Tuner manages `schema.ddl` and thread-files as follows:

- When you create a new thread and click Save and Exit, Upgrade Tuner creates a `schema.ddl` thread-file for the new thread.

For example, you do not have any parallel threads, and then create two new threads for table creation. When you exit, Upgrade Tuner creates a `schema_t1.ddl` and a `schema_t2.ddl` file. Upgrade Tuner also inserts steps in the driver file to execute the thread-files.

The `t1` thread-file contains the table creation information for the tables in Parallel Thread 1. Parallel thread 2 information is contained in the `t2` thread-file, and so on.
- When you create new threads in the Parallelize Index Creation page, the thread-files are named `i1`, `i2`, and so on. For example, the information for Parallel Thread 1 for index creation is contained in `schema_i1.ddl`.
- If you run Upgrade Tuner and change the tables or indexes assigned to a thread, Upgrade Tuner updates the thread-file for that thread. Upgrade Tuner does not create `.orig` or `.old` files for thread-files. Also, Upgrade Tuner does not change the step that executes the thread-file in the driver file.
- Upgrade Tuner does not revise the content of the main `schema.ddl` file (the serial thread) when you create thread-files. The thread-files duplicate the content in `schema.ddl`.

In the driver file, the order of execution of steps for `schema.ddl` and the thread-files is as follows:

- Table thread files beginning with file t1 (ddlmp in table creation mode)
- schema.ddl (ddlmp in table creation mode)
- Index thread files beginning with i1 (ddlmp in index creation mode)
- schema.ddl in index creation mode (ddlmp in index creation mode)

For both table and index creation, the parallel threads are executed first followed by the serial thread (schema.ddl).

TIP: To locate thread-file steps in the driver_upgrep_prod ucf file, query for "schema_".

SQL Files

When you make changes in the Deactivate 0-Row SQLs page, Upgrade Tuner makes changes to the SQL file containing the SQL command. Because the SQL file is already a step in the driver file, Upgrade Tuner does not modify the driver file.

Upgrade Tuner manages the SQL files as follows:

- When you first change an SQL file, Upgrade Tuner saves a copy of the SQL file and appends .orig to its file name. Upgrade Tuner then updates the SQL file.
- The next time you change the SQL file in Upgrade Tuner, it saves a copy of the SQL file and appends .old to the file name. Upgrade Tuner then updates the SQL file.
- Thereafter, when you modify the SQL file, Upgrade Tuner saves the SQL file to .old again and then updates the SQL file.

This file management strategy preserves the previous set of revisions to the file. It also preserves the original version of the file.

6

Siebel Database and UI Upgrade Planning

This chapter provides information to consider when planning your Siebel Database upgrade. This chapter includes the following topics:

- [“Upgrade Flow Orientation” on page 120](#)
- [“About Siebel Upgrade Planning Resources” on page 121](#)
- [“About Best Practices for Doing Your Siebel Database Upgrade” on page 123](#)
- [“About New Siebel Upgrade Features and Requirements” on page 126](#)
- [“About Upgrading Your RDBMS in the Siebel Environment” on page 127](#)
- [“About Siebel User Interface Changes” on page 128](#)
- [“About Database Sort Order in the Siebel Environment” on page 129](#)
- [“About the Siebel Party Model” on page 130](#)
- [“About Migrating Siebel HTML Attachments to Base Tables” on page 133](#)
- [“Upgrade Planning for Multilingual Siebel Deployments” on page 134](#)
- [“Upgrade Planning for Siebel Unicode Support” on page 135](#)
- [“Upgrade Planning for Siebel RC2 or AES Encryption” on page 137](#)
- [“Upgrade Planning for Siebel Web Template Files and Style Sheet” on page 137](#)
- [“Upgrade Planning for Siebel Access Control” on page 139](#)

Related Topics

[Chapter 3, “Simplified Overview of Siebel Database Upgrade Process”](#)

[Chapter 5, “How the Siebel Database Upgrade Works”](#)

[Chapter 4, “How to Perform a Siebel Database Upgrade”](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase One of the Siebel Database upgrade flow, as shown in [Figure 5](#). At this phase of the upgrade you are gathering information and upgrading your infrastructure.

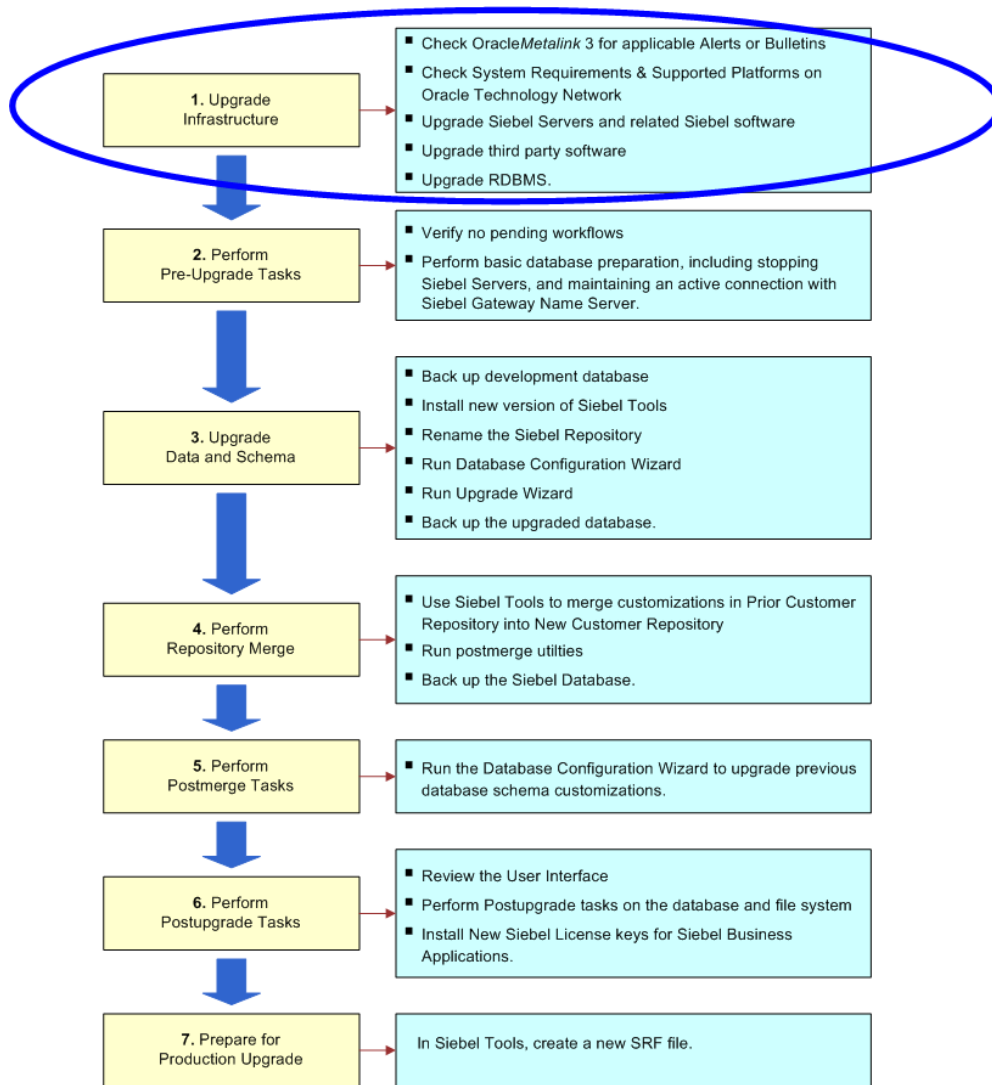


Figure 5. Phase One of Upgrade Flow

About Siebel Upgrade Planning Resources

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic lists important publications and resources for performing an upgrade. Review these as part of the upgrade planning process.

Product Documentation

Oracle's Siebel Product documentation is collectively called the Siebel Bookshelf. The Siebel Bookshelf is available on Oracle Technology Network (OTN) and Oracle E-Delivery. It might also be installed locally on your intranet or on a network location.

Siebel Installation and Deployment Documentation

The following publications are meant to be used in concert with installation or deployment of the Siebel application and can be found on *Siebel Bookshelf*:

- *Deployment Planning Guide*
- *Siebel Installation Guide for Microsoft Windows: Servers, Mobile Web Clients, Tools*
- *Siebel Installation Guide for UNIX: Servers, Mobile Web Clients, Tools*
- *Going Live with Siebel Business Applications*. Provides information about how to migrate customizations from the development environment to the production environment
- *Siebel System Administration Guide*. Details on how to administer, maintain, and expand your Siebel Servers
- *Security Guide for Siebel Business Applications*
- *Siebel Application Deployment Manager Guide*
- *Performance Tuning Guide*
- *Configuring Siebel Business Applications*. Provides information about configuring Siebel Business Applications in Siebel Tools
- *Siebel Data Model Reference (DMR)*. The DMR describes in detail the Siebel Database schema for a Release. It also lists certain types of schema changes. Use the DMR during upgrade planning to evaluate how data will be stored in the new Release. Consider obtaining a DMR for both the Release you are upgrading from as well as the Release you are upgrading to. Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services for information on ordering DMRs.

Oracle Technology Network

This site contains the following:

- **Siebel Bookshelf.** A searchable collection of Oracle's Siebel Product documentation.
- **Siebel System Requirements and Supported Platforms** on Oracle Technology Network. This document is the definitive list of system requirements and supported third-party products.

OracleMetalink 3

This site provides search access to Bulletins (previously published as Siebel Technical Notes) Alerts, Fix Packs, Release Notes, Maintenance Release Guides, Alerts, Technical Notes, and more:

- **Release Notes.** *Siebel Release Notes* on OracleMetalink 3 contain late-breaking information that the *Siebel Database Upgrade Guide* does not yet include.
- **Siebel Maintenance Release Guide** on OracleMetalink 3. Maintenance Release Guides contain important information about updates to applications in maintenance releases. See the applicable *Siebel Maintenance Release Guide* on OracleMetalink 3.
- **Bulletins and Alerts.** Bulletins and Alerts provide important information on specific upgrade issues. Bulletins related to a upgrade issues are located on OracleMetalink 3.

Alert 477833.1 (Doc ID) on OracleMetalink 3 (previously published as Siebel Alert 1002) and Alert 478177.1 (Doc ID) on OracleMetalink 3 (previously published as Siebel Alert 1179) are master Alerts particularly important to all upgrades. References to these and other Alerts and Bulletins are integrated throughout *Siebel Database Upgrade Guide*.
- **Troubleshooting.** Troubleshooting documents contain information about how to troubleshoot common upgrade issues such as error messages encountered during the upgrade process, and other unwanted behavior in Oracle's Siebel applications. Troubleshooting documents can be found for a variety of upgrade issues, including error messages found in upgrade logs. For more information browse Troubleshooting content on OracleMetalink 3.

Technical Account Manager

If you need assistance planning your upgrade or encounter problems during the upgrade, your Technical Account Manager can advise you on how best to use available Oracle resources.

Advanced Customer Services

Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services. They offer detailed implementation planning and technical consulting services as well as rapid response and resolution for critical technical issues affecting Siebel deployments.

Contact your Oracle sales representative for Oracle Advanced Customer Services for important information in the following areas:

- Migrating from Siebel Industry Solutions or Siebel Financial Services to Siebel Business Applications during upgrade
- Migrating to Unicode code page support during upgrade
- Changing operating system type during upgrade

About Best Practices for Doing Your Siebel Database Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic provides an overview of the best practices for planning upgrade resources, estimating the upgrade time line, and managing the data migration process.

Use the following steps to help plan your upgrade.

- 1 Determine your upgrade path.** First, refer to *Siebel System Requirements and Supported Platforms* on Oracle Technology Network to determine supported upgrade paths for major releases. Verify whether you can upgrade directly to the currently shipping release or whether you must upgrade to a previous release first. Second, refer to the *Siebel Maintenance Release Guide* on *OracleMetalink* 3 for the release to which you are upgrading. These Guides list the upgrade path from the major release to its most recent maintenance release. Maintenance releases roll up Fix Packs and also include new features. Because fixes to the upgrade process may exist in Fix Packs included in a maintenance release, you should install the most recent maintenance release before starting the upgrade.
- 2 Evaluate the complexity of the upgrade.** Determine the complexity of the upgrade effort based on Oracle's Siebel modules implemented, number of integration points, number of interfaces, total number of scripts, and number of user interface scripts.
- 3 Assess the current Siebel environment and evaluate the existing implementation.** Perform a detailed assessment of the current Siebel environment to determine how the implementation will be affected by the upgrade. Evaluate the current implementation in comparison with the architecture of the current release. The assessment will help you to identify areas where you can take advantage of new functionality to meet business requirements.
- 4 Estimate the level of effort to upgrade.** Determine the metrics and cost associated with each aspect of the upgrade. Determine the effort required to upgrade based on the results of your complexity evaluation, current environment assessment, and new functionality review. This will help you to estimate resources, time line, and costs.
- 5 Establish the upgrade team.** Assemble a cross-functional upgrade team that understands Siebel product architecture and performance best practices. Include IT professionals, executives, and users to ensure a broad base of experience in technical, business, and Siebel-specific skills.
- 6 Review interface migration tasks.** Determine the effort to migrate modified applets and views. This includes associating applets with Web template items and mapping them to Web template controls.
- 7 Plan for upgrade tuning.** Tuning your production upgrade scripts can significantly reduce downtime during the final stages of your upgrade. Examples of upgrade tuning include eliminating SQL statements that do not affect any data, executing long-running SQL statements in parallel, and executing table creation, table rebuilds, and index creation in parallel.

- 8 **Identify data migration tasks.** After the upgrade, there may be data migration and repository configuration tasks that must be performed manually. These tasks frequently involve customizations made in prior releases.
- 9 **Plan for end-user training.** Analyze the impact of change on the users, and develop a plan for end-user training and adoption.

The upgrade of your application requires several key things to be successful:

- A detailed understanding of customizations made to your current deployment
- Analysis and definition of the components within your enterprise
- Analysis of how to use new functionality provided by Oracle's Siebel software
- Strict adherence to industry best practices and best practices identified in this guide

The upgrade planning process will produce a roadmap for the entire upgrade project that outlines infrastructure, deployment, and training requirements.

Use the results of this process to develop a project plan that identifies required skills and resources for developing and deploying the upgraded application. This will help you with advance budgeting of resources, time, and training.

Upgrade Planning Best Practices

Here are important best practices to follow when planning an upgrade:

- Review *Siebel System Requirements and Supported Platforms* on Oracle Technology Network, *Siebel Release Notes* on OracleMetaLink 3, *Siebel Maintenance Release Guide* on OracleMetalink 3, and Alerts on OracleMetalink 3 related to upgrades to verify your upgrade path.
- Gather all relevant documentation that describes the current implementation, for example, requirements documents, design documents, and architecture context diagrams.
- Implement a change management program. For example, communicate rollout dates to users, schedule training, allow adequate time for users to adjust to the enhancements, and provide a process for end users to provide feedback to the project team.
- User adoption is critical to a successful upgrade. Provide access to a test environment that allows users to become familiar with the new version of the application, and provide end-user training on the upgraded application.

Database Planning Best Practices

Here are important best practices to follow when planning the upgrade of your database:

- Analyze the impact of the upgrade on table customizations that you have made. Determine if pre-upgrade data migration is required. Determine what postupgrade schema changes are required. For a list of tables that are affected by the upgrade, see [Appendix B, "Tables Modified or Seeded During a Siebel Upgrade."](#)
- Consider database layout in your planning. Plan to tune the database and database server for the upgrade, because settings and parameters for upgrade differ from those required for OLTP.

- If you are migrating multiple languages from a prior version, plan extra time (one to two weeks) for the repository merge process. The expected merge time may increase with the number of languages in the repository. You also might need to plan for additional installation-related tasks.
- For IBM DB2 platforms, consider increasing the size of your tablespaces before going live. Make sure that your custom tablespaces are large enough for upgraded tables. See ["Analyzing IBM DB2 Custom Tablespace Requirements for a Siebel Upgrade"](#) on page 174.

Production Database Upgrade Best Practices

Both the Prepare for Production step and the upgrade tuning process modify the SQL scripts used to perform the upgrade in the production test environment. In addition, it is common to further modify these scripts to meet local requirements.

The recommended way to perform the production upgrade is to use the SQL scripts that you have generated and modified for the production test upgrade. The steps in the upgrade process checklist for the production upgrade implement this as follows:

- Run the Database Server Configuration Utilities in the production test environment.
- In the Utilities, enter the information for the production environment instead of the production test environment. For example, you enter the ODBC connection for the production environment.

This configures the driver file to run against the production database rather than the production test database. It also configures the driver file to use the upgrade SQL files you generated for the production test upgrade.
- Run the Upgrade Wizard. The Upgrade Wizard uses the SQL files in the production test environment to upgrade the database in the production environment.

This approach has several advantages:

- You do not have to generate upgrade SQL files in the production environment and then manually transfer customizations to them from the production test environment.
- You do not lose any changes to the SQL files that were made by Siebel Upgrade Tuner in the production test environment.
- You do not have to run the Database Server Configuration Utilities in Prepare for Production mode again.
- With some exceptions, you do not have to perform database-related configuration tasks required Alerts or *Siebel Release Notes* on *OracleMetalink* 3.

If your network configuration prevents creating an ODBC connection to your production database from inside your production test environment, and you cannot complete your production upgrade, create a service request (SR) on *OracleMetalink* 3. Alternatively, you can phone Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers remain the same and are listed on *OracleMetalink* 3.

About New Siebel Upgrade Features and Requirements

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

The following new upgrade features are introduced for Siebel CRM 8.0:

- **Database Server Configuration Utilities enhancements.** The Database Server Configuration Utilities run in an InstallShield type wrapper under both Windows and UNIX. In addition, the utilities detect the local installation environment and automatically complete many of the fields requested by the configuration pages.
- **Additive Schema Changes.** This Database Server Configuration Utilities mode allows you to perform part of the production database upgrade without taking the production database offline. This reduces the production database downtime required for the upgrade. See [“About Siebel Additive Schema Changes Mode” on page 70.](#)
- **Multilingual upgrade.** Formerly, the upgrade process upgraded only the base language. All other deployed languages had to be upgraded manually. For Siebel CRM 8.0, the upgrade process upgrades the base language and all deployed languages. See [“Upgrade Planning for Multilingual Siebel Deployments” on page 134.](#)
- **ICL upgrade enhancements.** A new object property, ICL Upgrade Path, specifies what release the Upgrade Behavior property applies to. This allows objects to be nonpreservable for upgrades from releases prior to the specified release. For upgrades after the specified release, the objects are preservable. [“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 102.](#)
- **Repository merge for customized workflows.** The repository merge process will merge customizations to seeded workflows in the Prior Customer Repository into the related workflows in the New Customer Repository. The merge logic is the same as for other repository objects. [“Upgrade Planning for Siebel Workflow Designer” on page 152.](#)
- **Case insensitive and accent insensitive queries wizard.** A new wizard in Siebel Tools allows you to configure columns in the repository to support case insensitive and accent insensitive queries. You can either select columns manually or use an input file to run the wizard in batch mode. You run the wizard after the repository merge. [“About the Siebel Case Insensitivity Wizard” on page 75.](#)
- **Logparse utility reporting enhancements.** For upgrades on IBM DB2 and Oracle, benign errors are no longer included in Logparse summary reports. Logparse automatically checks all errors against a benign errors file. If an error is benign, Logparse does not include it in the summary report. [“About the Siebel Database Upgrade Log Files” on page 72.](#)

The following new upgrade features are introduced for Siebel 7.8:

- **Platform support for multiple versions of components.** You can install multiple versions of the Siebel Server and Siebel Web Server Extension on the same machine. You cannot install multiple versions of the Siebel Gateway Name Server.

For example, you are upgrading from Siebel 7.7 to the current release. You have thoroughly tested the Siebel Server in the current release and are ready to install it in your production environment. You can install the new Siebel Server on production machines where Siebel 7.7 of the Siebel Server is installed. You do not have to take the machines out of service and deinstall the 7.7 Siebel Server before installing the new Siebel Server.

This reduces the downtime required to transition from an installed version of an Enterprise component to a new version.

You cannot run two different versions of the same component on one machine. For example, in the scenario above, you cannot run both the 7.7 version of the Siebel Server and the current release version at the same time.

For information on how to install Siebel Enterprise components, including the Siebel Gateway Name Server, Siebel Server, and Siebel Web Server Extension, see *Siebel Installation Guide* for the operating system you are using.

- **Postmerge utilities can be run multiple times.** After the repository merge completes, the postmerge utilities no longer start automatically. You must start them manually, and you can run them multiple times.

If the postmerge utilities do not complete successfully, you do not always have to rerun the repository merge. Depending on the problem, you only have to rerun the postmerge utilities.

About Upgrading Your RDBMS in the Siebel Environment

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

If your currently installed RDBMS version is not supported in the new release, you must upgrade your RDBMS before performing the Siebel Database upgrade. For information on supported RDBMS versions, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

If your currently installed RDBMS version is supported in the new release, and you plan to upgrade your RDBMS, you can do so before or after upgrading your Siebel Database.

The following guidelines and requirements are for planning your RDBMS upgrade.

NOTE: When you upgrade the RDBMS, also be sure to upgrade your client database connectivity software. See *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

Oracle Database

- See Alert 477966.1 on Oracle *Metalink* 3 (previously published as Siebel Alert 1053) for planning information on upgrading an Oracle RDBMS. If you upgrade the Siebel Database and then upgrade the Oracle RDBMS, you must validate the Siebel Database schema after the upgrade. See [“Verifying an Upgraded Oracle RDBMS After a Siebel Upgrade” on page 303](#).
- Development, production test, and production upgrades on Oracle 10g can be run in CBO mode.
- If you plan to upgrade your RDBMS to Oracle Database 10g, see 478028.1 (Doc ID) on Oracle *Metalink* 3. This document was previously published as Siebel Technical Note 582. This document provides information on setting up Oracle Database 10g for use with Oracle Cost-based optimizer (CBO). Briefly, when using Oracle’s Cost Based Optimization with a Siebel Database, there are some requirements for database parameter configuration and for the Optimizer’s statistics generation on tables, columns and indexes.
- Upgrades on Oracle8i and Oracle9i must be run in Rule-Based Optimizer (RBO) mode. If you are running in Cost-Based Optimizer (CBO) mode, you must change to RBO mode before upgrading. For Oracle9i, you must return to CBO mode after the upgrade.
- Upgrades on Oracle Database 10g must be run in CBO mode.

IBM DB2

- After upgrading your IBM DB2 RDBMS, you must upgrade the database instance.
- The DB2 database must have 4-KB, 16-KB, and 32-KB tablespaces defined on it. Otherwise, your upgrade will not complete successfully.

About Siebel User Interface Changes

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

Several important changes to the user interface were introduced in Siebel 7.7:

- **New navigation scheme.** The key revisions are as follows:
 - The Show menu is replaced by a link bar. When you click a screen tab, the link bar displays below the screen tab and shows the views or view list for the screen.
 - A visibility filter menu, located in the parent list applet, provides alternative methods for filtering the data displayed in the applet.
 - To display view tabs, users click on a link in the parent list applet.
- **Revised UI configuration.** In prior releases, the placement of views and controls was based on the repository configuration and logic applied at run time. In Siebel 7.7, the placement of views and controls is declarative in the repository. No run-time logic is applied.

- **Expanded use of grid-layout applets.** To improve usability, most employee-specific form applets are converted to grid-based layout. The upgrade does not convert custom form-based applets to grid-based layout.
- **MVG shuttle applets.** MVG shuttle applets were introduced in Siebel 7.5x. In Siebel 7.7, MVGs are shuttle-enabled by default.

Because of these enhancements, plan a careful evaluation of the user interface after the upgrade. The upgrade logging utility lists any user interface-related problems encountered during the upgrade. For more information on the user interface, see *Fundamentals*.

About Database Sort Order in the Siebel Environment

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

Sort order (also called collation sequence) is specified during the initial installation of a database and defines the way in which the database sorts character data when executing queries. Sort order support depends on both the code page of the database and whether it is used in a development or a production environment:

- **Development environments.** For development databases, you must use a binary sort order due to the functional limitations of databases that use a nonbinary sort order.
- **Production environments.** For production databases, it is strongly recommended that you use binary sort order to prevent possible performance degradation.

The settings for binary sort order are unique for each database platform. See *Siebel System Requirements and Supported Platforms* on Oracle Technology Network to determine supported settings for your database platform.

Sort Order Considerations for Siebel Databases

If your deployment requires that you use a nonbinary sort order (for example, if your local language does not use binary sort order), you must consider several functional limitations that particularly affect development environment upgrades. If these limitations are unacceptable, consider re-creating your database to use binary sort order. However, be aware of these considerations:

- You cannot use Siebel Tools to generate a Siebel Repository file (SRF) on a database that uses a nonbinary sort order.
- You cannot perform a repository merge on a database that uses a nonbinary sort order.
- Databases that use nonbinary sort order might perform slower than databases that use binary sort order.

See *Siebel System Requirements and Supported Platforms* on Oracle Technology Network to verify sort order and code page requirements for your deployment.

About the Siebel Party Model

Upgrades: Applies to Siebel Financial Services upgrades from Siebel 7.x that have retained the Siebel 6.x form of household associations.

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

Siebel 7.x introduced a party table (S_PARTY) in which all persons and organizational units are held. Accounts, organizations, internal divisions, contacts, employees, positions, and households are all considered parties and can be referenced from this table.

Most of the tables that formerly contained this data still exist and are still used, but they are now extension tables to the S_PARTY base table. Data is loaded into the business components through an implicit join.

Additionally, Siebel 7.x uses a single-person table and a single-organization unit table. For example, Employees and Contacts are now combined in the same table (S_CONTACT). Similarly, internal and external Organization Units are now combined in the same table (S_ORG_EXT).

The S_PARTY table is the primary table in the Party or Single-Person model and is the base table for all Party business components.

Several extension tables support the Party Model:

- S_USER stores Siebel User information.
- S_EMP_PER stores attributes for Brand-Owner Employees and Partner Users who are considered agents of the Brand-Owner.
- S_BU stores Organization information.
- S_CONTACT
- S_ORG_EXT
- S_POSTN

Each nonperson party directly or indirectly has person members, such as employees or contacts.

The Party model makes several tables obsolete:

- S_EMPLOYEE. Its functionality is merged into S_CONTACT.
- S_ORG_INT. Its functionality is merged into S_ORG_EXT.
- S_EMP_POSTN has been replaced by S_PARTY_PER.
- S_EMPLOYEE_ATT
- S_ORG_INT_ATT
- S_POSTN_RPT_REL

Figure 6 depicts the Party changes to the data model that occur during upgrades from Siebel 6.x to Siebel 7.x.

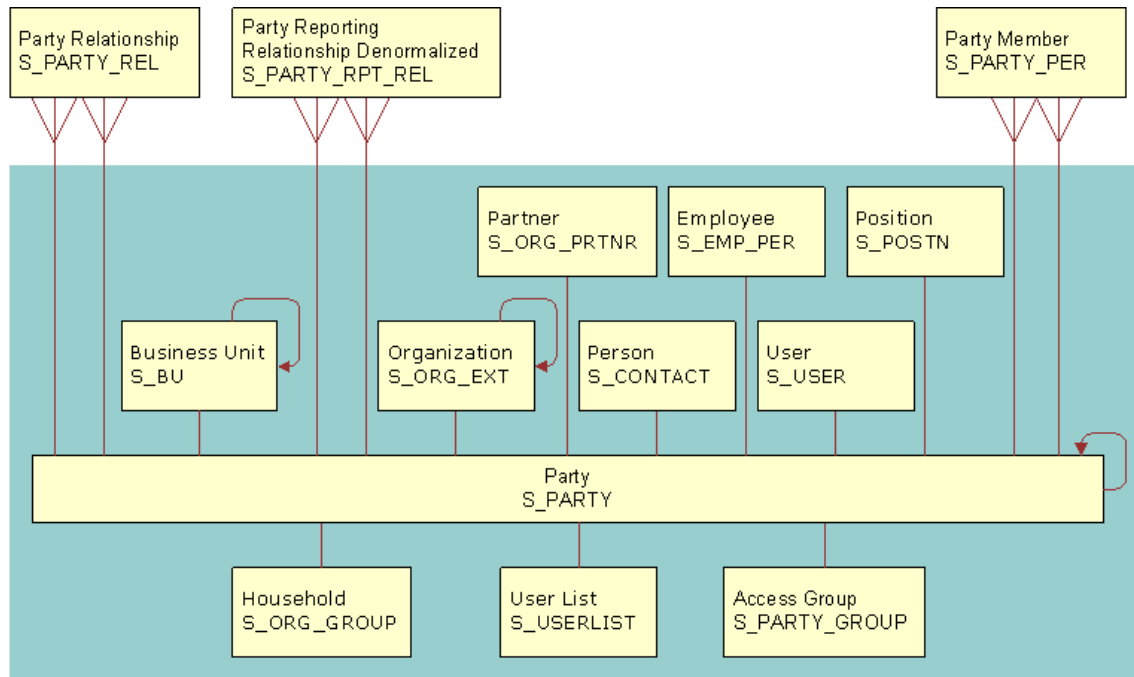


Figure 6. Party Model Changes

How the Party Model Is Implemented During an Upgrade

When you upgrade to Siebel 7.x, the upgrade process implements the Party model as follows:

Data Migration

- Migrates data from S_EMPLOYEE to S_CONTACT, S_USER, S_EMP_PER for standard columns
- Migrates data from S_ORG_INT to S_ORG_EXT, S_BU for standard columns
- Creates S_PARTY records for each previous contact, position, employee, account, division

Business Component Definitions

- Updates business component definitions to reference S_PARTY as the Primary Table (for example, Employee, Contact, Position, and Account business components)
- Changes standard and custom joins on S_EMPLOYEE to S_CONTACT, S_USER, S_EMP_PER
- Changes standard and custom joins on S_ORG_INT to S_ORG_EXT
- Sets implicit joins for custom fields created on business components that have been retargeted to S_PARTY. For example, if a custom field, Alternate Phone, existed on the Contact business component, the upgrade initiates the following actions:

- Retargets Contact business component to S_PARTY
- Defines join to S_CONTACT from S_PARTY on Contact business component
- Sets implicit join for the Alternate Phone field

How the Party Model Affects Siebel Financial Services Household Data

For Siebel 7.0.x, the Party model changed the relationship between a household and the following entities for Siebel Financial Services:

- Policy/Financial Account
- Activity
- Claim
- Service Request
- Opportunity
- Company

The relationship changed as follows:

- In Siebel Financial Services 6.x, these entities could be associated directly with a household.
- As of Siebel Financial Services 7.0.x, these entities cannot be associated directly with a household. Instead, they are associated with a contact. You associate an entity with a household, by adding a contact associated with the entity to the household.

To implement direct relationships between the entity tables and a household table in Siebel Financial Services 6.x, intersection tables were used for many-to-many relationships. A foreign key was used for one-to-many relationships. This design allowed a contact to be assigned to an entity but not be part of the household assigned to that entity. This caused possible data integrity problems, which the Party model resolves.

The tables required for maintaining the 6.x direct-relationship design are retained in Siebel 7.x. You can choose to maintain direct relationships between households and entities. However, this is not recommended. To maintain the 6.x direct-relationship design, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

How Siebel Financial Services 6.x Household Data Is Migrated

During the upgrade to Siebel 7.x, relationships in S_CONTACT_REL are migrated to S_PARTY_REL. Relationships in S_PER_ORG_UNIT are migrated to S_PARTY_PER.

When defining relationships between entities, no records are written to S_PARTY. Instead, the "PARTY (FIN)" and "Party Relationship To" business components drive the relationships displayed in the Relationship Hierarchy applet and adjacent Party Relationship list applet:

- S_PARTY includes the following fields:
 - PAR_PARTY_ID. This identifies the parent entity.
 - ROW_ID
 - NAME
 - PARTY_TYPE_CD. This identifies the type of entity.
- S_PARTY_REL includes three fields:

- PARTY_ID. This identifies the entity that has the relationship (explicit owner).
- REL_PARTY_ID. This identifies the entity that the relationship is with (implicit owner).
- REL_TYPE_CD. This identifies the type of relationship.

S_PARTY_REL can be used to define custom relationships such as lawyer, accountant, board member, and influencer. Valid relationships can be created between the following entities:

- Contact-to-contact
- Contact-to-household
- Contact-to-company
- Company-to-contact
- Company-to-household
- Household-to-contact
- Household-to-household
- Household-to-company
- Company-to-company

About Migrating Siebel HTML Attachments to Base Tables

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

The upgrade to Siebel 7.7 migrates data from the S_WEB_CNTNT table to the S_CB_CNTNT_SET, S_CB_ASSET, and other Content Base tables.

HTML Attachments, introduced in Siebel 7, are upgraded into Content Base tables. The Content Base table system allows content to be stored in the database or in the Siebel File System. The system stores several types of content including plain text, markup languages (for example, SGML and XML), and image files (for example, GIF and JPEG). The system breaks up the content when storing it and reassembles the content before rendering it.

Upgrade Planning for Multilingual Siebel Deployments

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

As of Siebel CRM 8.0, the upgrade process upgrades both the primary (base) language and all deployed languages. This applies to upgrades from Siebel 7.x on both non-Unicode and Unicode databases.

CAUTION: Each multilingual deployment must contain a primary language as well as those that are designated as secondary, or non-primary. For instance, a deployment may have ENU as the primary language and FRA, and JPN as its non-primary languages. Replacing ENU, the primary language in this example, with one of the non-primary languages during upgrade is not supported, as it will result in severe data corruption, making recovery extremely difficult.

The development and production upgrep process does the following for the primary language and all deployed languages:

- (Development upgrep). Imports UI strings into the Prior Standard Repository, New Customer Repository, and New Siebel Repository.
- Imports seed data into the database.
- In the S_LST_OF_VAL table, an SQL script activates all newly imported records and configures them to support multilingual lists of values (MLOVs). LOVs remain enabled for the primary language.
- For upgrades from Siebel 7.0.x, an SQL script makes required changes to the Marketing demand creation program table (S_SRC_DCP) and to the demand creation LOV or MLOV.

No language-related manual tasks are required to upgrade the deployed languages, except where noted.

Validation

Prior to starting a development or production upgrep, you must have installed the new release's language packs for all deployed languages.

The Database Server Configuration Utilities validate deployed languages by comparing the status of language packs installed on the Siebel Server with the language IDs of records in the S_LST_OF_VAL table. If there are Language IDs in S_LST_OF_VAL but no corresponding deployed language on the Siebel Server, the validation fails and you cannot continue the upgrade.

Prior to starting a development or production upgrep, you must review the records in S_LST_OF_VAL, using Siebel Tools. Remove or deactivate records for languages that you do not have deployed.

For more information, see [“Running the Siebel Database Configuration Wizard Under Windows” on page 218](#), or [“Running the Siebel Database Configuration Wizard Under UNIX” on page 221](#) depending on your operating system.

CAUTION: Regardless of whether you have American-English deployed, there will be records in S_LST_OF_VAL with Language Name = English-American. Do not deactivate or delete these records. The validation process ignores them.

Restrictions

The following restrictions apply to multilingual upgrades:

- Installed languages must be deployed. If you have installed a language but not deployed it, the language will not be upgraded.

A deployed language is one where you have installed the language pack, imported the UI strings into the Siebel Repository, imported the seed data into the database, activated the seed data records added to S_LST_OF_VAL, and configured the records for MLOVs.

- You cannot select which languages to upgrade. The Upgrade Wizard upgrades all deployed languages.
- You cannot add or drop a language during an upgrade. If you want to add a language that is not currently deployed, do so after the upgrade. For more information on adding new languages, see *Global Deployment Guide* and *Siebel Installation Guide* for the operating system you are using.

Upgrade Planning for Siebel Unicode Support

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

For Western European languages and Japanese, Oracle’s Siebel applications support both non-Unicode and Unicode code pages. For all other supported languages, Siebel applications support only Unicode code pages.

Unicode is recommended if your installation uses Siebel Email Response, correspondence, or similar functionality, particularly if content is generated on a separate system.

This is because Siebel applications use Unicode internally. If the RDBMS is not using Unicode, Siebel applications convert content from Unicode to the database code page. Using a Unicode code page for the database prevents string conversion problems on text content.

Before converting to Unicode, your encryption method must be RC2 or AES.

CAUTION: Migrating to Unicode is more complex than simply importing your existing data into a Unicode database. Failure to execute the migration correctly can result in serious data corruption or unrecoverable data loss. For this reason, Oracle’s Advanced Customer Services participation is mandatory. To perform a Unicode migration, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle’s Application Expert Services.

For a list of supported languages and code pages, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

For information about Unicode and global deployment for Siebel Business Applications, see *Global Deployment Guide*.

For information on upgrading to RC2 encryption, see *Security Guide for Siebel Business Applications*.

Planning Considerations for the Unicode Migration

Migrations to Unicode require the assistance of Oracle's Advanced Customer Services.

Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services to migrate your upgraded database from a non-Unicode code page to Unicode. To perform the migration, you can use either database vendor native utilities or Siebel utilities.

If you are planning to migrate your upgraded application to Unicode, consider the following points:

- **Database size increase. Migration to Unicode increases the size of your database.** For this reason, you need to allocate additional space for your database before migrating to Unicode. For more information, create a service request (SR) on Oracle *Metalink* 3. Also, you may contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- **IBM DB2 data truncation.** Migration to Unicode may cause truncation of certain data in DB2 databases. In the past, long columns with a type of varchar could have a maximum length of 16,383 characters. However, in Unicode, the maximum length of long columns with a type of varchar is 16,350. During the migration to Unicode, long columns of type varchar that exceed 16,350 are truncated. To prevent this, you can perform tasks to identify which data may be truncated and take appropriate measures before migration. For more information, create a service request (SR) on Oracle *Metalink* 3. Also, you may contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services. See also "[Identifying IBM DB2 Long Columns for Truncation in a Siebel Upgrade](#)" on [page 177](#).
- **IBM DB2 custom tablespace information.** The upgrade does not preserve custom tablespace information for DB2 databases. This presents a problem during your migration to a Unicode code page, because you need to know which tables need to be re-created. You must modify the upgrade scripts to handle custom tablespaces.

For instructions about how to modify upgrade scripts to handle custom tablespaces contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- **Third-party product integration.** Migration to Unicode may affect integration with third-party systems. For more information, create a service request (SR) on Oracle *Metalink* 3. Also, you may contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- **Handheld and wireless support.** Certain handheld and wireless application platforms do not support Unicode. For more information, create a service request (SR) on Oracle *Metalink* 3. Also, you may contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

- **Environment code pages.** You cannot use a non-Unicode code page for your development environment, and then later migrate to Unicode for your production environment.

Supported Types of Unicode

The current release supports two types of Unicode:

- **UTF-8.** UTF-8 uses the same encoding for Western European languages. It occupies one byte for Western European languages and up to three bytes for some Asian languages, such as Japanese.
- **UCS-2.** UCS-2 is supported for IBM DB2 and Microsoft SQL Server databases. UCS-2 does not map one-to-one with Western European languages. It occupies two bytes for all languages.

Upgrade Planning for Siebel RC2 or AES Encryption

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

Siebel 7.7 introduced support for the AES encryption method—the government standard for secure applications. Siebel Business Applications continue to support RC2 data encryption.

The default encryption method (called the standard encryptor) is no longer supported. Data that used the standard encryptor cannot be read by applications in the current release. You must upgrade your encryption method to RC2 or AES.

Use the Encryption Upgrade Utility to convert unencrypted data and data that was encrypted using the standard encryptor to the RC2 or AES encryption method. You must run the Encryption Upgrade Utility even if you are upgrading to stronger encryption, for example, from RC2 56-bit to RC2 128-bit encryption. Upgrading to RC2 128-bit or AES encryption requires the Siebel Strong Encryption Pack.

To upgrade your encryption method, see *Security Guide for Siebel Business Applications*.

Upgrade Planning for Siebel Web Template Files and Style Sheet

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Web template files help define the layout and formatting of the user interface, such as views, applets, and controls. The Siebel Web Engine in the Siebel Server uses the Web template files to build Web pages, which it then forwards to the Web Server.

A Web template file contains regular HTML, WML, or XML tags interspersed with Siebel tags. Siebel tags are prefixed by “swe” and contain placeholders for UI objects such as controls and data. HTML formatting tags are defined in a cascading style sheet (main.css).

When you install a new release of the Siebel Server or Siebel Tools, you receive a new set of Web template files and a new style sheet file. The upgrade process does not use your existing files. If you have manually customized your Web template files or style sheet file, you must evaluate whether to reimplement these customizations in the new Web template files and style sheet file.

Observe the following planning guidelines for reimplementing customizations:

- Resolve any UI problems related to object definitions in the Siebel Tools repository first. While doing so, review the areas of the new UI where you plan to implement Web template file customizations.
- Evaluate existing customizations to Web template files, and decide which ones to reimplement. Changes to the UI in the new release may make some customizations obsolete.
- Document why each customization was reimplemented. This reference will help you evaluate customization issues later.
- Use formal change control for managing versions of the Web template files and style sheet file. This allows you to maintain orderly distribution of the files to developers.
- Applets typically have a separate Web template file for each applet mode. Customize all the mode Web template files for an applet at the same time. This allows you to verify applet functionality in a single test pass in the UI.
- Individual Web template files are typically used by multiple screens, views, or applets. Set up formal test plans that verify customizations are correct across all UI objects that use each Web template file. This reduces the amount of time required to verify customizations and prevents unintended changes to the UI.
- Reimplement style sheet customizations in two passes. On the first pass implement only those changes required to UI usability issues. On the second pass, implement the remaining style sheet customizations after Web template customizations are complete. This shortens the time required to resolve functional problems in the UI.

After customizations have been completed, the Web template files and style sheet file must be copied to the Siebel Servers in the environment. The files must also be included in upgrade kits sent to remote sites.

Web template files have an .swt file extension and are located in the webtempl directory in both the Siebel Tools and Siebel Server installation.

The style sheet file is located as follows in the installation directory of Siebel Tools and the Siebel Server (Windows path syntax):

Siebel Tools: \publ ic*lang*\files\main.css

Siebel Server: *SIEBEL_ROOT*\webmaster\files*lang*\main.css

where *lang* is the installation language, for example, enu.

The steps for reimplementing customizations to these files and copying them to new environments are included in the upgrade process topics in [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

For information on how Web template files and style sheet file work, see *Configuring Siebel Business Applications*.

Upgrade Planning for Siebel Access Control

Upgrades from: Siebel 7.0.x.

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

Access control was significantly revised in Siebel 7.5. Access control refers to all mechanisms that control visibility of screens, views, and data within Siebel Business Applications. Access control includes, but is not limited to positions, responsibilities, organizations, and access groups.

To implement access control within your Siebel Business Applications, your Siebel administrator creates relationships between people and resources (a more general term for data that includes views and functionality). These relationships or policies are authorizations. Both people and resources can be grouped and placed in hierarchies to simplify administration.

External users, such as customers and channel partners, can be assigned varying access levels that control visibility of data and application functionality. When planning access policies, consider the following:

- The complexity of access control policies (one data item or group of data items can be accessed by one or many users or groups, but not by all).
- The amount of content that is distributed by the Siebel Business Applications, including Master data (data that is static and referential, such as Products) and Customer data (data that is created and managed by users of applications, such as Opportunities).
- The number of users and entities that access the data. Also consider the complexity of relationships between users (partners, competitors, browsers, customers).

For more information on access control, see *Security Guide for Siebel Business Applications*.

Person, Household and Service Request Visibility

Beginning with Siebel 7.5, Person, Household, and Service Request can be made visible to multiple organizations, also called Business Units. Siebel 7.5 introduced several new tables to support this:

- S_CONTACT_BU
- S_ORG_GROUP_BU
- S_SRV_REQ_BU

The upgrade to Siebel 7.7 populates the S_CONTACT_BU, S_ORG_GROUP_BU, and S_SRV_REQ_BU tables with one record for each record in the S_CONTACT, S_ORG_GROUP, and S_SRV_REQ tables. After the upgrade, Contacts, Households, and Service Requests continue to be visible from the Business Unit they belonged to before the upgrade.

Access Group and Userlist Attributes

In Siebel 7.5, two new Siebel Extension tables were added to the S_PARTY, S_PARTY_GROUP and S_USERLIST tables to hold Access Group and User List attributes, respectively.

The upgrade to Siebel 7.7 adds records to the S_PARTY_GROUP and S_USERLIST tables for existing S_PARTY Access Group and User List records.

To support Multi-Org visibility, the upgrade also adds corresponding intersection table records to the S_PARTY_GRP_BU and S_USERLIST_BU tables.

Bulletin 476479.1 (Doc ID) on Oracle *Metalink* 3 (previously published as Siebel Technical Note 312) provides guidance and best practices for implementing access control. This Bulletin includes background information about the Access Group access control mechanism implemented in Siebel CRM 7, discusses migration considerations, and outlines steps for deploying Access Group access for Siebel Business Applications.

For detailed information about access control, see *Security Guide for Siebel Business Applications*.

7

Application Planning for a Siebel Upgrade

This chapter provides information on Siebel Business Applications and issues to consider when upgrading your Siebel Database. This chapter includes the following topics:

- ["Upgrade Flow Orientation" on page 142](#)
- ["Determining Your Siebel 7.5.x Release Level" on page 143](#)
- ["Upgrade Planning for Migrating Siebel Address Data" on page 144](#)
- ["Upgrade Planning for Siebel Employee Relationship Management \(ERM\)" on page 146](#)
- ["Upgrade Planning for Siebel Marketing" on page 146](#)
- ["Upgrade Planning for Siebel Workflow Designer" on page 152](#)
- ["Upgrade Planning for Handheld Devices in the Siebel Environment" on page 156](#)
- ["Upgrade Planning for Resonate Central Dispatch in the Siebel Environment" on page 157](#)
- ["Upgrade Planning for Siebel String Translation" on page 157](#)
- ["Upgrade Planning for Siebel Configurator" on page 158](#)
- ["Upgrade Planning for Siebel Personalization" on page 158](#)
- ["Upgrade Planning for Siebel Pricer and Order Management" on page 159](#)
- ["Upgrade Planning: Additional Siebel Application Changes" on page 159](#)

Related Topics

[Chapter 5, "How the Siebel Database Upgrade Works"](#)

[Chapter 4, "How to Perform a Siebel Database Upgrade"](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase One of the Siebel Database upgrade flow, as shown in [Figure 7](#). At this phase of the upgrade you are gathering information and upgrading your infrastructure.

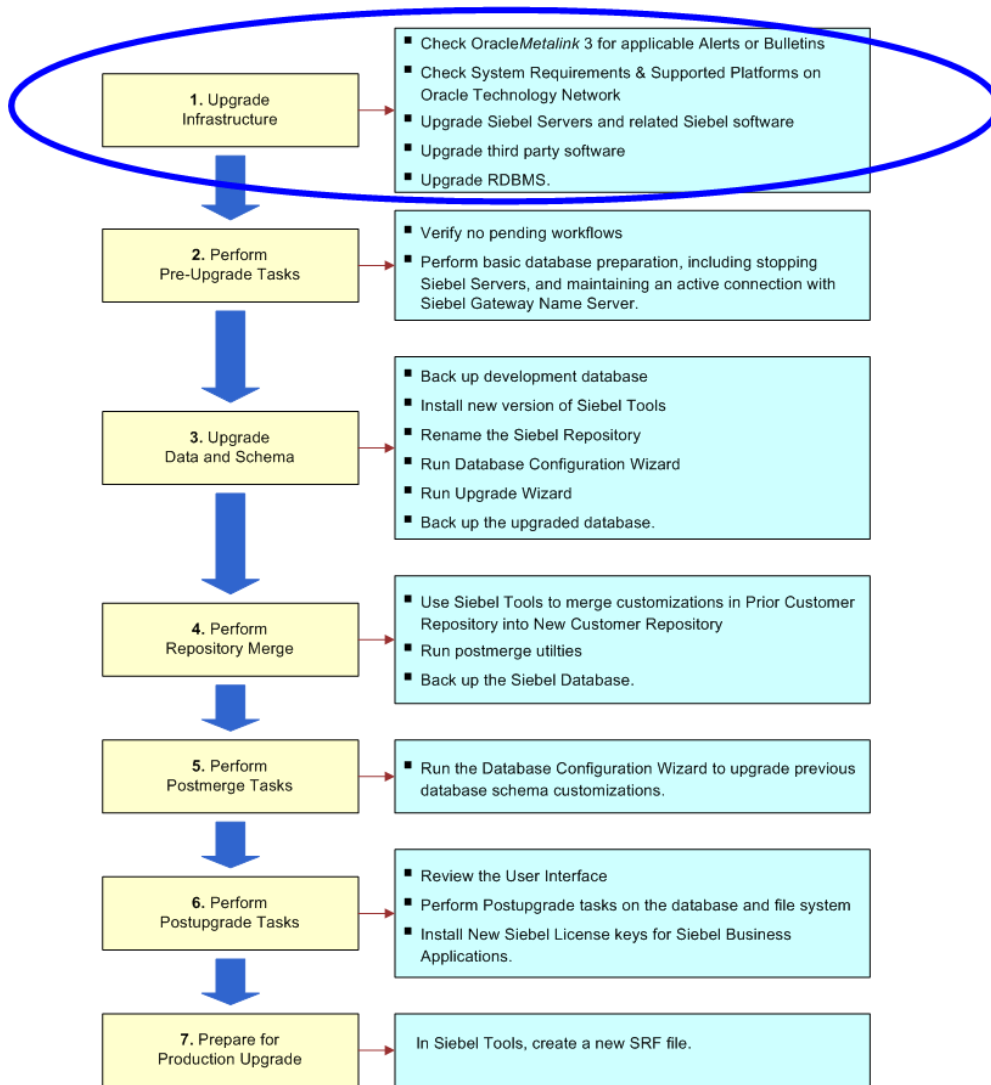


Figure 7. Phase One of Upgrade Flow

Determining Your Siebel 7.5.x Release Level

Upgrades from: Siebel 7.5.x.

Environments: Development environment only.

Before upgrading your enterprise to the new release, determine your 7.5.x release level.

When you run the Database Server Configuration Utilities to perform the upgrade, the utilities prompt you to choose which release you are upgrading from. The choices include: 7.5.2, 7.5.2.200+, and 7.5.3.

If you choose the wrong 7.5.x release level, the repository merge will complete successfully, but you may have UI configuration problems.

The release level of an installed release is contained in the base.txt file located in directories throughout the enterprise installation.

CAUTION: Perform the following tasks before upgrading enterprise components to the new release. Installing the new release replaces the base.txt files in the old release.

Determining Your Release Level

Perform this task to identify your 7.5.x release level.

To determine your release level

- 1 On the machine where the Siebel Server you will use to perform the upgrade is installed, locate the Siebel Server installation directory, siebsrvr.
- 2 In the siebsrvr directory, open base.txt and write down the release level.
For example, in the following base.txt string, the release level is 7.5.3:
`7.5.3 SIA [XXXXX] LANG_INDEPENDENT full release`
- 3 Store the 7.5.x release level where you can refer to it when you run the Database Server Configuration Utilities to upgrade the Siebel Database.

Verifying the Repository Update Is Installed

Perform this task if your release level is 7.5.2.200+ or 7.5.3.

The 7.5.2.200+ and 7.5.3 maintenance releases included a repository update that must be installed manually. Verify that this update is installed. If you have not installed the update, refer to 546768.1 (Doc ID) *Maintenance Release Guide for Siebel 7.5.3* on [OracleMetalink](#) 3.

Verifying the repository update is installed

- 1 Open Siebel Tools.
- 2 In the Object Explorer, select Applet.

- 3 Query for Account Team Mvg Applet, and select this applet.
- 4 In the Object Explorer, select Control, a child object of Applet.
- 5 With the Controls list active, query for *shuttle.
- 6 Verify the following controls are included in the query returns:
 - Add - Shuttle
 - AddAll - Shuttle
 - Remove - Shuttle
 - RemoveAll - Shuttle
 - LabelSelected - Shuttle
- 7 Choose one of the following actions:
 - If all the controls are present, the repository update is installed. No further action is required.
 - If none of the controls are present, the repository update is not installed. Refer to the *Maintenance Release Guide for Siebel 7.5.3* on *OracleMetalink 3* and install the update.
 - If some of the controls are present, but not all, create a service request (SR) on *OracleMetalink 3*, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Upgrade Planning for Migrating Siebel Address Data

Upgrades:

- From Siebel Financial Services 7.0.x to Siebel Industry Applications (SIA) 8.0.x
- From Siebel Business Applications 8.0.x to Siebel SIA 8.0.x
- From Siebel Financial Services 6.2.1 on IBM z/OS platforms to Siebel SIA 8.0.x on IBM z/OS platforms

NOTE: This topic does not apply to Siebel Business Applications (HOR) that you are upgrading to a later release of Siebel Business Applications (HOR).

Siebel 7.7 (and Siebel 7.5 SIA) changed the way address information is stored. You must perform several tasks to ensure address migration is handled correctly.

Previous Releases

In previous releases, address data was stored as follows:

- The relationship between person and address was 1:M and was stored in the table S_ADDR_PER.
- The relationship between account and address was 1:M and was stored in S_ADDR_ORG.
- Both tables included a column ADDR_NAME, which is a computed value based on other attributes in the address table.

- The user key for S_ADDR_PER included PER_ID and ADDR_NAME.

Siebel 7.7

In Siebel 7.7, the relationship between person and address and between account and address is M:M. Address information is stored in S_ADDR_PER. The relationship between addresses and contacts and between addresses and organization is stored in S_CON_ADDR.

The upgrade process revises storage of address data as follows:

- Inserts data into S_CON_ADDR from S_ADDR_PER and S_ADDR_ORG.
- Migrates data from S_ADDR_ORG to S_ADDR_PER. The table S_ADDR_ORG is obsolete.
- Sets S_ADDR_PER.PER_ID to NULL. It is no longer used as part of the user key but remains part of the physical key.
- Uniqueness of addresses in S_ADDR_PER is enforced only on ADDR_NAME.
- The table S_CON_ADDR becomes the intersection table for data stored in S_ADDR_PER and account or contact data stored in S_ORG_EXT or S_CONTACT respectively.

How Address Data Is Preserved

Because PER_ID is no longer part of the user key for S_ADDR_PER, the ADDR_NAME must be unique for all records.

It is possible that records within or across S_ADDR_ORG and S_ADDR_PER could have the same ADDR_NAME. If this occurs, the ADDR_NAME for one of the records is preserved, and the upgrade process appends the ROW_ID to ADDR_NAME for the others. This prevents records from being deleted and preserves all records from both tables.

How to Manage Address Migration

You must perform the following tasks to migrate address data:

- Before upgrading the database, you must run a script to identify records that have the same ROW_ID between S_ADDR_PER and S_ADDR_ORG. You must eliminate duplicate row IDs.
- You must evaluate whether to modify upgrade scripts to migrate address data in custom extension columns in S_ADDR_PER and S_ADDR_ORG. Review the new schema and determine if your custom columns can be mapped to standard columns. If no matching standard columns exist, create new columns on the target tables. During the database upprep, you do this after running the Database Server Configuration Utilities but before running the Upgrade Wizard.
- After the upgrade is complete, review the records in S_ADDR_PER and eliminate duplicate and obsolete records.

To manage address migration, follow the steps in [“Process of Upgrading a Siebel Production Test Environment” on page 35](#). Each of the address migration tasks is included as a step in this process. Each step refers you to a procedure for performing the task.

Upgrade Planning for Siebel Employee Relationship Management (ERM)

Upgrades from: Siebel 7.0.x and 7.5.x.

Platforms: Windows, UNIX, IBM z/OS.

Siebel 7.7 ERM introduced important changes to workflows, Siebel Training, Microsite management, and Group News.

Workflows

Siebel 7.5.3 included approval business process workflows in the Sample database. If you activated these workflows, or created workflows containing approval steps, you must manually upgrade these workflows.

Siebel Training

In previous releases, the product catalogs containing Siebel Training courses were of type Buying. Siebel 7.7 provides a new catalog type called Training. If you have catalogs that contain both Siebel Training products and other kinds of products, revise these catalogs so that they contain only Siebel Training products. This prevents nontraining products from being moved to the training catalog during upgrade.

Microsite and Group News

If you have created customized microsite or Group News page sections, you must revise the associated business component and applet definition.

Upgrade Planning for Siebel Marketing

Upgrades from: Siebel 7.0.x and 7.5.x.

Platforms: Windows, UNIX, IBM z/OS.

Siebel 7.7 introduced three architectural changes to Siebel Marketing:

- **Simplified hierarchy.** Program occurrences and campaign occurrences have been removed. This simplifies the Siebel Marketing hierarchy.
- **Merged Business Objects.** In previous releases, there were two business objects that supported similar campaign functions, DBM Campaign and Campaign. Siebel 7.7 merged all campaign administration functions into the Campaign business object.
- **Marketing Server Migrated to Oracle Business Intelligence Enterprise Edition.** The Marketing Server is obsolete. Its functions have been migrated to Oracle Business Intelligence Enterprise Edition and are implemented on the Oracle Business Intelligence Server.

Simplified Hierarchy

The Program Plan > Campaign Plan > Campaign Occurrences hierarchy has been simplified. The new hierarchy is Programs > Campaigns > Waves.

During the upgrade, objects that were related to the campaign occurrence are re-parented to the surviving campaign. Child objects are re-parented from a campaign occurrence to a campaign, as shown in Figure 8.

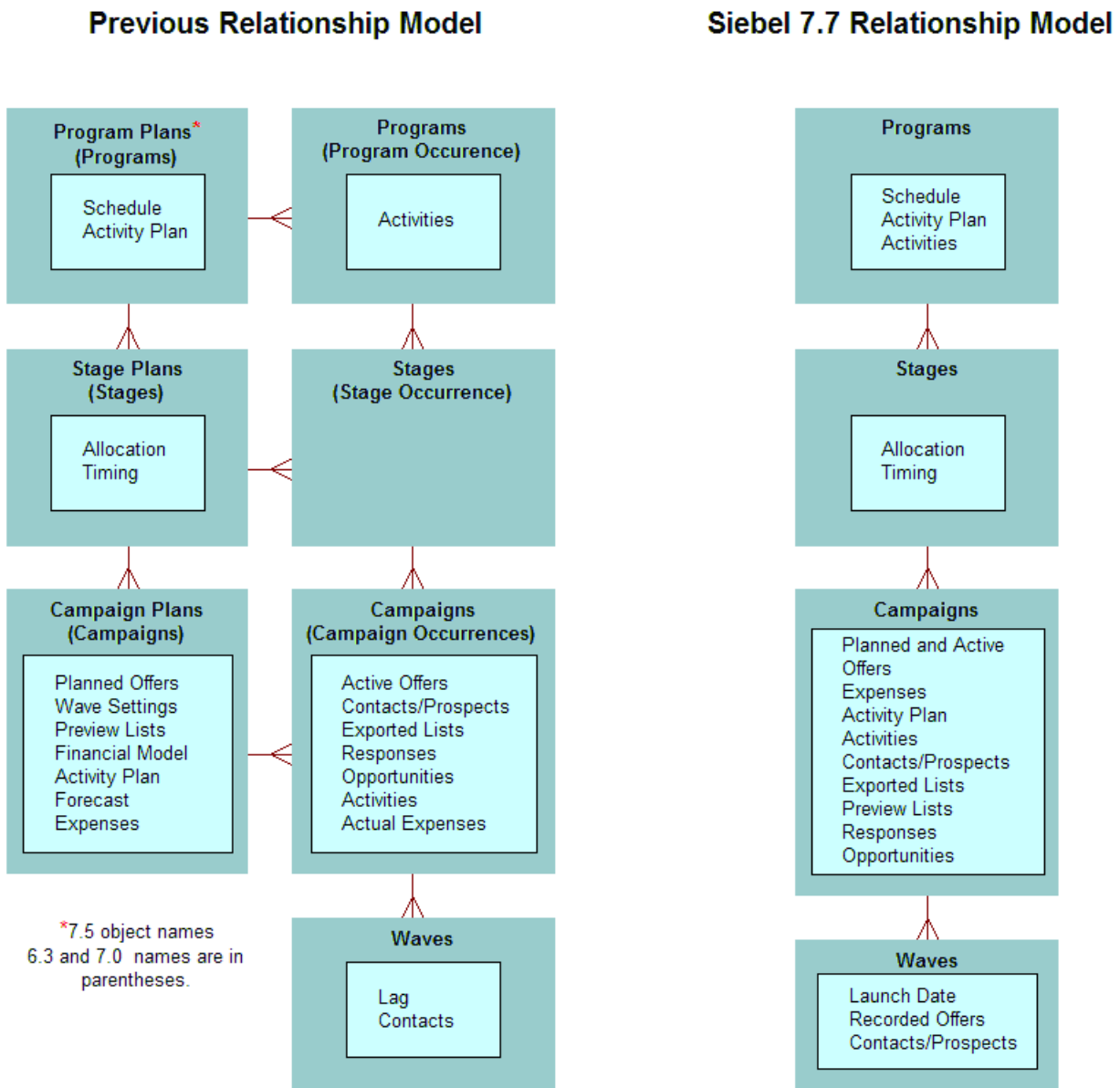


Figure 8. Relationship Model for Siebel Marketing

Merged Campaign Business Objects

In previous releases, campaign functions were handled by two business objects—DBM Campaign and Campaign. These objects shared similar functions, but many related elements such as views and child objects were not shared. The upgrade merges all campaign functions and views into a single business object—the Campaign business object.

During the upgrade, the following changes are performed to merge campaign plans and campaigns:

- Campaign Plans from the previous release are converted into Campaigns.
- The campaign occurrences that were children of the Campaign Plan are made obsolete.
- A new Load record is created for each previous occurrence.

Any child objects that were related to the obsolete campaign occurrence are re-parented to the surviving campaign.

Marketing Server Migrated to Oracle Business Intelligence Enterprise Edition

The introduction of Siebel 7.7 makes obsolete the Data Dictionary Manager and Marketing Server components in the Siebel Server. In previous releases, the Marketing Server component group performed segmentation and list export. Segmentation and list export are handled by subsystems in the Oracle Business Intelligence Web Presentation Services server.

Due to the obsolete server components, a set of business components and other objects are also made obsolete. These objects remain in the repository but are no longer used by the application.

The upgrade does not obsolete the List Import Service Manager, Marketing AOM (SMObjMgr), eMarketing AOM (eMarketObjMgr), or the eEvents AOM (eEventsObjMgr). They are actively used in the application.

[Appendix A, “Siebel Marketing Upgrade Reference”](#) lists the following types of obsolete items:

- Server components
- Business objects and business components
- Screens and views
- Data that is obsolete

About Siebel Marketing Customizations

Siebel 7.7 introduced a feature, Incorporate Custom Layouts (ICL). When you select this feature in the Upgrade Wizard, customized view and applet layouts are preserved, consistent with new features and schema changes.

Because of the large number of user interface changes in Siebel Marketing, the views and applets for Siebel Marketing have been excluded from ICL support. (The Upgrade Behavior property for Siebel Marketing objects in the Siebel Tools Repository is set to ADMIN.)

This means that the Repository upgrade will not preserve customizations to Siebel Marketing views and applets. If you select ICL mode for your upgrade, then other views and applets (except those for Siebel Marketing) will be preserved under ICL.

Summary of Database Changes for Siebel Marketing

Table 19 on page 149 summarizes Siebel Database changes that apply to Siebel Marketing.

Table 19. Summary of Database Changes for Siebel Marketing for Siebel 7.7

Upgrade Step	Description
Updates S_SRC and S_SRC_REL	Updates marketing plan tactics to use the S_SRC.MKTG_PLAN_ID to indicate the parent marketing plan, rather than S_SRC_REL.
Updates S_SRC	<p>Updates Status values (STATUS_CD) for campaigns and programs.</p> <p>Updates campaign plans and stand-alone campaigns to use the same CAMP_TYPE_CD.</p> <p>Sets CAMP_TYPE_CD to Campaign when SUB_TYPE is Marketing_Campaign and other conditions specified in upgrade scripts are met.</p> <p>Updates the values for S_SRC.RESPNSE_TYPE.</p>
Inserts into S_DD_CAMP_WAVE and S_CAMP_LD_WAVE	<p>Creates a Load record for each campaign occurrence. Load numbers are assigned based on date of execution.</p> <p>Creates one load wave for each stand-alone campaign.</p> <p>Sets the status of the load wave to the corresponding execution status from the original campaign occurrence.</p>
Inserts into S_CAMP_WAVE_DCP	Records the offer history for load wave records (pre-upgrade occurrences).
Index change and update to S_CAMP_CON	<p>Modifies the user key to use Campaign ID + Contact ID + Load Number + Token Number.</p> <p>When campaign occurrences are converted to Load records, S_CAMP_CON is updated to the appropriate Load number for the campaign contact.</p>
Updates S_SRC_COST	Updates TYPE_CD to the new values for Fixed and Per Unit expenses.
Updates S_SRC_GOAL	Updates goal type codes (GOAL_TYPE_CD).
Updates S_EVT_ACT	<p>Re-parents activities from campaign occurrences to campaigns.</p> <p>Re-parents activities from program occurrences to programs.</p>

Table 19. Summary of Database Changes for Siebel Marketing for Siebel 7.7

Upgrade Step	Description
Updates S_SRC for Stages	Updates CAMP_TYPE_CD indicates "STAGE" for stage records.
Updates S_OPTY_SRC	Re-parents opportunities from campaign occurrences to campaigns.
Updates S_SRC_DCP	Re-parents offers from campaign occurrences to campaigns.
Updates S_ORDER	Re-parents orders from campaign occurrences to campaigns.
Updates S_CS_PATH_SRC	Re-parents SmartScripts from campaign occurrences to campaigns.
Updates S_SRC_POSTN	Re-parents team members from campaign occurrences to campaigns.
Updates S_COMMUNICATION	Re-parents responses from campaign occurrences to campaigns.
Updates S_CAMP_SKILL	Re-parents Assignment Skills from campaign occurrences to campaigns.
Updates S_QTA_PLAN	Re-parents Quota Plans from campaign occurrences to campaigns.
Updates S_PROD_INT_SRC and S_PROD_LN_SRC	Re-parents related products and product lines from campaign occurrences to campaigns.
Updates S_CALL_LST	Re-parents internal lists from campaign occurrences to campaigns.
Inserts into S_SRC_GOAL (for Campaign Plan)	Updates actual and forecast values for Goals based on Financial Modeler columns in S_SRC (Revenue, #Leads, Response Rate, Conversion Rate, Avg. Revenue per Sale, Avg. Contribution Margin, ROI Amount).

Summary of How Marketing Data Will Be Migrated

Table 20 on page 151 describes how Siebel Marketing data will be migrated.

Data that is obsolete as of Siebel 7.7 is listed in [Appendix A, "Siebel Marketing Upgrade Reference."](#)

Table 20. Siebel Marketing Data Migration for Siebel 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
Marketing Plans	Preserved	N/A	N/A	
Program Plans	Preserved	N/A	N/A	
Stage Plans	Preserved	N/A	N/A	
Campaign Plans	Preserved	N/A	N/A	Campaign Plans are converted to Campaigns.
Campaign History	Preserved	N/A	N/A	Index on S_CAMP_CON is modified to preserve history for campaign and campaign load.
Stand-alone Campaigns (campaigns not related to a program)	Preserved	N/A	N/A	
Response History	Preserved	N/A	N/A	
Lists	Preserved	N/A	N/A	
Offers (except eNewsletter offers)	All offer types preserved, except for eNewsletters	See Obsolete Objects table	See Obsolete Objects table	
Segments	Usage history in programs is preserved Segment criteria are obsolete	S_CALL_LST_CRIT S_CALL_LST_DTL S_CALL_LST_QRY	Segment Detail-DD Segment Expression Campaign Segment Allocation Segment Campaign Allocation	Segments should be reconstructed in the Siebel 7.7 Segment Designer.

Upgrade Planning for Siebel Workflow Designer

Platforms: Windows, UNIX, IBM z/OS.

Important changes were made for Siebel 7.7 and Siebel 8.0 to how workflows are upgraded. Use this topic to determine how these changes affect your upgrade.

Changes for Siebel 7.7

In Siebel 7.7, the Workflow Process Designer was moved to Siebel Tools. Workflow components and definitions are defined as Siebel Tools objects and are stored in the repository.

The upgrade script migrates workflows from the main Siebel Database to the repository. Before upgrading your development or test environments, make sure that they include all the workflows from your production environment.

The upgrade copies or moves all workflow definitions to the repository as follows:

- **Seed workflows.** The old seed workflows are overwritten by new seed workflows. Their status is Inactive.
- **Customer workflows, status Inactive.** These are converted to workflow definitions in the repository. They will not have a status.
- **Customer workflows, status In-Progress.** These are converted to workflow definitions in the repository. Their status remains In-Progress.
- **Customer workflows, status Active.** These are converted to workflow definitions in the repository. Their status is changed to Completed. These workflow definitions are not copied to the main Siebel Database. This means that after the upgrade, no workflows are deployed. You must manually deploy seed workflows and customer workflows after the upgrade.

The upgrade automatically migrates customized business service scripts that are called by workflow operations.

You must deploy and activate repository workflows in order to use them. Workflow policy object and policy program data is upgraded normally. No data is changed or lost. Database triggers are not upgraded. After the upgrade, you must regenerate database triggers.

For information on how to deploy workflows, see *Siebel Business Process Framework: Workflow Guide*.

Schema Changes

The main Siebel Database tables that contain workflow definitions have changed. The new tables contain the workflow definitions for deployed workflows. The definitions of workflows that are Inactive or In-Progress are located in the repository. The tables are named as follows:

- Siebel Database tables containing workflow information begin with S_WFA.
- Repository tables containing workflow information begin with S_WFR.

- Siebel Database tables that contain workflow information for releases prior to Siebel 7.7 begin S_WF_ (note the underscore after WF). After the upgrade to Siebel 7.7, these tables are obsolete and are not referenced by applications.

Changes for Siebel CRM 8.x

As of Siebel CRM 8.x, the repository merge process preserves customizations you have made to seeded workflows. A seeded workflow is a Workflow Process object and all child objects shipped as part of a Siebel release.

Special pre-merge and postmerge steps have been added to the regular repository merge to allow merging customer-modified seeded workflows:

- 1 Workflow pre-merge.** When you start a repository merge, this step runs and prepares customer-modified seeded workflows in the Prior Customer Repository for the repository merge.
- 2 Repository merge.** After the workflow pre-merge completes, the main repository merge runs. The repository merge identifies modifications you have made to seeded workflows in the Prior Customer Repository. It then copies the modifications to the corresponding seeded workflows in the New Customer Repository.
- 3 Workflow postmerge.** After the main repository merge completes, the workflow postmerge runs and sets workflow versions in the New Customer Repository.

All three parts of the repository merge display in the status bar of the Merge Repositories dialog box when you perform a repository merge.

The workflow merge steps fit into the overall flow of a repository merge as follows:

- Repository merge starts
- Workflow pre-merge
- Main repository merge
- Workflow post-merge
- ICL merge steps (If this is an ICL merge)
- Merge ends
- Run the postmerge utilities

Workflow Pre-Merge

When you start the repository merge, the workflow pre-merge step runs and prepares customer-modified seeded workflows for the repository merge.

If the workflow exists in both the Prior Customer Repository and New Siebel Repository, and the version number in the Prior Customer Repository is greater than 0, then the workflow is customer-modified.

For these workflows, the pre-merge process makes the following changes:

- Deletes version 0 of the customer-modified seeded workflow in the Prior Customer Repository.

- In the Prior Customer Repository, copies the most recent version of the customer-modified seeded workflow and sets the copy's version to 0.

The most recent version of the workflow is the one with the highest version number and Status indicates Completed. This is called the version n workflow.

- Sets the copy's status indicates Completed.

The workflow pre-merge creates a version 0 copy of version n in the Prior Customer Repository because the merge process requires that workflows it compares between repositories must have the same name, must be version 0, and must have Status indicates Completed.

Repository Merge

The main repository merge compares workflows in the Prior Customer Repository, Prior Siebel Repository, and the New Siebel Repository:

- **Customer-Modified seeded workflows.** The repository merge copies the modifications to version 0 of the workflow to the Prior Customer Repository to the same workflow in the New Customer Repository. The merge also copies versions 1 through n of the workflow to the New Customer Repository.

An exception is when a workflow's object attributes are different in all three compared repositories. This means the object is both customer-modified and has also changed in the new release. This causes a merge conflict.

The repository merge handles workflow attribute conflicts in the same manner as other objects. The merge process typically resolves merge conflicts in favor of the New Siebel Repository. You can review workflow-related attribute conflicts in the Siebel Tools Application Upgrade attribute List screen.

If you modify a seeded workflow by deleting any of its child objects, the merge process does not delete the child objects in the New Customer Repository. After the merge, you must review child objects and delete them as desired.

- **Unmodified seeded workflows.** If the workflow exists in both the Prior Customer Repository and New Siebel Repository, and the highest version level is 0 in the Prior Customer Repository, then the workflow is unmodified.

After the merge, the seeded workflow in the New Customer Repository will be the one that shipped with the new release.

- **Customer-Created workflows.** If the workflow exists in the Prior Customer Repository but not the Prior Siebel Repository, the workflow is customer-created. The repository merge copies all versions of customer-created workflows to the New Customer Repository.

- **Customer-Deleted Workflows.** If the workflow exists in the Prior Siebel Repository but not the Prior Customer Repository, the workflow is customer-deleted. The repository merge does not delete the workflow from the New Customer Repository. After the merge, you must review these workflows and delete them as desired.

- **Obsolete seeded workflows.** If the workflow exists in the Prior Siebel Repository but not the New Siebel Repository, the workflow is obsolete. The repository merge does not copy obsolete seeded workflows from the Prior Customer Repository to the New Customer Repository unless they are customer-modified. After the merge, you must review customer-modified, obsolete workflows and delete them as desired.

Workflow Postmerge Step

The workflow postmerge step runs after the main repository merge completes and does the following for seeded workflows that are customer-modified:

- Changes the version of the workflow from 0 to $n + 1$ and sets the $n+1$ version's workflow status to In Progress.
For example, if the most recent version of customer-modified Workflow A in the Prior Customer Repository is version 3, then the merged version in the New Customer Repository will be version 4 and will have Status indicates In Progress.
- Copies version 0 of the workflow from the New Siebel Repository to the New Customer repository. This reinstates the version 0 seeded workflow that shipped with the new release.

Status of Customer-Modified Workflows After the Merge

After the whole repository merge is complete, customer-modified seeded workflows appear as follows in the New Customer Repository:

- **Version 0.** This is the seeded workflow that shipped with the new release.
- **Versions 1 through n .** These versions are copied intact from the Prior Customer Repository.
- **Version $n+1$.** This is the new merged workflow version. It is a combination of the seeded workflow that shipped with the new release and the modifications contained in version n in the Prior Customer Repository.

Logging

Workflow pre-merge and postmerge. The workflow pre-merge step and postmerge steps write to the same log file:

```
Tools_install_dir\bin\merge0_ver.txt
```

where:

Tools_install_dir is the directory in which Siebel Tools is installed.

Each time you run the merge process, the name of the merge0_ver.txt file is incremented, for example merge1_ver.txt.

If the log file contains the error IDS_ERR_DEV_MRG_PREMERGE_FAILED, this means that the pre-merge step could not delete one or more version 0 seeded workflows. This causes the main merge process to be unable to merge the customer-modified seeded workflow into the New Customer Repository. You must manually reimplement customizations to these workflows. You need not rerun the repository merge.

If the log file contains postmerge errors, create a service request (SR) on Oracle*Metalink* 3, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Main repository merge. The main repository merge logs workflow-related messages to the standard merge log file, merge0.txt. If you find workflow-related errors in this log file, create a service request (SR) on Oracle*Metalink* 3, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

The main repository merge log file is located in the same directory as the workflow pre-merge/postmerge log file.

Upgrade Planning for Handheld Devices in the Siebel Environment

Upgrades from: Siebel 7.0.x and 7.5.x.

Platforms: Windows, UNIX, IBM z/OS.

Device operating system support, third-party product support, and application configuration management for handheld devices were revised in Siebel 7.7. Before the upgrade, verify that handheld devices are running an OS supported by Siebel 7.7. Also, verify that third-party software is the correct version.

After the upgrade, you must enter any handheld device-related application configuration changes into the handheld device administration screen.

Supported Device OS

The 2002 version of the Pocket PC operating system is no longer supported as of Siebel 7.7. Devices running Pocket PC 2002 must be upgraded to Pocket PC 2003. For a full description of supported handheld devices and operating systems, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

Active Sync Support

The third-party product Active Sync must be at version 3.7.1 or higher as of Siebel 7.7. For a complete description of supported third-party products, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

Handheld Device Administration

Siebel 7.7 introduced a handheld device administration screen. If you have modified the PDA portion of the Siebel Sales, Siebel Service, Siebel ePharma, or Siebel eConsumer application .cfg file, do the following before upgrading:

- Save the modified .cfg file under a new name so that it is not overwritten during the upgrade.
- After the upgrade, go to the Administration—Mobile screen and enter the changes from the saved .cfg file. For information on using this screen, refer to the Handheld guide for your application.

Handheld Application Upgrade

Handheld applications do not upgrade automatically. Users must uninstall the application and install the current release.

Upgrade Planning for Resonate Central Dispatch in the Siebel Environment

Upgrades from: Siebel 7.0.x and 7.5.x.

Platforms: Windows, UNIX, IBM z/OS.

Support for Resonate Central Dispatch is discontinued as of Siebel 7.7. It has been replaced by a load balancing module that is included in the Siebel Web Server Extension. In addition, Oracle has certified several third-party HTTP load balancers for use with the Siebel Web Server Extension.

For a description of the load balancing module, see *Deployment Planning Guide*. For a list of supported HTTP load balancers, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

For information on configuring load balancing, see *Siebel Installation Guide* for the operating system you are using.

Upgrade Planning for Siebel String Translation

Upgrades from: Siebel 7.0.x and 7.5.x.

Platforms: Windows, UNIX, IBM z/OS.

Siebel 7.7 introduced a new method of generating data-migration SQL. Based on the language for the Siebel upgrade process, the Siebel upgrade scripts are created accordingly.

NOTE: To avoid unintended results, you must perform your upgrade in the same language as that of the base language in the prior release.

When the Database Configuration Wizard is started, the sqlgen utility generates the Siebel upgrade scripts based on the information located in the upgfile.xml file. When the Database Configuration Wizard completes, SQL scripts are generated, and replace the placeholders in the upgfile.xml with the language-specific values located in the dbsrvr/<LANG>/upglocale.<LANG> file.

In the following example, the locale file is upglocale.fra, for a FRA based installation. The upgfile.xml contains the following statement:

```
update S_DOC_QUOTE
set QUOTE_SUB_TYPE_CD = &apos; <XTL_STRING>Private</XTL_STRING>&apos;;
where QUOTE_TYPE = &apos; <XTL_STRING>Template</XTL_STRING>&apos;;
```

The placeholders `<XTL_STRING>Private</XTL_STRING>` and `' <XTL_STRING>Template</XTL_STRING>` are values that the `sqlgen` utility changes during the upgrade. The `XTL_STRING` values are replaced based on what the `sqlgen` utility locates as a match for the *Private* and *Template* strings in the `upglocale.fra` file, located under `dbsrvr/FRA`. For this example, the FRA base language dictates *Template = Modèle* and *Private = Privé*.

CAUTION: If you modified string mappings in your previous Siebel release, you must contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services in order to modify standard upgrade scripts accordingly. If you do not, your Siebel application data from the prior release may not be migrated to the new Siebel release's data model.

Upgrade Planning for Siebel Configurator

Upgrades from: Siebel 7.0.x and 7.5.x.

At Siebel 7.7, you can set a user property in a Siebel Configurator model that causes configuration sessions to run in high interactivity mode. This feature improves performance and is intended for use with Oracle's Siebel applications that are typically deployed in high interactivity mode, such as employee applications.

For information on how to deploy Siebel Configurator models in high interactivity mode, see Bulletin 477610.1 (Doc ID) on Oracle *Metalink* 3. This document was previously published as Siebel Technical Note 500.

Browsers that will host high interactivity applications must meet certain requirements. These are described in the *Siebel System Requirements and Supported Platforms* on Oracle Technology Network. For more information on high interactivity mode, see the *Deployment Planning Guide*. If you intend to deploy Siebel Configurator models in high interactivity mode, include verifying that users have appropriate browsers in your upgrade planning.

Upgrade Planning for Siebel Personalization

Upgrades from: Siebel 7.0.x and 7.5.x.

At Siebel 7.7 access control based on primary user role has been replaced by access based on primary responsibility name. If you have defined conditional expressions for applets in the Administration-Personalization screen, plan to review these after the upgrade and verify that they use Primary Responsibility name. For more information on access control, see *Security Guide for Siebel Business Applications*. For more information on personalization management, see *Siebel Personalization Administration Guide*.

Upgrade Planning for Siebel Pricer and Order Management

Upgrades from: Siebel 7.0.x, 7.5.x, and 7.7.x.

Siebel 7.8 introduces a new architecture for managing pricing. Data associated with the Pricer features will be upgraded as shown in [Table 21](#).

Table 21. How Pricer Features Are Upgraded

Feature	How Upgraded
Price lists and cost lists	Automatically upgraded as part of price list
Customizable product pricing	Automatically upgraded as part of price list
Service pricing (based on covered product)	Automatically upgraded as part of price list
Volume discount	Automatically upgraded
Attribute pricing	Automatically upgraded as attribute adjustment
Pricing model	Not automatically upgraded. Must be redesigned and reimplemented as pricing procedures.
<ul style="list-style-type: none"> ■ Aggregate factor ■ Bundle factor ■ Single factor ■ Matrix factor ■ Script based 	Bundle factor definitions are upgraded to aggregate discounts and aggregate discount sequences.

Pricer API user properties for internal or external application integration are also obsolete. After the upgrade, you must reimplement integrations using pricing procedures, signals, variable maps and other features of the order management infrastructure.

For information on the new Pricer architecture, on application integration, and on order management infrastructure, see *Pricing Administration Guide* and *Siebel Order Management Infrastructure Guide*.

For a scenario that describes how to reimplement Siebel 7.x pricing models, see Bulletin 473908.1 (Doc ID) on [OracleMetalink](#) 3. This document was previously published as Siebel Technical Note 639.

Upgrade Planning: Additional Siebel Application Changes

Upgrades from: All Supported Siebel releases.

Table 22 displays additional application changes. Review these changes and make needed revisions before beginning the upgrade.

Table 22. Additional Application Changes

Applications or Views Affected	Applies to Upgrades from	Description
Siebel Service	Siebel 7.7 and earlier	In Siebel 7.8 the Siebel Service application repository object is set to inactive (Inactive property = True). In previous releases, this object was active.
Applications using the table S_PRSP_CONTACT	Siebel 7.7 and earlier	In Siebel 7.8, the column S_PRSP_CONTACT.OU_ID is used by Oracle's Siebel applications. In previous releases, this column was unused. If you have stored data in this column, move it to another column before beginning the upgrade.
eCalendar Detail View	Siebel 7.5 and earlier	In Siebel 7.7, the HI Gantt Chart Applet is no longer used. It is replaced by the Calendar Gantt Chart AX Applet.

Bulletin 477759.1 (Doc ID) on Oracle *Metalink* 3 (previously published as Siebel Technical Note 511) lists application objects in the Siebel Repository that are obsolete. Recommended actions are listed for each object, if applicable. Before upgrading, review this list and determine if any of your customizations are using objects that obsolete as of the new Release.

8

Basic Database Preparations for a Siebel Upgrade

This chapter provides initial preparatory information for the Siebel Database upgrade. This chapter includes the following topics:

- ["Upgrade Flow Orientation" on page 162](#)
- ["Verifying Siebel Database Connectivity" on page 162](#)
- ["Preparing Siebel Tables and Views for Upgrade" on page 163](#)
- ["Preparing Siebel Custom Indexes for Upgrade" on page 164](#)
- ["Exporting Siebel Interface Table Data" on page 164](#)
- ["Archiving Unneeded Siebel Repositories" on page 165](#)
- ["Preserving Siebel Dock Objects and Visibility Rules" on page 165](#)
- ["Securing AIX Memory Allocation Segment Space for the Siebel Database" on page 166](#)
- ["Preparing for a Multilingual Upgrade" on page 166](#)

Related Topics

[Chapter 3, "Simplified Overview of Siebel Database Upgrade Process"](#)

[Chapter 5, "How the Siebel Database Upgrade Works"](#)

[Chapter 4, "How to Perform a Siebel Database Upgrade"](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase One of the Siebel Database upgrade flow, as shown in Figure 9. At this phase of the upgrade you are performing basic tasks which will prepare your Siebel Database for the upgrade.

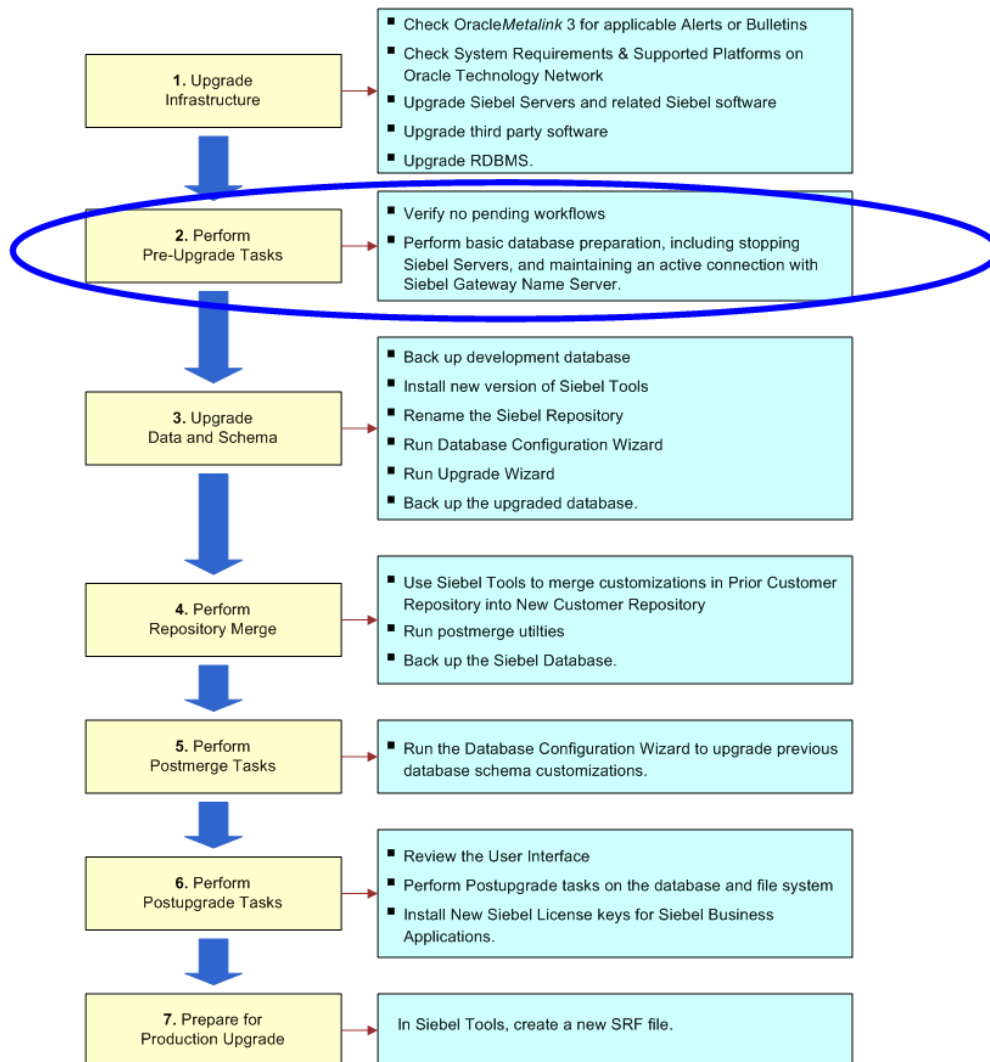


Figure 9. Phase Two of Upgrade Flow

Verifying Siebel Database Connectivity

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

From the production test environment, you must be able to make ODBC connections to both the Siebel Database in the development environment and the Siebel Database in the production environment. Verify that you can define these ODBC connections in the production test environment.

If you cannot connect to these databases from the production test environment, create a service request (SR) on Oracle *Metalink* 3, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

In the production environment, you do not have to define an ODBC connection to the development environment Siebel Database or the production test environment Siebel Database.

Preparing Siebel Tables and Views for Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

To prepare tables and views for upgrade

- 1 Drop temporary tables and non-Siebel tables.

If the upgrade process detects a column with a datatype not acceptable to Siebel tables, the upgrade will fail.

- 2 Disable customized triggers.

You must re-create them after the upgrade.

- 3 Drop defined database views on Siebel tables.

You must re-create them after the upgrade.

- 4 Export interface table data that you want to preserve.

Interface tables are dropped and then re-created during upgrade. You can import the data after the upgrade.

Preparing Siebel Custom Indexes for Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Consider the following guidelines when preparing custom indexes for upgrade:

- **Custom indexes against extension columns on obsolete tables.** If you have created custom indexes that use extension columns on obsolete tables, you must migrate the data to new extension columns before upgrading the Siebel Database. For assistance, create a service request (SR) on Oracle *Metalink* 3, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- **Custom indexes that were not defined through Siebel Tools.** Custom indexes created without using Siebel Tools are not included in the schema definition in the Siebel Repository. These indexes are dropped during the database upgrade. To preserve these indexes, add them to the Siebel Repository using Siebel Tools.
- **Custom indexes on interface tables.** Custom indexes on interface tables are not re-created during the upgrade. You must re-create them after the upgrade is complete.
- **Custom indexes on base tables.** The Siebel 7.x upgrade automatically removes and re-creates custom indexes on base tables.
- **Custom indexes may need to be changed to reflect schema changes.** Reevaluate custom indexes for applicability in the new release. They may no longer be needed due to schema changes in the new release.

For more information about custom indexes, see *Configuring Siebel Business Applications*. For information on schema changes in a release, see *Siebel Data Model Reference*.

Exporting Siebel Interface Table Data

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

During the upgrade process, your interface tables are dropped and then re-created. To retain data in your interface tables, use the appropriate tools for your RDBMS to export data before the upgrade and then import the data after you have completed the upgrade.

During the upgrade, all custom indexes on interface tables are dropped from both logical and physical schema.

Archiving Unneeded Siebel Repositories

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Perform this task before running the Database Server Configuration Utilities in upgrep mode for the first time in an environment.

The upgrep mode scripts expect only the Siebel Repository to be present. If you have multiple repositories, you must export them to archive files and then delete them from the database.

Because the upgrade changes the schema of the database, in most cases you cannot import these archived repositories into the upgraded database. If you want to access the archived repositories, you must import them into a database that has the same schema as the one from which they were exported.

To archive unneeded repositories

- 1 Export all repositories.
- 2 Place exported repository files in a safe location.
- 3 In the Siebel Database, delete all repositories except the Siebel Repository.

For information on exporting and deleting repositories, see the Managing Repositories chapter in *Using Siebel Tools*.

Preserving Siebel Dock Objects and Visibility Rules

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Modified visibility rules are dropped during a development environment upgrade. Manually record any changes to dock object visibility rules, so you can evaluate whether you must reapply the changes after the upgrade is complete.

Dock objects and visibility rules created by using Docking Wizard are preserved unless they become invalid after the upgrade. Manually record any changes that you made through the Docking Wizard so that you can evaluate whether you need to reapply the changes after the upgrade is complete.

Changing the definition of dock objects requires the assistance of Oracle Advanced Customer Services. Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Securing AIX Memory Allocation Segment Space for the Siebel Database

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases. (Exception: this topic does not apply to Microsoft SQL Server.)

Platforms: UNIX only.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

Before you run an upgrade on AIX, set the following environment variable on the AIX machine that you are using for the upgrade:

```
setenv LDR_CNTRL LOADPUBLI C@MAXDATA=0x60000000
```

This will prevent a shortage of memory allocation segment space that might occur on the machine where both the Siebel Database Server and Siebel Server are installed. After a successful upgrade, reset this parameter to the original value.

Preparing for a Multilingual Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

As of Siebel CRM 8.0, the upgrade process upgrades both the primary (base) language and all deployed languages.

The Database Server Configuration Utilities validate deployed languages by comparing the status of the language packs installed on the Siebel File System with the language IDs of records in the S_LST_OF_VAL table. If there are Language IDs in S_LST_OF_VAL but no corresponding deployed language on the Siebel File System, the validation fails and you cannot continue the upgrade.

Before doing an upgrep, use Siebel Tools to remove or deactivate any records in S_LST_OF_VAL for undeployed languages.

To prepare for a multilingual upgrade

- 1 Make a list of your deployed languages.

A *deployed language* is one where you have installed the language pack, imported the UI strings into the Siebel Repository, imported the seed data into the database, activated the seed data records added to S_LST_OF_VAL, and configured the records for MLOVs.

- 2 In Siebel Tools, navigate to Screens > System Administration > List of Values.

- 3 Locate the Language Name field, and verify that no languages appear that are not currently deployed.

Even if you do not have English installed, some records will have Language Name = English-American. You can ignore these system records.

- 4 For records where you do not have the corresponding language pack deployed, investigate why these records are present and take one of the following actions:

- If the record is not needed, delete it.
- If you are not sure whether the record is needed, remove the check-mark from the Active field. This deactivates the record and prevents it from being included in the validation check performed by the Database Server Configuration Utilities.

CAUTION: Do not deactivate or delete records with Language Name = English-American, even if you do not have English-American deployed. These records are needed by the system. The validation process ignores these records.

Related Topics

[“Upgrade Planning for Multilingual Siebel Deployments” on page 134](#)

9

Preparing an IBM DB2 Database for a Siebel Upgrade

This chapter provides initial preparatory information for a Siebel Database upgrade on the IBM DB2 Database. This chapter includes the following topics:

- [“Verifying the IBM DB2 Client for a Siebel Upgrade” on page 169](#)
- [“Verifying IBM DB2 Sort Order for a Siebel Upgrade” on page 170](#)
- [“Setting IBM DB2 Parameters for a Siebel Upgrade” on page 170](#)
- [“Verifying IBM DB2 Permissions for a Siebel Upgrade” on page 172](#)
- [“Verifying IBM DB2 Instance Owner Permissions for a Siebel Upgrade” on page 173](#)
- [“Creating IBM DB2 Temporary Tablespaces and Bufferpools for a Siebel Upgrade” on page 173](#)
- [“Analyzing IBM DB2 Custom Tablespace Requirements for a Siebel Upgrade” on page 174](#)
- [“Verifying the DB2 Application Development Client for a Siebel Upgrade” on page 176](#)
- [“Identifying IBM DB2 Long Columns for Truncation in a Siebel Upgrade” on page 177](#)

Related Topics

[Chapter 5, “How the Siebel Database Upgrade Works”](#)

[Chapter 4, “How to Perform a Siebel Database Upgrade”](#)

Verifying the IBM DB2 Client for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: IBM DB2 only. This topic applies only to the 64 bit DB2 RDBMS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The Siebel Server supports only the 32-bit DB2 client. Verify that you have not installed the 64-bit DB2 client on the Siebel Servers. If you have installed the 64-bit DB2 client, replace it with the 32-bit client.

IBM supports the 32-bit DB2 client working with the 64-bit DB2 8.x and 9.x RDBMS.

Verifying IBM DB2 Sort Order for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: IBM DB2 only.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

Binary sort order is required for the development environment upgrade and is strongly recommended for the production environment upgrades.

Sort order is specified during creation of the database. If you find that your IBM DB2 development database was not created using Identity sort order, you must re-create your database using the option `COLLATE USING IDENTITY`.

If sort order is correct, but you are still encountering errors, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services to help in further analysis.

To verify that your database was created using Identity sort order

1 Run the following query on Siebel Database:

```
select count (*) from SIEBEL.S_APP_VER where '$' > '/'
```

2 Review the result.

■ If sort order is correct, the result is

```
1
-----
0
(1) record selected.
```

■ If sort order is incorrect, you must re-create the database, using this option:

```
COLLATE USING IDENTITY
```

Related Topics

["About Database Sort Order in the Siebel Environment" on page 129](#)

Setting IBM DB2 Parameters for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: IBM DB2 only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Before upgrading an IBM DB2 database, verify that your database server meets or exceeds the following configuration criteria:

- The DMS tablespace has at least 25% of free pages.
- The file system has sufficient space to allow your DMS tablespace to grow.
- Siebel tablespaces for the DB2 platform should be database-managed tablespaces (DMS) rather than system-managed tablespaces (SMS).
- Verify that the tablespaces are not near their capacity. This can be done by connecting to the database and issuing the following command:

```
db2 list tablespaces show detail
```

The tables which follow provide upgrade-specific settings for the Database Manager and database. Use the following strategy to set parameters:

- Set parameters using the recommendations in *Siebel Installation Guide* for the operating system you are using. Recommendations are located in the chapter on configuring the RDBMS.
- For the upgrade, revise the configuration parameters listed in the tables which follow.
- After the upgrade, reset the configuration parameters to the values listed in *Siebel Installation Guide* for the operating system you are using.

DB2 Database Manager Settings

The Upgrade Setting column in [Table 23](#) provides guidelines for setting configuration parameters specifically to optimize upgrade performance. Set these parameters for each DB2 instance.

Table 23. DB2 Database Manager Configuration Parameters

Parameter	Explanation	Upgrade Setting
SHEAPTHRES	Sort heap threshold (4 KB) If you reset SHEAPTHRES or SORTHEAP, rebinding the instance is recommended.	Double the value allocated for SORTHEAP. See Table 24 on page 172 .

DB2 Database Configuration Parameters

The Upgrade Setting column in [Table 24](#) provides guidelines for setting configuration parameters specifically to optimize upgrade performance. Set these parameters for each DB2 instance.

Table 24. DB2 Database Configuration Parameters

Parameter	Explanation	Upgrade Setting
SORTHEAP	Sort list heap (4 KB)	20000–40000 Recommended size; this may increase or decrease depending on the amount of memory in the database server machine and the size of the data. A 20000 setting allows SORTHEAP to increase up to 80 MB.
MAXLOCKS	Percentage of lock lists per application	5
CHNGPGS_THRESH	Changed pages threshold	5
LOGRETAI N	Sequential or circular log files	NO Setting this parameter to YES means that log files are archived. You must periodically move or archive the log files to prevent the file system containing the log files from filling up.
LOGFILSI Z	Log file size (4 KB)	Development environments: 8000–16000
SOFTMAX	Triggers bufferpool flushing	50

Verifying IBM DB2 Permissions for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: IBM DB2 only.

Platforms: UNIX only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

If you are running IBM DB2 on the AIX or Solaris platform, perform the following steps before executing the Siebel Database upgrade.

To verify DB2 permissions

- 1 Navigate to the instance home directory.
- 2 Use the following command to verify that the directory `sql lib/function/routine/sql proc` has write permission for the group:

```
ls -ld sql lib/function/routine/sql proc
```

- 3 To authorize group write permission, enter the following command:

```
chmod g+w sql lib/function/routine/sql proc
```

Verifying IBM DB2 Instance Owner Permissions for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: IBM DB2 only.

Platforms: UNIX only.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

If you are running IBM DB2 on the AIX or Solaris platforms, verify that the Siebel Database instance owner belongs to the primary group of the fenced user. If the instance owner is not part of this group, errors will occur during the Siebel Database upgrade.

Creating IBM DB2 Temporary Tablespaces and Bufferpools for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: IBM DB2 only.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

If your RDBMS is IBM DB2, verify that you have 16-KB and 32-KB temporary tablespaces to use for sorting and other SQL processing. Both the 16-KB and 32-KB temporary tablespaces require dedicated bufferpools.

To create a 16-KB temporary tablespace

- 1 Create a 16-KB bufferpool with at least 5000 16-KB pages.

- 2 Create a 16-KB temporary tablespace as system managed space (SMS) that can be expanded to 2 GB of storage.

To create a 32-KB temporary tablespace

- 1 Create a 32-KB bufferpool with at least 1000 32-KB pages.
- 2 Create a 32-KB temporary tablespace as SMS that can be expanded to 2 GB of storage.

Analyzing IBM DB2 Custom Tablespace Requirements for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: IBM DB2 only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

There are four standard database managed tablespaces (DMS) that hold Siebel tables and indexes—a 4-KB, 16-KB, 32-KB tablespace, for tables, and a tablespace to hold indexes. The upgrade process moves tables between these spaces as required.

If you have placed Siebel tables in other tablespaces, the upgrade process will not move these tables if they grow to exceed the tablespace size during the upgrade. If one of these tables has an estimated page size after upgrade greater than its current page size, it will not fit in its tablespace after the upgrade, and the upgrade will fail.

Oracle provides a sizing utility that determines whether tables will increase in size to the point that they must be moved to a larger tablespace.

Run the utility before upgrading the database. If the sizing utility reports any problems, you must resolve them before you proceed with the upgrade.

To analyze tablespace requirements for DB2

- 1 Navigate to the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Type the following command line:

```
tblsize /U TABLEOWNER /P PASSWORD /C ODBC_DATASOURCE /F DDL_FILE  
/B DEFAULT_TABLESPACE /X DEFAULT_INDEXSPACE /K 16K_TABLESPACE /N 32K_TABLESPACE  
/O REPORT_FILENAME /L LOG_FILENAME
```

where:

- `TABLEOWNER` = Tableowner

- *PASSWORD* = Tableowner password
- *ODBC_DATASOURCE* = Data source of the database
- *DDL_FILE* = Absolute path to the DDL file (This file is called *ddl.ctl*, and it is located in the *dbsrvr/db2udb* directory.)
- *DEFAULT_TABLESPACE* = Name of the 4-KB page standard Siebel tablespace
- *DEFAULT_INDEXSPACE* = Name of the standard Siebel index space
- *16K_TABLESPACE* = Name of the 16-KB page standard Siebel tablespace
- *32K_TABLESPACE* = Name of the 32-KB page standard Siebel tablespace
- *REPORT_FILENAME* = Name of the report generated by the utility
- *LOG_FILENAME* = Name of the log file (The default name is *custtbl.log*.)

Example:

```
tblsize /U siebel /P siebel /C ssa /F d: /sea77/dbsrvr/db2udb/ddl.ctl /B siebel_4k  
/X siebel_idx /K siebel_16k /V siebel_32k /Q d: /sea77/dbsrvr/db2udb/report.txt /  
L $SIEBEL_ROOT/log/tblsize.log
```

- 3 Review the report generated by the utility to determine if the estimated table pagesize postupgrade is larger than the size of the actual custom table pagesize.

An example of the report generated by this utility is displayed in the following example:

```
Table Name = S_EVT_ACT  
Custom Tablespace Id = 5  
Custom Tablespace Name = CUST_TBS_EVT_ACT  
Custom Tablespace Pagesize = 4096  
Estimated Table Pagesize (postupgrade) = 5067  
Status = Does not fit in its custom tablespace
```

- 4 For each table that has Status: Does not fit in its custom tablespace, you must create a larger custom tablespace that is larger than the estimated table pagesize postupgrade.

- 5 Move the tables from their old tablespaces to the new ones by running *ddl move*.

ddl move is a utility for moving tables from one tablespace to another tablespace. This utility is located in the following directory:

Windows: *SIEBEL_ROOT\bin*

UNIX: *\$SIEBEL_ROOT/bin*

- 6 To run *ddl move*, submit the following arguments:

```
ddl move /U TABLEOWNER /P TABLE_PASSWORD /C ODBC_DATASOURCE /E STOP_ON_DDL_ERROR  
/G GRANTEE /B TABLESPACE /X INDEX_TABLESPACE /M TABLE_NAME /L LOG_FILENAME  
/Z UCS2_DATABASE
```

where:

- *TABLEOWNER* = Tableowner of the database (Required)
- *TABLE_PASSWORD* = Password of the tableowner of the database (Required)
- *ODBC_DATASOURCE* = Data source of the database (Default environment variable: SIEBEL_DATA_SOURCE)
- *STOP_ON_DDL_ERROR* = Stop on DDL Error (Default: Y)
- *GRANTEE* = Grantee for tables
- *TABLESPACE* = Name of the tablespace that you are moving the table to
- *INDEX_TABLESPACE* = Name of the index space that you are moving the table to
- *TABLE_NAME* = Table Name Like Support (Default: N)
- *LOG_FILENAME* = Name of the log file (The default name is ddlmove.log.)
- *UCS2_DATABASE* = (Default: N)

Verifying the DB2 Application Development Client for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: IBM DB2 only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The IBM DB2 Application Development Client must be installed on the RDBMS server. [Table 25](#) lists the required DB2 Application Development Client components.

Table 25. DB2 Application Development Client Components

Platform	DB2 Application Development Client Components
Windows	DB2 Application Development Client
AIX	Application Development Tools (ADT) ADT Sample Programs
HP	Application Development Tools for HP-UX
Solaris	Application Development Tools (ADT) ADT Sample Programs

For information on installing the Application Development Client, refer to the relevant IBM documentation.

Identifying IBM DB2 Long Columns for Truncation in a Siebel Upgrade

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Databases: IBM DB2 only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

In Siebel 7.7, the maximum length for IBM DB2 long columns with a type of varchar was reduced to 16350 from 16383. The upgrade truncates long columns of type varchar that exceed 16,350.

To prevent a data truncation error that may cause transaction processing (txnproc) or transaction routing (txnroute) to fail, perform the following steps to identify these columns and reduce the data in these columns.

CAUTION: If you do not truncate the data in long varchar columns that exceed the maximum length specified in the following task, a “data truncated” error will occur, and transaction processing and transaction routing may fail.

To identify and reduce the length of long varchar columns

- 1 From any shell, open the script `chk16350.bat` (Windows) or `chk16350.ksh` (UNIX), and edit the following parameters as appropriate for your deployment:

`SRC_USR` = username of the source database

`SRC_PSWD` = password for the source database

`SRC_TBL0` = tableowner of the source database

`SRC_TBL0_PSWD` = tableowner password for the source database

`SRC_ODBC` = ODBC data source name of the source database (edit the value “CHANGE_ME”)

`SRC_REPOSITORY_NAME` = repository name of the source database

`DBSRVR_ROOT` = The directory where you installed the Siebel Database Server files on the Siebel Server. For example, `C:\sba80\dbsrvr` (Windows). Edit the value “CHANGE_ME”.

`SIEBEL_ROOT` = The directory where you installed the Siebel Server. For example, `C:\sba80\siebsrvr` (Windows). Edit the value “CHANGE_ME”.

`VALID_RESULTS_DIR` = directory where you want the output files to be generated (edit the value “CHANGE_ME”); this must be an existing directory

This script produces two files:

- **long_trunc_cols.rpt.** This report identifies all long varchar columns that are longer than 16,350 characters.
- **update_trunc.sql.** This SQL file generates update statements that truncate identified columns to 16,350 characters.

- 2 Reduce the data in these columns using either of the following methods:
 - Manually review the columns in the Long_trunc_cols.rpt report and manually reduce the size of each column identified.
 - Run update_trunc.sql using the DB2 command line processor.

10 Preparing an Oracle Database for a Siebel Upgrade

This chapter provides initial preparatory information for a Siebel Database upgrade on the Oracle Database. This chapter includes the following topics:

- [“Verifying Oracle Database Sort Order for a Siebel Upgrade” on page 179](#)
- [“Setting Oracle Database Configuration Parameters for a Siebel Upgrade” on page 180](#)
- [“Verifying Oracle Database Parameters for Multiple CPUs in a Siebel Upgrade” on page 181](#)
- [“Verifying the Oracle Database ODBC Definition for a Siebel Upgrade” on page 182](#)

Related Topics

[Chapter 5, “How the Siebel Database Upgrade Works”](#)

[Chapter 4, “How to Perform a Siebel Database Upgrade”](#)

Verifying Oracle Database Sort Order for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: Oracle only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The NLS_SORT parameter determines the sort order of query returns. NLS_SORT must be set to BINARY for the development environment upgrade, including the repository merge. It must also be set to BINARY when using Siebel Tools to compile an SRF file.

This setting is strongly recommended for production test environment upgrades and production upgrades.

To verify that your database is using binary sort order

- 1 Run SQLPlus to connect to the Oracle Database.
- 2 Issue the following query:

```
SQL> SELECT * FROM NLS_DATABASE_PARAMETERS;
```
- 3 In the returned parameters, locate NLS_SORT and verify that its value is BINARY.
 - If NLS_SORT has a value of BINARY, then the sort order is binary and no action is required.
 - If NLS_SORT is anything other than BINARY, reset the value to BINARY.

Related Topics

[“About Database Sort Order in the Siebel Environment” on page 129](#)

Setting Oracle Database Configuration Parameters for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: Oracle only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

This topic provides upgrade-specific settings for the Oracle RDBMS. Use the following strategy to set parameters:

- Read *Siebel System Requirements and Supported Platforms* on Oracle Technology Network to determine the supported Oracle version and any required patches. If required, upgrade your Oracle Database and client software to the supported version, using Oracle’s tools and documented procedures.
- Set parameters using the recommendations in Bulletin 478028.1 (Doc ID) on *OracleMetalink 3*. Bulletin 478028.1 (Doc ID) was previously published at Siebel Technical Note 582.
- Set additional parameters using the recommendations in *Siebel Installation Guide* for the operating system you are using. Recommendations are located in the chapter on configuring the RDBMS.
- Configure the database and set parameters as indicated in the following topics.
- After the upgrade, reset the configuration parameters to the values listed in *Siebel Installation Guide* for the operating system you are using and in Bulletin 478028.1 (Doc ID) on *OracleMetalink 3*. This document was previously published as Siebel Technical Note 582.

General Configuration

Before upgrading an Oracle Database, complete the following tasks:

- Verify that your Oracle clients and Oracle Database server are at the same release level.
- **UNIX only.** If you have installed the Oracle 64-bit client on the Siebel Server, verify that \$ORACLE_HOME/lib32 instead of \$ORACLE_HOME/lib is included in LIBPATH (AIX), SHLB_PATH (HP-UX), LD_LIBRARY_PATH (Solaris).
- **pctincrease.** For upgrades, compute a high enough value for pctincrease on tablespaces that contain application tables and indexes so that upgrading does not create large numbers of extents.

- **pctfree.** Rebuild some of your larger tables with a large value for pctfree (30 or higher). Table size depends on which Siebel applications you have installed. For example, if you are upgrading Siebel Financial Services, S_ASSET is a large table and S_ADDR_ORG is not used at all.

You must increase pctfree before the upgrade because many new columns are added to tables during the upgrade. Migrating data into the new columns during the upgrade is likely to cause row chaining, which degrades upgrade performance.

- **DB_CACHE_SIZE.** Set this parameter to a minimum of 394264576.
- **SORT_AREA_SIZE.** Set this parameter to a minimum of 1524288. This significantly reduces the time required to complete a repository merge.
- Other `ini t. ora` parameters that the DBA should review are:
 - `sort_area_retained size`
 - `log_buffers`
- **UNDO_MANAGEMENT (Oracle 10g and later).** Set `UNDO_MANAGEMENT = MANUAL` before the repository merge. This turns off Automatic Undo Management (AUM). You can turn AUM back on after the repository merge, as desired. For more information on how AUM affects upgrade, see Alert 477025.1 on Oracle *Metalink* 3. This document was previously published as Alert 848.
- **Rollback Segments.** Verify that you have only one large rollback segment on line that is appropriately sized so that the largest of transactions can be accommodated. Take all other rollback segments off line.

The upgrade may affect some of the largest tables in your implementation of Siebel 7.x, causing them to grow by as much as 40%.

Customer experience has shown that repository merges involving multiple languages can require a rollback segment as large as 1 GB.

Query Optimizer Settings

For the upgrade, use the query optimizer settings listed in Bulletin 478028.1 (Doc ID) Oracle *Metalink* 3.

This document was previously published as Siebel Technical Note 582 and Siebel Alert 1011.

Verifying Oracle Database Parameters for Multiple CPUs in a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: Oracle only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

If you have multiple CPUs, verify that the following parameters are set correctly. For information on settings, refer to Oracle documentation:

- `parallel_max_servers`. To enable use of multiple CPUs, this must be set to a number greater than 1.
- `parallel_min_servers`
- `_db_block_lru_latches`

Verifying the Oracle Database ODBC Definition for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: Oracle only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

When you install a Siebel Server, the installer creates the ODBC definition for the Oracle Database you specify.

When you upgrade your production test environment and your production environment, you must manually create an ODBC definition for connecting to your development environment database.

To verify or edit an ODBC definition under Windows

- 1 Start the registry editor (regedit).

CAUTION: Editing the registry can adversely affect the operating system. Be sure you understand how to use regedit correctly.

- 2 Navigate to the following location:

HKEY_LOCAL_MACHINE\Software\ODBC\ODBC.INI\ODBC_Name

Where:

ODBC_Name is the ODBC name for the Oracle Database.

- 3 In the data display pane, verify that the following entries are present:

Name	Type	Data
ColumnsAsChar	REG_SZ	1
ColumnSizeAsCharacter	REG_SZ	1

If these entries are not present, right-click in the data display pane and choose New > String Value to add them.

- 4 Step off the `ODBC_Name` and return to it. Verify that the two new entries are present and correct.
- 5 Close the Registry Editor and reboot.

To verify or edit an ODBC definition under UNIX

- 1 Navigate to the following file in the Siebel Server installation directory:
\$SIEBEL_ROOT/sys/.odbc.ini
- 2 Open the .odbc.ini file and add the following two entries:
ColumnsAsChar=1
ColumnSizeAsCharacter=1
- 3 Save the file.
- 4 Stop and restart any processes that are using this .odbc.ini file.

11

Preparing a Microsoft SQL Server Database for a Siebel Upgrade

This chapter provides initial preparatory information for a Siebel Database upgrade on the Microsoft SQL Server Database. This chapter includes the following topics:

- [“Verifying Microsoft SQL Server Sort Order for a Siebel Upgrade” on page 185](#)
- [“Setting Microsoft SQL Server Temporary Space Size for a Siebel Upgrade” on page 186](#)
- [“Setting Microsoft SQL Server Configuration Parameters for a Siebel Upgrade” on page 187](#)
- [“Rebuilding Microsoft SQL Server Clustered Indexes for a Siebel Upgrade” on page 188](#)

Related Topics

[Chapter 5, “How the Siebel Database Upgrade Works”](#)

[Chapter 4, “How to Perform a Siebel Database Upgrade”](#)

Verifying Microsoft SQL Server Sort Order for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: MS SQL Server only.

Platforms: Windows only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Verify that the sort order of the master database and the database instance are the same. This prevents repository merge failure due to sort order mismatch.

Binary sort order is required for the development environment upgrade and strongly recommended for the production environment upgrades.

When you install Microsoft SQL Server, the collation method of the database instance is set by default to dictionary sort order. Every database you create thereafter inherits this setting.

When you create a database, you can accept the inherited sort order or specify the sort order. It is recommended that you set the sort order to binary at the Microsoft SQL Server instance level so that this sort order is inherited by newly created databases.

The sort order of the master database cannot be changed without rebuilding the instance. Consult your Microsoft documentation for instructions on setting database collation.

To verify that your database was created using a binary collation sequence

- 1 In the Query Analyzer window, enter the following command:

```
sp_hel psort
```

This command provides a sort order description.

- 2 Review the sort order description to verify binary sort order; for example:

```
Lati n1_General _BIN
```

If you find that your Microsoft SQL Server database was not created using a binary collation sequence, you must rebuild your database and reload your data. Review the Microsoft documentation for detailed instructions.

Related Topics

[“About Database Sort Order in the Siebel Environment” on page 129.](#)

Setting Microsoft SQL Server Temporary Space Size for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: MS SQL Server only.

Platforms: Windows only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Set the size of the database that Microsoft SQL Server uses for temporary space needed to execute queries.

To setup TEMPDB space

- 1 Make TEMPDB as big as the biggest table in the Siebel Database, or half the size of the Siebel Database.

- 2 Make sure that the files used by TEMPDB are configured to allow auto-growth.

This allows SQL Server to expand the temporary database as needed to accommodate upgrade activity. Alternatively, you can set MAXSIZE to the size of the biggest table or to 50% of the size of the Siebel Database.

- 3 Consider putting TEMPDB on a separate drive to improve performance.

- 4 Execute `dbcc shrinkdatabase` against TEMPDB.

Setting Microsoft SQL Server Configuration Parameters for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: MS SQL Server only.

Platforms: Windows only.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

This topic provides upgrade-specific settings for the Microsoft SQL RDBMS. Use the following strategy to set parameters:

- Set parameters using the recommendations in *Siebel Installation Guide* for the operating system you are using. Recommendations are located in the chapter on configuring the RDBMS.
- For the upgrade, revise the configuration parameters listed in [Table 26](#).
- After the upgrade, reset the configuration parameters to the values listed in *Siebel Installation Guide* for the operating system you are using.

[Table 26](#) lists upgrade settings for Microsoft SQL Server database parameters. For parameters not listed in this table, it is recommended that you accept the default settings.

Most of the parameter settings in the table are the default settings.

Table 26. Microsoft SQL Configuration Parameters

Parameter	Setting
Max. degree of parallelism	1
Cost threshold for parallelism	5
Fill factor (%)	90
Index create memory (KB)	0

For the Siebel Database, set the following options to ON (enabled) for the upgrade:

- **truncate log on chkpt.** Set this option to ON (enabled) for upgrade only. Also, for upgrade only, execute the `alter` command against the Siebel Database, specifying `set recovery simple`.
- **torn page detection.**
- **auto create statistics.**
- **auto update statistics.**
- **Database size.** Increase your database file size by resetting the Autogrowth parameter to between 25% and 50%. Failure to do this could diminish upgrade performance and possibly impact the success of your upgrade.

- For a full list of recommended settings for your postupgrade production environment, see the chapter on configuring the RDBMS in *Siebel Installation Guide* for the operating system you are using.

Rebuilding Microsoft SQL Server Clustered Indexes for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: MS SQL Server only.

Platforms: Windows only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

If you have large tables that you use extensively (such as S_EVT_ACT, S_CONTACT, S_OPTY, S_OPTY_POSTN, S_ORG_EXT), use the Microsoft SQL Server `create index` command with `drop_existing` clause to rebuild large tables with high fill factor (60%-70%).

12 Preparing Siebel Application Data for Upgrade

This chapter provides guidelines for preparing your Siebel Application data for the Siebel Database upgrade. This chapter includes the following topics:

- [“Upgrade Flow Orientation” on page 190](#)
- [“Preparing Siebel Workflow Processes for Upgrade” on page 191](#)
- [“Preparing Siebel Customized Seed Data for Upgrade” on page 192](#)
- [“Migrating Siebel Household Data” on page 193](#)
- [“Setting Up Campaign Status Values for Siebel Marketing” on page 196](#)
- [“Preserving Siebel Marketing Segment Descriptions” on page 198](#)
- [“Preparing Siebel Mobile User Data for Upgrade” on page 198](#)
- [“Setting the Value of Siebel S_SRC_PAYMENT.TYPE_CD for Upgrade” on page 200](#)
- [“Preparing Siebel Address Data for Upgrade” on page 200](#)
- [“Migrating Siebel Address Data from Custom Extension Columns” on page 201](#)
- [“Preparing Siebel Territory Management Rules for Upgrade” on page 208](#)
- [“Preparing Siebel Customizable Product Data for Upgrade” on page 209](#)

Related Topics

[Chapter 5, “How the Siebel Database Upgrade Works”](#)

[Chapter 4, “How to Perform a Siebel Database Upgrade”](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase Two of the Siebel Database upgrade flow, as shown in [Figure 10](#). At this phase of the upgrade you are performing basic tasks which will prepare your Siebel application data for the upgrade.

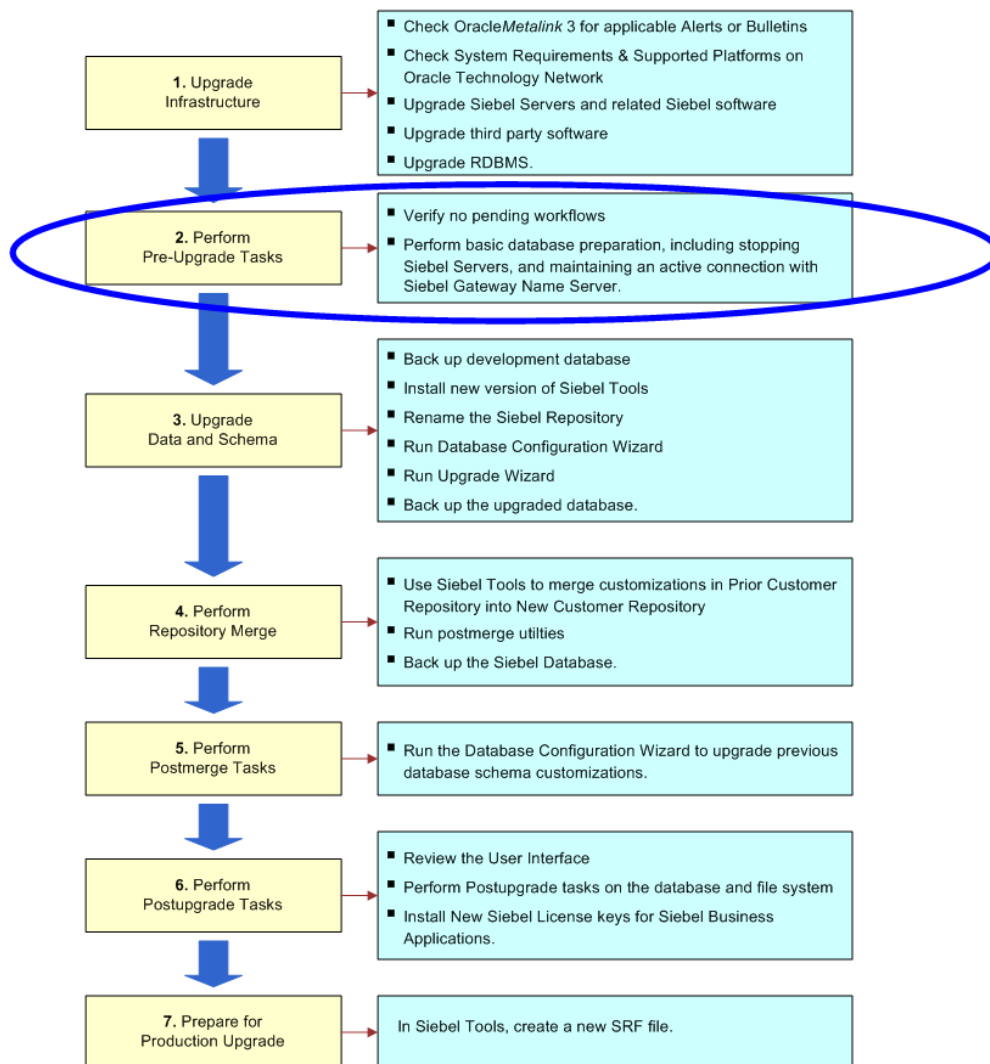


Figure 10. Phase Two of Upgrade Flow

Preparing Siebel Workflow Processes for Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

This task prepares workflows for upgrade.

At Siebel 7.7 the following changes were made to workflow management:

- The upgrade migrates custom workflows from S_WF_* tables to S_WFR_* tables in the Siebel Repository. The upgrade also updates seed workflows.
- The tables S_WF_PROP_VAL and S_WF_STEP_INST no longer store workflow process instances. Workflow process instances are now stored in S_WFA_INST_LOG, S_WFA_INSTP_Log, and S_WFA_STPRP_LOG.

Upgrades from Releases Prior to Siebel 7.7

Perform the following task before upgrading from releases prior to Siebel 7.7.x.

Preparing workflow processes for upgrade

- 1 Development environment.** Verify that all customized workflows and all seed workflows from the production database are present in the development environment Siebel Database.

If the workflows from the production environment are not present, they will not be moved to the Siebel Tools Repository. You will have to manually move them using database tools.

Oracle provides a script and an input file for migrating workflow data between your production database and your development database. See Alert 477668.1 on Oracle *Metalink* 3. This document was previously published as Siebel Alert 1016.

- 2 All environments.** Verify there are no queued or running workflows.
- 3 All environments.** Purge S_WF_PROP_VAL and S_WF_STEP_INST.

For information on purging these tables, see the topic on purging workflow process instances in *Siebel Business Process Framework: Workflow Guide*.

Upgrades from All Releases

Perform the following task for all upgrades

Preparing workflow processes for upgrade

- 1 Deactivate all workflow processes.

This ensures that run-time event actions associated with old workflows are cleaned up properly.

- 2 Verify that all tasks with status = QUEUED are removed from S_SRM_REQUEST.

The upgrade process does not clear these.

Related Topics

[“Upgrade Planning for Siebel Workflow Designer” on page 152](#)

Preparing Siebel Customized Seed Data for Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

Modified seed data are not upgraded, even if the upgraded seed data are required for an application to run normally, with the following exceptions:

- If you modify seed data through the GUI or through EIM, the repository merge preserves the modifications using the same logic as for other repository objects.
- As of Siebel CRM 8.0, customizations to seeded workflows in the Prior Customer Repository are merged into workflows in the New Customer Repository by the repository merge process.

Before starting the upgrade, evaluate whether you need to retain your seed data modifications. If practical, consider discarding the modifications. After the upgrade, you can reimplement those modifications that do not interfere with operation of applications.

Seed data are records provided in the Siebel Database tables as part of a Release. Seed data provides information needed to run a Siebel application. Examples of seed data are the values in a List of Values (LOV) definition, default mappings of views to responsibilities, and predefined queries.

The ROW_ID value for seed data records provided in a Release begins with 0 (zero). In addition, these records have a default LAST_UPD value of 1980_01_01. The LAST_UPD field stores the date when the record was last updated.

If you modify a seed data record, the value of LAST_UPD is changed from 1980_01_01 to the date the modification was made. If you add a new seed data record, the ROW_ID does not begin with 0, and the LAST_UPD value is the record creation date.

When you perform the upgrep part of the upgrade, the dataimp utility upgrades seed data records as follows:

- 1 For records where ROW_ID begins with 0, it erases seed data records where the value for LAST_UPD is 1980_01_01, unless prevented by scripting.

- 2 Dataimp replaces these records with those contained in seed data files included in the Release. In the new records, the value for LAST_UPD is 1980_01_01
- 3 Dataimp does not erase and replace records where the value for LAST_UPD is later than 1980_01_01. Instead, for each of these records, dataimp writes an error to the log file. The error is benign and does not cause the upgrade to fail.

Use these error entries to verify which seed data records were not changed.

For seed data records you have modified, you can discard the modifications by changing the value of LAST_UPD to 1980-01-01. This causes dataimp to replace these records with those from the new Release. For seed data records you have created, the upgrade process retains these records.

To discard seed data modifications

- 1 In Siebel Tools, select the Table object.
- 2 In the list applet, query for "*" in the Seed Filter column.
The query returns a list of tables containing seed data.
- 3 Use one of the following scripts to set LAST_UPD to 1980-01-01 for customized seed data records in these tables. In the scripts, *tablename* is the name of table containing seed data.

Database	Script
Oracle	UPDATE <i>tablename</i> SET LAST_UPD = TO_DATE(' 1980-01-01' , ' YYYY-MM-DD') WHERE ROW_ID LIKE ' 0%' AND LAST_UPD > TO_DATE(' 1980-01-01' , ' YYYY-MM-DD')
IBM DB2UDB and IBM DB2 for z/OS	UPDATE <i>tablename</i> SET LAST_UPD = TIMESTAMP(' 1980-01-01-00.00.00') WHERE ROW_ID LIKE ' 0%' AND LAST_UPD > TIMESTAMP(' 1980-01-01-00.00.00')
Microsoft MS SQL	UPDATE <i>tablename</i> SET LAST_UPD = CONVERT(DATETIME, ' 1980-01-01') WHERE ROW_ID LIKE ' 0%' AND LAST_UPD > CONVERT(DATETIME, ' 1980-01-01')

Related Topics

- ["About the Siebel Database Upgrade Log Files" on page 72](#)
- ["About the Siebel Repository Merge" on page 91](#)
- ["Upgrade Planning for Siebel Workflow Designer" on page 152](#)

Migrating Siebel Household Data

Upgrades: Applies to Siebel Financial Services upgrades from 7.x that have retained the Siebel 6.x form of household associations.

Environments: Production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Siebel 7.0.x introduced the Party model. This changed the way relationships between households and entities, such as activity and claim, are handled.

You have two options for migrating household data:

- Migrating household relationships to the Party model (recommended)
- Retaining the Siebel 6.x form of household relationships

Migrating Household Relationships to the Party Model

To check household data integrity and support migration of household data to the Party model, you must run the household verification script (HH_MIG_populate.sql).

The script verifies that at least the same number of entities will belong to a household after the upgrade as belong to it before the upgrade.

The household verification script makes the following assumptions:

- A household has at least one contact.
- The primary contact of a Policy/Financial Account is one of the contacts associated with this Policy/Financial Account.
- The primary contact of a Claim is one of the contacts associated with this Claim.
- The primary contact of an Opportunity is one of the contacts associated with this Opportunity.
- The primary contact of a Company is one of the contacts associated with this Company.

The script populates a temporary table with data, TEMP_HH_OBJ_MIG and generates a report based on an output file. Output is in the form of row IDs. The script verifies that every household associated with an entity includes a contact associated with that entity.

If there is no output, this means data integrity is good, and no action is required. If you receive output, you must examine the relationship between contacts and households.

Do not run the household verification script if you intend to maintain the Siebel 6.x method of handling household relationships.

To run the household verification script

1 Type the following command:

- Windows:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /a /c REM /separator / /O  
OutputFileLocation\HH_Mig_populate.txt  
/L LogFileLocation\HH_Mig_populate.log ScriptLocation\HH_Mig_populate.sql /v y
```

■ UNIX:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /a /c REM /separator / /O
OutputFileLocation/HH_Mi g_popul ate. txt
/L LogFileLocation/HH_Mi g_popul ate. l og ScriptLocation/HH_Mi g_popul ate. sql /v
```

where:

Tableowner = Tableowner

Password = Tableowner password

ODBCDataSource = Data source of the database

OutputFileLocation = Location of the output file:

■ Windows: *SI EBEL_ROOT\l og\HH_Mi g_popul ate. txt*

■ UNIX: *\$SI EBEL_ROOT/l og/HH_Mi g_popul ate. txt*

LogFileLocation = Location of the log file :

■ Windows: *SI EBEL_ROOT\l og\HH_Mi g_popul ate. l og*

■ UNIX: *\$SI EBEL_ROOT/l og/HH_Mi g_popul ate. l og*

ScriptLocation = Location of the script:

■ Windows: *DBSRVR_ROOT\database_pl atform\HH_Mi g_popul ate. sql*

■ UNIX: *DBSRVR_ROOT/database_pl atform/HH_Mi g_popul ate. sql*

Windows example:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /a /c REM /separator / /O
C: \sea7xx\si ebsrvr\Log\HH_Mi g_popul ate. txt /L
C: \sea7xx\si ebsrvr\Log\HH_Mi g_popul ate. l og
C: \sea7xx\dbsrvr\DB2UDB\HH_Mi g_popul ate. sql /v y
```

2 If you receive output, review the temporary table and check the following for each contact. Make corrections as needed:

- Contact is correct and household is incorrect.
- Contact is incorrect and household is correct.
- Contact is incorrect and household is incorrect.

Retaining the Siebel 6.x Form of Household Relationships

You have the option to retain the Siebel 6.x style of handling household relationships. However, this is not recommended. If you choose not to upgrade to the new household design, do the following:

- Verify that the temporary table TEMP_HH_OBJ_MIG does not exist in the database.
- Do not run the HH_Mi g_popul ate. sql script.
- Before doing the upgrade, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services to do the following:

- Verify that the business components needed to support the Siebel 6.x form of household relationships are present in the new release.
- Obtain instructions for revising these business components to support the Siebel 6.x form of household relationships.

Related Topics

[“About the Siebel Party Model” on page 130.](#)

Setting Up Campaign Status Values for Siebel Marketing

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

This optional step is not required for upgrade processing, but will simplify program and campaign reporting by making historical values equivalent with the values used as of Siebel 7.7.

The database upgrade process expects the standard Status values for the following applets. The process uses the standard values to convert records to the new campaign model during the upgrade:

- Campaigns (Business Component is Campaign)
- Campaign Plans (Business Component is DBM Campaign)
- Campaign Occurrences (Business Component is Campaign Occurrences)

If you customized the values for the Status field for these objects, convert the Status (S_SRC.STATUS_CD) of any existing records to the standard values before doing the upgrade.

Any campaign records that do not have standard Status values will not create Load Wave records during the upgrade and will not be displayed as launched campaigns in the Campaign Agent views.

Optionally, you can also migrate the status values for the following components:

- Program (Business Component is Program Container)
- Stage (Business Component is Program (DBM))

Table 27 lists the standard Status values for each object.

Table 27. Standard Status Values

Object	Standard Status Values
Program Plan	Planned Active Completed
Stage Plan	Planned Active Completed
Campaign Plan	Planned Active Completed
Campaign	Planned Active Completed
Campaign Occurrence	Pending Active Finished Manual Error Cancelled

Table 28 lists the search specifications for locating records containing Status values. Siebel 7.5 terminology is used for the object names.

Table 28. Search Specifications for Locating Records

Object	Business Component	Base Table	Search Specification
Program Plan	Program Container	S_SRC	WHERE SUB_TYPE = 'MARKETING_CAMPAIGN' AND CAMP_TYPE_CD = 'PROGRAM_CONTAINER'
Stage Plan	Program (DBM)	S_SRC	WHERE SUB_TYPE = 'MARKETING_CAMPAIGN' AND CAMP_TYPE_CD = 'PROGRAM'

Table 28. Search Specifications for Locating Records

Object	Business Component	Base Table	Search Specification
Campaigns	Campaign	S_SRC	WHERE SUB_TYPE = 'MARKETING_CAMPAIGN' AND CAMP_TYPE_CD IS NULL AND CUST_TRGT_METH_CD IS NULL AND TMPL_ID IS NULL
Campaign Plans	DBM Campaign	S_SRC	WHERE SUB_TYPE = 'MARKETING_CAMPAIGN' AND CAMP_TYPE_CD IS NULL AND CUST_TRGT_METH_CD IS NOT NULL
Campaigns (Occurrences)	Campaign Occurrences	S_SRC	WHERE SUB_TYPE = 'MARKETING_CAMPAIGN' AND CAMP_TYPE_CD IS NULL AND CUST_TRGT_METH_CD IS NULL

Related Topics

[“Upgrade Planning for Siebel Marketing” on page 146](#)

Preserving Siebel Marketing Segment Descriptions

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

When upgrading from a prior release of Siebel Marketing to Siebel 7.7 or later, the database upgrade overwrites the Description field for all Segment records (S_CALL_LST.DISC_TEXT) with the string “DO NOT USE: Segment from previous release.” If you want to preserve segment descriptions, migrate the existing descriptions to another text extension column before running the upgrade.

Related Topics

[“Upgrade Planning for Siebel Marketing” on page 146](#)

Preparing Siebel Mobile User Data for Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic applies primarily to developers running the Mobile Web Client in the development environment and to end users in the production environment. This topic applies to the production test environment only if it has Mobile Web Client users.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

After synchronizing, mobile users must make no further changes to their local databases until the upgrade has been completed. Any changes made during the upgrade are lost when they are reinitialized following the upgrade.

Complete the following steps before beginning the upgrade of either a development environment or a production environment. For additional information on these steps, see *Siebel Remote and Replication Manager Administration Guide* and *Siebel System Administration Guide*.

To prepare mobile users for the database upgrade

- 1** Perform a partial synchronization for mobile users, sending all transactions to the server database.
- 2** Verify that Mobile Web Clients have synchronized and that all changes have been merged into the server database as follows:
 - a** Check that no transaction files remain in the synchronization inbox and outbox for any mobile user. The synchronization inbox for each user is on the Siebel Server:
Windows: `SI EBEL_ROOT\docki ng\MOBI LEUSERNAME`.
UNIX: `$SI EBEL_ROOT/docki ng/MOBI LEUSERNAME`.
Transaction files are in the format *number.dx*; for example, 00000023.dx.
 - b** Check the mobile users Remote Status view and resolve any insert conflicts.
 - c** Log onto a Siebel Business Application, such as Call Center, as the Siebel administrator. Use the Administration-Server Management > Tasks screen to make sure that each Transaction Merger task has successfully completed.
 - d** Verify that Workflow Monitor and Workflow Action agents have processed all pending requests. If Workflow Manager has completed successfully, the S_ESCL_REQ table should not have any rows.
- 3** To prevent synchronization of Mobile Web Clients with the database server, stop or disable all Siebel Remote components on all Siebel Servers.
- 4** Disconnect all Web Clients from the Siebel Server by stopping the appropriate Application Object Managers, as described in *Siebel System Administration Guide*.
- 5 Upgrades from Siebel 7.5x and earlier.** Make sure that Dedicated Web Clients have disconnected from the Siebel Database Server.

The method you use to do this depends on your database. For example, with an Oracle RDBMS, you would stop the primary listener. However, all RDBMS types require starting the database in restricted mode. Refer to the documentation that you received from your RDBMS vendor for more information.

Setting the Value of Siebel S_SRC_PAYMENT.TYPE_CD for Upgrade

Upgrades from: Siebel 7.0.x.

All environments. This topic applies to all Siebel environments.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

At Siebel 7.5, S_SRC_PAYMENT.TYPE_CD became a required (NOT NULL) column. If TYPE_CD is null, the value is set to Payment during the upgrade.

Payment indicates that this is a payment from your organization to an external organization. For example, this might denote a payment of marketing funds to your customer.

Before the upgrade, review records where TYPE_CD is null. For those where you do not want the upgrade to set the value of TYPE_CD to Payment, set the value of TYPE_CD.

To set the value of TYPE_CD

- 1 In S_SRC_PAYMENT, query for records where TYPE_CD is null.
- 2 For the desired records, use Siebel EIM to set a value for TYPE_CD.
Use the values in the field's LOV.

Preparing Siebel Address Data for Upgrade

Upgrades:

- From Siebel Financial Services 7.0.x to Siebel Industry Applications (SIA) 8.0.x
- From Siebel Business Applications 8.0.x to Siebel SIA 8.0.x
- From Siebel Financial Services 6.2.1 on IBM z/OS platforms to Siebel SIA 8.0.x on IBM z/OS platforms

NOTE: This topic does not apply to Siebel Business Applications (HOR) that you are upgrading to a later release of Siebel Business Applications (HOR).

Environments: Production test, production.

Databases: All databases.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

At Siebel 7.7, the way address data is stored is changed. To prepare for the revised storage scheme, you must verify that there are no records with the same row IDs within or across the tables S_ADDR_PER and S_ADDR_ORG.

CAUTION: There must be no duplicate row IDs in these tables or the upgrade will fail.

To prepare address data for upgrade

- 1 Run `rpt_dup_addr_rowids.sql` against the Siebel Database. The script is located in the following directory:

Windows: `DBSRVR_ROOT\database_platform`

UNIX: `DBSRVR_ROOT/database_platform`

where

`database_platform` is the database type, for example DB2.

- 2 Review the output generated by the script.
- 3 If the output contains records with duplicate row IDs, use EIM or the application to delete unwanted records.
- 4 After addressing all the duplicate row IDs, rerun the script and verify there are no more duplicates.

Related Topics

[“Upgrade Planning for Migrating Siebel Address Data” on page 144](#)

[“Migrating Siebel Address Data from Custom Extension Columns” on page 201](#)

Migrating Siebel Address Data from Custom Extension Columns

Upgrades:

- From Siebel Financial Services 7.0.x to Siebel Industry Applications (SIA) 8.0.x
- From Siebel Business Applications 8.0.x to Siebel SIA 8.0.x
- From Siebel Financial Services 6.2.1 on IBM z/OS platforms to Siebel SIA 8.0.x on IBM z/OS platforms

Environments: Development and production test.

Platforms: Windows, UNIX, IBM z/OS.

NOTE: This topic does not apply to Siebel Business Applications (HOR) that you are upgrading to a later release of Siebel Business Applications (HOR).

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

If you have custom extension columns on the Address table `S_ADDR_ORG`, you must manually edit your upgrade scripts to prevent loss of data during an upgrade. Perform the following task, and contact your Oracle sales representative for Oracle Advanced Customer Services if you need assistance from Oracle's Application Expert Services.

Prerequisite. You must have run the Database Server Configuration Utilities in `upgrep` mode. This creates the files required for this procedure. You perform this procedure before running the Upgrade Wizard.

To migrate address data from custom extension columns

- 1 Development environment upgrades only.** Open the ddl.ctf file with a text editor.

Here is the path to the file in the Siebel Database Server installation:

Windows: *DBSRVR_ROOT\database_platform*

UNIX: *DBSRVR_ROOT/database_platform*

where

database_platform is the database type, for example DB2.

- 2 Development environment upgrades only.** Add the custom extension columns from S_ADDR_PER and S_ADDR_ORG that are attributes of the association of the address to contact or account to the S_CON_ADDR definition.

Example of the file:

[Object *nmn*]

Type = Table

Name = S_CON_ADDR

Space = USERSPACE1

```
Column 1 = ROW_ID VARCHAR(15) OTNULL
Column 2 = CREATED_TIMESTAMP NOTNULL DEFAULT %NOW%
Column 3 = CREATED_BY VARCHAR(15) NOTNULL
Column 4 = LAST_UPD_TIMESTAMP NOTNULL DEFAULT %NOW%
Column 5 = LAST_UPD_BY VARCHAR(15) NOTNULL
Column 6 = DCKING_NUM NUMERIC(22, 7) DEFAULT 0
... ..
Column 46 = RURAL_ROUTE_NUM VARCHAR(10)
Column 47 = START_DT TIMESTAMP
Column 48 = X_CUSTOM_EXTENSION_COLUMN1 VARCHAR(15)
Column 49 = X_CUSTOM_EXTENSION_COLUMN2 VARCHAR(15)
```

- 3 Development environment upgrades only.** In the ddl.ctf file, add the custom extension columns from S_ADDR_ORG that are attributes of the address to the S_ADDR_PER definition (provided corresponding columns do not already exist on S_ADDR_PER) similar to the example illustrated in the previous step.

- 4 Development environment and production test environment upgrades.** In the file `preschm_sia.sql`, add the custom extension columns for `S_CON_ADDR` to the migration scripts if any custom extension column was added to the definition of `S_CON_ADDR` in the `ddl.ctf` file in the preceding steps.

Here is the path to the file in the Siebel Database Server installation:

Windows: `DBSRVR_ROOT\database_platform\upgrade\version`

UNIX: `DBSRVR_ROOT/database_platform/upgrade/version`

Example of the script (this example is for inserts into `S_CON_ADDR` from `S_ADDR_PER`. (The example also applies to custom extension columns to be inserted on `S_CON_ADDR` from `S_ADDR_ORG`):

```
Insert into S_CON_ADDR
(ROW_ID
, CREATED
, CREATED_BY
, LAST_UPD
, LAST_UPD_BY
, MODIFICATION_NUM
, CONFLICT_ID
, ADDR_PER_ID
, ADDR_TYPE_CD
, CONTACT_ID
, EMAIL_ADDR
, FAX_PH_NUM
, PH_NUM
, ADDR_MAIL_CD
, END_DT
, START_DT
, ACTIVE_FLG
, X_CUSTOM_EXTENSION_COLUMN1
, X_CUSTOM_EXTENSION_COLUMN2)
select
ROW_ID
```

```
, CREATED
, CREATED_BY
, LAST_UPD
, LAST_UPD_BY
, MODIFICATION_NUM
, CONFLICT_ID
, ROW_ID
, ADDR_TYPE_CD
, PER_ID
, EMAIL_ADDR
, FAX_PH_NUM
, PH_NUM
, ADDR_MAIL_CD
, END_DT
, START_DT
, case
when ACTIVE_FLG is null then 'Y'
else ACTIVE_FLG
end
, X_CUSTOM_EXTENSION_COLUMN1
, X_CUSTOM_EXTENSION_COLUMN2
from S_ADDR_PER APT
where not exists
(select 'x' from S_CON_ADDR CAD
where CAD.CONTACT_ID = APT.PER_ID
and CAD.ADDR_PER_ID = APT.ROW_ID
and CAD.CONFLICT_ID = APT.CONFLICT_ID
)
and not exists
(select 'x' from S_CON_ADDR CAD1
```

where CAD1.ROW_ID = APT.ROW_ID

)

and APT.PER_ID is not null

- 5 Development environment and production test environment upgrades.** In the file `preschm_sia.sql`, add the custom extension columns for S_ADDR_PER to the migration script if any custom extension column was added to the definition of S_ADDR_PER in the `ddl.ctf` file.

Here is the path to the file in the Siebel Database Server installation:

Windows: `DBSRVR_ROOT\database_platform\upgrade\version`

UNIX: `DBSRVR_ROOT/database_platform/upgrade/version`

Example of the file:

```
insert into S_ADDR_PER
```

```
(ROW_ID
```

```
, CREATED
```

```
, CREATED_BY
```

```
, LAST_UPD
```

```
, LAST_UPD_BY
```

```
, MODIFICATION_NUM
```

```
, CONFLICT_ID
```

```
, DISA_CLEANSER_FLG
```

```
, ADDR
```

```
, ADDR_NUM
```

```
, ADDR_TYPE_CD
```

```
, CITY
```

```
, COMMENTS
```

```
, COUNTRY
```

```
, COUNTY
```

```
, INTEGRATION_ID
```

```
, INTEGRATION2_ID
```

```
, INTEGRATION3_ID
```

```
, PROVINCE
```

```
, STATE
```

```
, ZI PCODE
, ADDR_NAME
, NAME_LOCK_FLG
, DESCR I PTOR
, EMAI L_ADDR
, FAX_PH_NUM
, PH_NUM
, ADDR_LI NE_2
, ADDR_LI NE_3
, DCKI NG_NUM
, LATI TUDE
, LONGI TUDE
, ACTI VE_FLG
, ADDR_MAI L_CD
, ADDR_SUB_CD
, END_DT
, RURAL_ROUTE_NUM
, START_DT
, X_CUSTOM_EXTENSI ON_COLUMN1
, X_CUSTOM_EXTENSI ON_COLUMN2
)
sel ect
ROW_ID
, CREATED
, CREATED_BY
, LAST_UPD
, LAST_UPD_BY
, MODI FI CATI ON_NUM
, CONFLI CT_ID
, DI SA_CLEANSE_FLG
```

```

, ADDR
, ADDR_NUM
, ADDR_TYPE_CD
, CITY
, COMMENTS
, COUNTRY
, COUNTY
, INTEGRATION_ID
, INTEGRATION2_ID
, INTEGRATION3_ID
, PROVINCE
, STATE
, ZIPCODE
, case when ADDR_NAME = (select t.DUP_ADDR_NAME
    from    TMPTBL_ADDR t
    where t.DUP_ADDR_NAME = S_ADDR_ORG.ADDR_NAME and t.MIN_ROW_ID <>
    S_ADDR_ORG.ROW_ID
        )
    then {fn concat({fn concat({fn rtrim({fn substring(ADDR_NAME, 1, 99- {fn
    length(ROW_ID)}})}), ' : ' ), ROW_ID)}
    el se ADDR_NAME
end
, NAME_LOCK_FLG
, DESCRIPTOR
, EMAIL_ADDR
, FAX_PH_NUM
, PH_NUM
, ADDR_LINE_2
, ADDR_LINE_3
, DCKING_NUM
, LATITUDE

```

```
, LONGITUDE  
, ACTIVE_FLG  
, ADDR_MAIL_CD  
, ADDR_SUB_CD  
, END_DT  
, RURAL_ROUTE_NUM  
, START_DT  
, X_CUSTOM_EXTENSION_COLUMN1  
, X_CUSTOM_EXTENSION_COLUMN2  
from S_ADDR_ORG
```

- 6 Development environment upgrades only.** In Siebel Tools, add the custom extension columns defined in the previous steps to S_CON_ADDR and S_ADDR_PER to the Prior Customer Repository.

Remember that the Prior Customer Repository is the renamed Siebel Repository for the release you are upgrading from.

- 7 Development environment upgrades only.** Configure your application to expose the custom extension columns added to S_ADDR_PER.

Related Topics

[“Upgrade Planning for Migrating Siebel Address Data” on page 144](#)

[“Preparing Siebel Address Data for Upgrade” on page 200](#)

Preparing Siebel Territory Management Rules for Upgrade

Upgrades from: Siebel 7.8.x.

Environments: Production test, production.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Territory rules management was introduced in Siebel 7.8. As of Siebel 8.0, territory rules cannot have overlapping effective date ranges. For example, the following two rules have overlapping effective date ranges:

Rule 1

Territory: A100

Account: XYZ Corp

Effective Start Date: 1/1/2006

Eff. End Date: 6/30/2006

Rule 2

Territory: A100

Account: XYZ Corp

Effective Start Date: 4/1/2006

Eff. End Date:

You must identify rules with overlapping effective date ranges and modify the rules to eliminate any overlap.

To prepare territory rules for upgrade

- 1 Run the following script against the database:

Windows: `DBSRVR_ROOT\common\TM_DupRuleCheck.sql`

UNIX: `DBSRVR_ROOT/common/TM_DupRuleCheck.sql`

The script returns a list of all territory rules that have overlapping effective date ranges.

- 2 In Territory Management, revise rule effective date ranges to remove the overlaps.

Preparing Siebel Customizable Product Data for Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Production test, production.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

Customizable Products in Work Spaces

The upgrade does not migrate unreleased customizable products in work spaces. If you want to migrate unreleased customizable products, you must release them before the upgrade. This includes products with components and products with attributes.

Class Products

Verify that the Orderable flag is not set for class products. When this flag is not set, class products do not display as selectable products in quotes and orders after the upgrade.

The upgrade converts class products to a product and a product class. The upgrade sets the Product Class property for the product to Product Class.

Related Topics

["Upgrade Planning for Siebel Configurator" on page 158](#)

13 Upgrading the Siebel Database

This chapter provides guidelines for performing a Siebel Database upgrade. This chapter includes the following topics:

- [“Upgrade Flow Orientation” on page 212](#)
- [“Renaming the Siebel Repository” on page 213](#)
- [“Changing the Siebel Database Configuration Utilities Language” on page 214](#)
- [“Preparing to Run the Siebel Database Configuration Wizard” on page 214](#)
- [“Running the Siebel Database Configuration Wizard Under Windows” on page 218](#)
- [“Running the Siebel Database Configuration Wizard Under UNIX” on page 221](#)
- [“Starting the Siebel Upgrade Wizard” on page 224](#)
- [“Moving the Siebel Repository Files” on page 227](#)
- [“Running the Siebel Case Insensitivity Wizard” on page 228](#)
- [“Applying Siebel Additive Schema Changes” on page 232](#)
- [“Regenerating SQL Files for a Siebel Upgrade” on page 233](#)
- [“Identifying and Dropping Obsolete Indexes for a Siebel Upgrade” on page 234](#)
- [“Preparing for a No-Development-Environment Siebel Upgrade” on page 235](#)
- [“Installing New Siebel License Keys During an Upgrade” on page 236](#)

Related Topics

[Chapter 5, “How the Siebel Database Upgrade Works”](#)

[Chapter 4, “How to Perform a Siebel Database Upgrade”](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase Three of the Siebel Database upgrade flow, as shown in [Figure 11](#). At this phase of the upgrade you are performing upgrade tasks on your Siebel Database, including running the Siebel Database Configuration Utilities, and installing the new version of Siebel Tools and renaming your Siebel Repository.

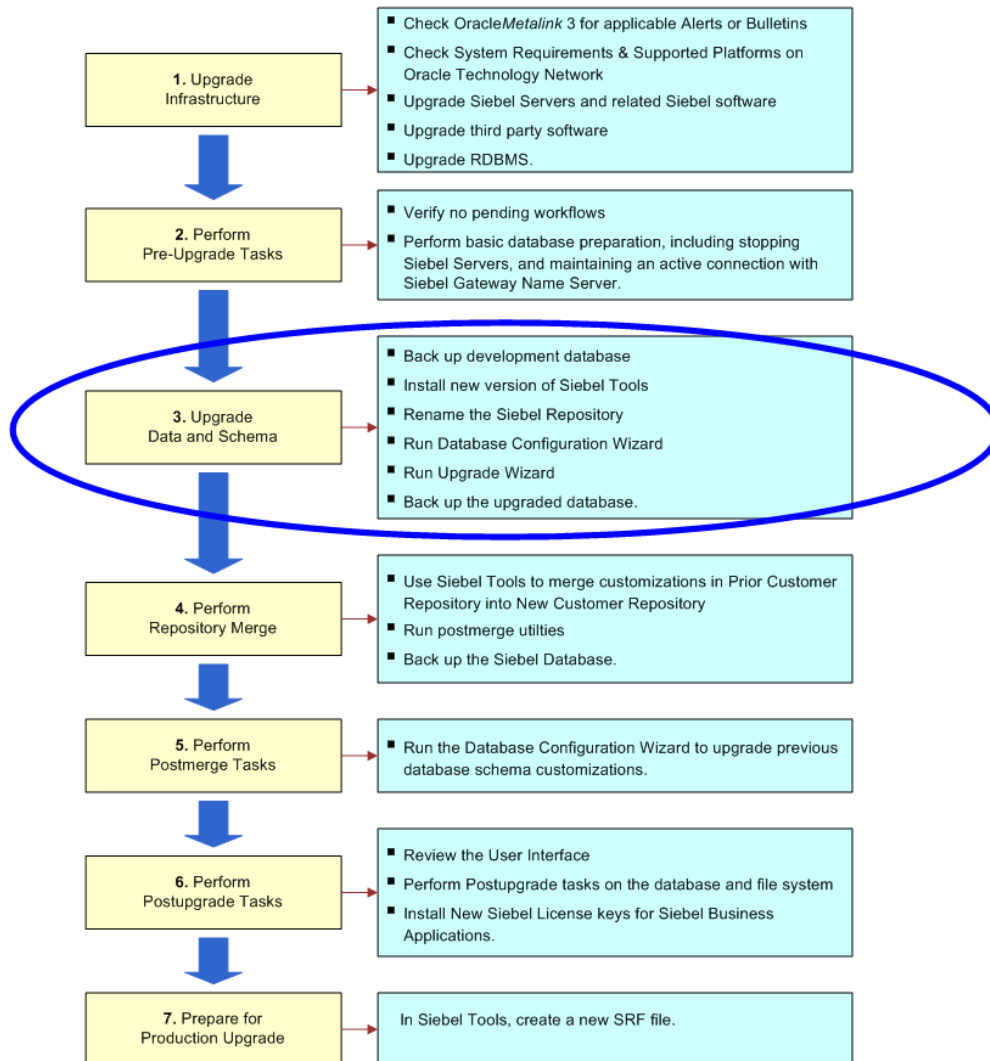


Figure 11. Phase Three of Upgrade Flow

Renaming the Siebel Repository

Upgrades from: All Supported Siebel releases.

Environments: Development environment only. (Also Production environment under UNIX)

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

To prevent a naming conflict, before you run the upgrade, you must rename your existing development repository (*Siebel Repository*) to *Prior Customer Repository*. After the upgrade, your new development repository is given the name *Siebel Repository*.

When you rename the repository, you may prefix the name with additional characters (except for a leading space). For example, “12-JUL-2007 Prior Customer Repository” would be acceptable, whereas “ 12-JUL-2007 Prior Customer Repository” would not due to the leading space. Do not append the repository name with additional characters, because this results in an error. For example, “Prior Customer Repository 12-JUL-2007” would result in an error.

To rename the repository

- 1 Start Siebel Tools and connect to the Siebel Database.

Use the version of Siebel Tools for the Siebel release from which you are upgrading.

- 2 If you archived repository objects as .sif files, and you want to have them available in your application, import these archive files back into the repository.

If you do not check these objects back into the repository, they will not be upgraded. You need only to check in those archived objects that you need in the future and want to have available in your upgraded application.

- 3 Choose View > Options.

- 4 Click the Object Explorer tab.

The Object Explorer hierarchy displays.

- 5 Locate Repository in the list, put a check mark in the adjacent box, and then click OK.

This exposes the repositories.

- 6 In the Object Explorer, click the Types tab, and then Click Repository.

- 7 In the Repositories list view, verify that your existing repositories do not use the names reserved for the upgrade process:

- **New Customer Repository**
- **New Siebel Repository**

- 8 Locate your current Siebel Repository in the list applet.

- 9 Click on the name and change it to Pri or Customer Repository.

When you rename the repository, you may prefix the name with additional characters (except for a leading space, which causes the validation to fail).

Do not append the repository name with additional characters, because this results in an error.

For more information about renaming repositories, see *Configuring Siebel Business Applications*.

TIP: Prefix the repository name with a date.

- 10 Step off the list to commit the record to the database.

If the validation check fails, verify that the repository name is not prefixed with a leading space.

Changing the Siebel Database Configuration Utilities Language

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The Database Configuration Utilities launch in the language selected when you ran the Siebel Enterprise Server Installer. You can change the language in which the Utility runs, if desired, from the language chosen during installation.

To change the Database Configuration Utilities language, see the *Siebel Installation Guide* for the operating system you are using.

If you want an additional language to appear in the language list in the Database Configuration Utilities, you first need to install the appropriate language pack on the database server and on the Siebel Server. For information about installing additional language packs, see the *Siebel Installation Guide* for the operating system you are using.

Related Topics

[“About the Siebel Database Configuration Utilities” on page 62.](#)

Preparing to Run the Siebel Database Configuration Wizard

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Use this topic to identify the information you must enter when running the Database Configuration Utilities. Collect this information, and verify that it is correct before running the utilities.

The Database Configuration Utilities are a group of wizards that request information about the upgrade process you want to perform. They add this information to a master upgrade file and call an SQL generator. The SQL generator uses the information to create or populate SQL files:

- [Table 29 on page 215](#) lists the information that the utilities request for performing an upgrade.
- [Table 30 on page 217](#) lists the additional information specific to Prepare for Production mode.
- [Table 31 on page 218](#) lists the additional information specific to Additive Schema Changes mode.

After the Database Configuration Wizard exits you run the Upgrade Wizard. The Upgrade Wizard executes the SQL files against the Siebel Database.

Table 29. Information Required for the Database Configuration Utilities

Field Name or Menu	Required Information
Gateway Name Server Address	The Siebel Gateway Name Server machine name.
Enterprise Server Name	The Enterprise Server name.
Siebel Server Directory	The absolute path of the directory where the Siebel Server is installed. For UNIX, do not enter the string \$SIEBEL_ROOT.
Siebel Database Server Directory	The absolute path of the directory where the Siebel Database Server is installed—for example: C: \sba80\dbsrvr.
RDBMS Platform	Choose the RDBMS type.
Siebel Database Operation menu	For upgrep, upgphys and Prepare for Production modes, choose Upgrade Database. For Apply Additive Schema Changes mode, choose Apply Additive Schema Changes. The remaining menu choices are for database installation and administration.
Environment Type	Choose Development for development environment upgrades. Choose Production for production test environment and production environment upgrades.

Table 29. Information Required for the Database Configuration Utilities

Field Name or Menu	Required Information
Upgrade Options	<p>Choose one of the following:</p> <ul style="list-style-type: none"> ■ Development Environment: <ul style="list-style-type: none"> ■ Upgrade Siebel Database Schema (upgrep) ■ Upgrade Custom Database Schema (upgphys) ■ Production Environment: <ul style="list-style-type: none"> ■ Prepare for production ■ Upgrade Siebel Database Schema (upgrep and upgphys)
Siebel Industry Application	<p>Choose the industry application you are upgrading from.</p> <p>If you have upgraded to the base Siebel Business Application as part of upgrading to the new Siebel Industry Application release, choose Siebel Horizontal Application.</p>
Current Siebel Version	Choose the application version you are upgrading from.
Database Encoding	Indicate whether your database uses a Unicode code page.
Language Selection	Upgrades with more than one language pack installed only. Choose the primary (base) language that is installed. The primary language is the language in which the data is stored in the Siebel Database that is being upgraded.
ODBC Data Source Name	Verify the ODBC name for connecting to the Siebel Database you are upgrading. If it is not correct, enter the correct ODBC name.
Database User Name and Database Password	Account name and password for the Siebel administrator of the Siebel Database you are upgrading.
Database Table Owner and Database Table Owner Password	Account name and password for the Siebel Database table owner.
Index Table Space Name and Table Space Name	Oracle and IBM DB2 only. Index table space name and table space name (4-KB table space name for DB2).
16-KB Table Space Name, 32K Table Space Name	IBM DB2 only. The 16-KB and 32-KB tablespace names.
Database Server OS	Choose the RDBMS server operating system type.

Table 29. Information Required for the Database Configuration Utilities

Field Name or Menu	Required Information
Parallel Indexing	<p>Oracle only. Select parallel indexing if you want SQL commands for index creation to include the arguments parallel, no logging.</p> <p>This causes an index to be created using parallel processing. This requires an RDBMS server with multiple CPUs. See your DBA to confirm if your RDBMS server is configured for parallel processing.</p> <p>TIP: Oracle Library search phrase: parallel execution.</p> <p>Selecting parallel indexing does not cause multiple indexes to be created simultaneously, in parallel. To do this, you must set up parallel index-creation threads using Siebel Upgrade Tuner. You create parallel threads as part of tuning the production upgrade files. See Chapter 20, "Tuning the Siebel Upgrade Files."</p>
Security Group ID/Grantee	<p>Security group or grantee name for Siebel application users. Must have select, update, insert, and delete privileges on Siebel application tables. Default is SSE_ROLE.</p>
Log Output Directory	<p>Accept the default or enter the directory name. If the directory does not exist, it will be created. Do not use special characters such as spaces or slashes.</p>

Additional Information Required for Prepare for Production Mode

NOTE: This topic is only applicable for upgrades from Siebel Business Applications release 7.5.3 to version 8.1. If, for example, you are upgrading from version 7.7 to 8.0, you need not perform this step.

Before performing the production upgrep and upgphys, you must run the Database Server Configuration Utilities in Prepare for Production Mode. The additional information shown in [Table 30](#) is required when you run the Database Server Configuration Utilities in Prepare for Production mode.

Note that several screens request information about the Siebel Database in the development environment, not the production test or production environment.

Table 30. Additional Information Required for Prepare for Production Mode

Screen Name	Required Information
ODBC Data Source Name for Development Database	<p>The ODBC name for connecting to the development environment Siebel Database. If you are upgrading without a development environment, this is the ODBC name of the reference database.</p>
Database User Name for Development Database	<p>Account name and password of the Siebel administrator of the Siebel Database in the development environment.</p>

Table 30. Additional Information Required for Prepare for Production Mode

Screen Name	Required Information
Database Table Owner for Development Database	Account name and password for the Siebel Database table owner in the development environment.
Repository Name for Development Database	Enter the name of the upgraded Siebel Tools repository in the development environment database. Typically, this is Siebel Repository.

Additional Information Required for Apply Additive Schema Changes Mode

You can optionally run the Database Configuration Utilities in Apply Additive Schema Changes mode. Additive Schema Changes generates an SQL script, `schema.additive.sql`. The script contains production upprep schema changes that can be performed while the database is online. This reduces the amount of time required for the upprep when the database is offline.

The additional information shown in [Table 31](#) is required when you run the Database Configuration Utilities in Apply Additive Schema Changes mode.

Table 31. Additional Information Required for Apply Additive Schema Changes Mode

Field Name	Required Information
Schema File	The absolute path to the <code>schema.ddl</code> file. If a path is displayed, verify that <code>schema.ddl</code> is located in this directory.
Output Directory	The absolute path to the directory where <code>schema.additive.sql</code> will be placed after the Upgrade Wizard generates it. The Upgrade Wizard also places the log file in this directory.

Related Topics

[“About the Siebel Database Configuration Utilities” on page 62.](#)

[“About Siebel Additive Schema Changes Mode” on page 70](#)

Running the Siebel Database Configuration Wizard Under Windows

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Run the Database Configuration Wizard to upgrade the Siebel Database. The Wizard collects information, populates a master configuration file, and calls the SQL generator to create SQL commands. The Upgrade Wizard then uses the configuration file and SQL commands to upgrade the Siebel Database.

Prerequisites:

- Collect the information that the Database Configuration Wizard requires. See [“Preparing to Run the Siebel Database Configuration Wizard” on page 214](#).
- Install the new release’s languages packs for all deployed languages. See [“Upgrade Planning for Multilingual Siebel Deployments” on page 134](#).

NOTE: There are two procedures listed in this topic. All users should run the first procedure, and, depending on the outcome of that procedure, run the second. The procedures, and the outcomes of each are explained in the following two bullet points:

- [“To run the Database Configuration Wizard under Windows” on page 219](#). All users should perform this procedure. This procedure allows you to determine whether you have deployed languages that are not shipped with the Siebel product, or have unintended languages within your system that need to be removed. For a list of shipped languages, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network. If this procedure fails due to the presence of unshipped languages within your system, you will receive an error message stating that your present installation was found to be incomplete. You are also displayed a list of languages which caused the error. At this point, you must run the Database Configuration Wizard using a special .scm file, which is detailed in the second task.

If your system only contains shipped languages, and you still receive an installer warning, you should review the records in the S_LST_OF_VAL database table using Siebel Tools. For more information, see [“Preparing for a Multilingual Upgrade” on page 166](#).

- [“To run the Database Configuration Wizard under Windows using the dbsrvr_lang.scm model file” on page 220](#). Use this procedure only if your Configuration operation failed due to unshipped languages being deployed within your system. This procedure allows you to run the configuration utility, validate all data in shipped languages, and pass over all data in unshipped languages. You may then manually verify the data in unshipped languages following the successful completion of the Database Configuration Wizard.

To run the Database Configuration Wizard under Windows

- 1 Ensure that no server tasks except the Siebel Gateway Name Service are running in the background.

To verify, navigate to Start > Settings > Control Panel > Services.

NOTE: The Siebel Gateway Name Server must be running for the Database Configuration Wizard to execute.

- 2 Start the Database Configuration Wizard by selecting Start > Programs > Siebel SES Configuration > Configure DB Server.

The first window of the Database Configuration Wizard appears.

- 3 Enter the information requested in each screen and click Next.

- 4 After you have entered all the requested information, the wizard displays a screen that lists the values you entered. If you need to make changes, click Back.
- 5 When the window displays inquiring whether you want to start the Upgrade Wizard, do the following:
 - **Development Upgrep mode:** Answer No. Do not start the Upgrade Wizard. Instead, return to the upgrep process checklist and perform the next steps.
 - **Development Upgphys mode:** Answer Yes to start the Upgrade Wizard.
 - **Combined Production Upgrep and Upgphys mode:** Answer No. Do not start the Upgrade Wizard. Instead, return to the upgrep process checklist, and perform the next steps.
 - **Additive Schema Changes:** You will not be prompted. The Upgrade Wizard starts automatically and creates the schema. addi ti ve. sql script.
 - **Prepare for Production mode:** Answer Yes to start the Upgrade Wizard.

Just before displaying the prompt, the wizard calls the SQL generator to create or populate SQL scripts.

To run the Database Configuration Wizard under Windows using the `dbsrvr_lang.scm` model file

- 1 Verify that no server tasks are running in the background.

If necessary, stop Siebel Servers and Siebel Gateway Name Server service by navigating to Start > Settings > Control Panel > Services.
- 2 Back up the `dbsrvr.scm` file, from the `si ebsrvr\admin` directory, then rename it to `dbsrvr_orig.scm`.
- 3 In the same location, rename `dbsrvr_lang.scm` to `dbsrvr.scm`.
- 4 Select Start > Programs > Siebel SES Configuration > Configure DB Server.

The first window of the Database Configuration Wizard appears.
- 5 Enter the information requested in each screen and click Next.
- 6 After you have entered all the requested information, the wizard displays a screen that lists the values you entered. If you need to make changes, click Back.
- 7 When the window displays inquiring whether you want to start the Upgrade Wizard, do the following:
 - **Development Upgrep mode:** Answer No. Do not start the Upgrade Wizard. Instead, return to the upgrep process checklist and perform the next steps.
 - **Development Upgphys mode:** Answer Yes to start the Upgrade Wizard.
 - **Combined Production Upgrep and Upgphys mode:** Answer No. Do not start the Upgrade Wizard. Instead, return to the upgrep process checklist, and perform the next steps.
 - **Additive Schema Changes:** You will not be prompted. The Upgrade Wizard starts automatically and creates the schema. addi ti ve. sql script.

- **Prepare for Production mode:** Answer Yes to start the Upgrade Wizard.

Just before displaying the prompt, the wizard calls the SQL generator to create or populate SQL scripts.

Related Topics

[“About the Siebel Database Configuration Utilities” on page 62.](#)

[“Starting the Siebel Upgrade Wizard” on page 224.](#)

Running the Siebel Database Configuration Wizard Under UNIX

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: UNIX only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Run the Database Configuration Wizard to upgrade the Siebel Database. The Wizard collects information, populates a master configuration file, and calls the SQL generator to create SQL commands. You then run the Upgrade Wizard to upgrade the Siebel Database.

Prior to running the Database Configuration Wizard you must install the new release’s languages packs for each language that you deploy.

Prerequisites:

- Collect the information that the Database Configuration Wizard requires. See [“Preparing to Run the Siebel Database Configuration Wizard” on page 214.](#)
- Install the new release’s languages packs for all deployed languages. See [“Upgrade Planning for Multilingual Siebel Deployments” on page 134.](#)

NOTE: There are two procedures listed in this topic. All users should run the first procedure, and, depending on the outcome, run the second. The procedures, and the outcomes of each are explained in the following two bullet points:

- [“To run the Database Configuration Wizard under UNIX”](#). All users should perform this procedure. This procedure allows you to determine whether you have deployed languages that are not shipped with the Siebel product, or have unintended languages within your system that need to be removed. For a list of shipped languages, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network. If this procedure fails due to the presence of unshipped languages within your system, you will receive an error message stating that your present installation was found to be incomplete. You are also displayed a list of languages which caused the error. At this point, you must run the Database Configuration Wizard using a special .scm file, which is detailed in the second task.

If your system only contains shipped languages, and you still receive an installer warning, you should review the records in the S_LST_OF_VAL database table using Siebel Tools. For more information, see [“Preparing for a Multilingual Upgrade” on page 166.](#)

- ["To run the Database Configuration Wizard under UNIX using the dbsrvr_lang.scm model file"](#). Use this procedure only if your Configuration operation failed due to unshipped languages being deployed within your system. This procedure allows you to run the configuration utility, validate all data in shipped languages, and pass over all data in unshipped languages. You may then manually verify the data in unshipped languages following the successful completion of the Database Configuration Wizard.

To run the Database Configuration Wizard under UNIX

- 1 Verify that all servers are stopped:
 - Stop all Siebel Servers.
 - Stop the Siebel Gateway Name Server.
- 2 Make \$SIEBEL_ROOT the current directory.
- 3 Source the environment variables from the siebsrvr root directory: *<install location>/siebsrvr*
Korn shell: `. si ebenv. sh`
C shell: `source si ebenv. csh`
- 4 Review the values of the following environment variables and confirm the settings are correct:
 - SIEBEL_ROOT. This path must end in siebsrvr. For example, `/usr/siebel/sba80/siebsrvr`.
 - LANGUAGE. This is the language in which the Database Configuration Wizard runs. The value of this variable is a language identifier string. For example, `enu` is the identifier string for English.

If either \$SIEBEL_ROOT or \$LANGUAGE is not set or is incorrect, you must correct them before proceeding.
- 5 Start the Database Configuration Wizard:
`$SIEBEL_ROOT/bin/ssincfgw -args MODE=LIVE MODEL_FILE=$SIEBEL_ROOT/admin/dbsrvr.scm`
- 6 Enter the information requested in each screen and click Next.
After you have entered all the requested information, the wizard displays a screen that lists the values you entered.
- 7 If you need to make changes, click Back.
- 8 If you are performing a development upgrep or combined production upgrep and upgphys, do not start the Upgrade Wizard. Instead, return to the upgrep process checklist and perform the next steps.
- 9 If you are performing any of the following, enter the command to start the Upgrade Wizard:
 - **Development environment upgphys.**
 - **Additive Schema Changes.** The Upgrade Wizard creates the schema. additive. sql script.
 - **Prepare for Production.**
- 10 The Database Configuration Wizard will now exit and prompt you to launch the Upgrade Wizard (srvrupgwiz).

To run the Database Configuration Wizard under UNIX using the *dbsrvr_lang.scm* model file

- 1 Verify that all servers are stopped:
 - Stop all Siebel Servers.
 - Stop the Siebel Gateway Name Server.
- 2 Make \$SIEBEL_ROOT the current directory.
- 3 Source the environment variables from the siebsrvr root directory: *<install location>/siebsrvr*
 - Korn shell: `siebenv.sh`
 - C shell: `source siebenv.csh`
- 4 Review the values of the following environment variables and confirm the settings are correct:
 - SIEBEL_ROOT. This path must end in siebsrvr. For example, `/usr/siebel/sba80/siebsrvr`.
 - LANGUAGE. This is the language in which the Database Configuration Wizard runs. The value of this variable is a language identifier string. For example, `enu` is the identifier string for English.

If either \$SIEBEL_ROOT or \$LANGUAGE is not set or is incorrect, you must correct them before proceeding.
- 5 Back up the *dbsrvr.scm* file, from the following location: `$SIEBEL_ROOT/admin`. Then rename the file *dbsrvr_orig.scm*.
- 6 In the same location, rename *dbsrvr_lang.scm* to *dbsrvr.scm*.
- 7 Start the Database Configuration Wizard:
`$SIEBEL_ROOT/bin/ssincfgw -args MODE=LIVE MODEL_FILE=$SIEBEL_ROOT/admin/dbsrvr.scm`
- 8 Enter the information requested in each screen and click Next.

After you have entered all the requested information, the wizard displays a screen that lists the values you entered.
- 9 If you need to make changes, click Back.
- 10 If you are performing a development upgrep or combined production upgrep and upgphys, do not start the Upgrade Wizard. Instead, return to the upgrep process checklist and perform the next steps.
- 11 If you are performing any of the following, enter the command to start the Upgrade Wizard:
 - **Development environment upgphys.**
 - **Additive Schema Changes.** The Upgrade Wizard creates the `schema_additive.sql` script.
 - **Prepare for Production.**
- 12 The Database Configuration Wizard will now exit and prompt you to launch the Upgrade Wizard (`srvrupgwiz`).

Starting the Siebel Upgrade Wizard

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

The Siebel Upgrade Wizard executes the upgrade of the Siebel Database. It takes a master configuration file as input. This file contains environment information and a driver file name. The Upgrade Wizard executes the steps in the driver file to perform the upgrade.

As the Upgrade Wizard performs the steps in the driver file, it lists the steps in a state log. The state log is located in `/siebsrvr/LOG/process/state` where *process* is the upgrade process, for example `upgrep_prod_752` (upgrade from 7.5.2, upgrep process, production test or production environment).

If the Upgrade Wizard encounters an error and exits during an upgrade, you can restart it after correcting the error. The Upgrade Wizard reads the state log and continues the upgrade from the last successfully completed step.

When you run the Database Configuration Utilities under Windows, they will prompt you if you want to start the Upgrade Wizard. When you run the Database Configuration Utilities under UNIX, you must start the Upgrade Wizard manually.

Prerequisites for Restarting the Upgrade Wizard

If the Upgrade Wizard stops due to errors, verify that you have met these prerequisites before restarting the wizard:

- Carefully review the relevant log files to make sure that your upgrade has completed successfully up to that point.
- Back up your complete set of log files, from the beginning of the process to the point at which it stopped, to another directory.

This backup maintains a complete record of your log files, and prevents your previous log files from being overwritten, which could prevent accurate diagnosis of the reason for the break in the upgrade.

- If you are continuing a previous and incomplete schema upgrade, do not change the Log Output Directory that you previously selected.

- If problems with your environment prevent the upgrade from restarting, you must restore the database from the prior base version (the version from which you are upgrading). For example, environment problems may occur when table creation fails due to a database problem (insufficient storage or network problems), which cause subsequent upgrade steps to fail.

If you need to restore your database and restart the upgrade, delete or store the upgrade log files. The files are located in the following directory:

Windows: `$SIEBEL_ROOT\log\PROCESS\output`

UNIX: `$SIEBEL_ROOT/log/PROCESS/output`

Also delete the state.log file. It is located in the following directory:

Windows: `$SIEBEL_ROOT\log\PROCESS\state`

UNIX: `$SIEBEL_ROOT/log/PROCESS/state`

Starting the Upgrade Wizard

Use this procedure to start the Upgrade Wizard. See the following topics to stop the Upgrade Wizard.

To start the Upgrade Wizard

- 1 Navigate to the following directory:

Windows: `$SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Enter the following command:

Windows: `siebug /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`

UNIX: `srvrupgwi z /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`

where:

`_UPGRADEOPTION_ENVIRONMENT_VERSION` is the portion of the upgrade configuration file name that lists upgrade process, upgrade environment, and the Siebel Release from which you are upgrading. The file is located in `$SIEBEL_ROOT\bin` (UNIX: `$SIEBEL_ROOT/bin`).

The following numbers are used for the Siebel Release portion of the file name:

- 703
- 704
- 752
- 77
- 782

■ 80

Table 32 lists an example of the file names for an upgrade from Siebel 7.5.3.

Table 32. Example of File Names for an Upgrade from Siebel 7.5.3

Upgrade Mode	File Name
Development env. upgrep	master_upgrep_dev_753.ucf
Development env. upgphys	master_upgphys_dev_753.ucf
Additive Schema Changes	master_additive_gen.ucf
Prepare for Production	master_prepare_for_production_upgrade.ucf
Production env. upgrep and upgphys	master_upgrep_prod_753.ucf

- 3 To begin the upgrade, click OK (Windows) or click ENTER (UNIX).

The Upgrade Wizard will notify you when the upgrade process is complete.

Stopping the Upgrade Wizard Under Windows

Do not stop the Upgrade Wizard unless you are confident that an error has occurred, and the Upgrade Wizard or a utility it has called is hanging. Some SQL commands issued by the Upgrade Wizard or by its utilities can take considerable time to complete.

If you are not sure whether the Upgrade Wizard is hanging, create a service request (SR) on Oracle *Metalink* 3, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Stopping the Upgrade Wizard can have varying effects on the RDBMS. Before restarting the Upgrade Wizard, review the RDBMS log files. Run SQL commands as needed to resolve errors found in the RDBMS log files.

To stop the Upgrade Wizard, do one of the following:

- If the Upgrade Wizard has launched a separate command window in which a utility is running, close the command window. This terminates the utility and stops the upgrade.
- In the Upgrade Wizard dialog box, click Cancel.

The Upgrade Wizard will exit when the current upgrade step is complete. There may be a delay while the step completes in the RDBMS.

Stopping the Upgrade Wizard Under UNIX

Do not stop the Upgrade Wizard unless you are confident that an error has occurred, and the Upgrade Wizard or a utility it has called is hanging. Some SQL commands issued by the Upgrade Wizard or by its utilities can take considerable time to complete.

If you are not sure whether the Upgrade Wizard is hanging, create a service request (SR) on *OracleMetalink 3*, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Stopping the Upgrade Wizard can have varying effects on the RDBMS. Before restarting the Upgrade Wizard, review the RDBMS log files. Run SQL commands as needed to resolve errors found in the RDBMS log files.

To stop the Upgrade Wizard under UNIX

- 1 If the Upgrade Wizard has started a utility in a child process, stop the child process.
- 2 Exit the shell in which the Upgrade Wizard is running.
- 3 Locate and stop any orphaned child processes started by the Upgrade Wizard.

After the processes terminate, there may be a delay while the RDBMS executes the issued SQL commands.

Related Topics

[“About the Siebel Upgrade Wizard and Driver Files” on page 68](#)

Moving the Siebel Repository Files

Upgrades from: All Supported Siebel releases.

Environments: Development and production test.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Before doing the production test environment upgrade, you must copy the upgraded repository definition files (schema.ddl and custrep.dat) from the development environment to the production test environment.

Prerequisite: If you modified repository objects or schema definitions after completing the development upgrade, you must regenerate the schema.ddl and custrep.dat files. See [“Regenerating the Siebel Repository Definition Files” on page 259.](#)

To move the repository files

- 1 In the development environment, navigate to the following directory:

Windows: *DBSRVR_ROOT\PI atform*

UNIX: *DBSRVR_ROOT/PI atform*

where:

PI atform is the database platform, for example, *DBSRVR_ROOT\DB2UDB*.

- 2 Copy the following files:
custrep.dat
schema.ddl
- 3 In the production test environment, put these files in the following location:
Windows: *DBSRVR_ROOT\PI atform*
UNIX: *DBSRVR_ROOT/PI atform*
- 4 Make a copy of these files, and store them in a safe location.

Running the Siebel Case Insensitivity Wizard

Upgrades from: Siebel 8.x.

Environments: Development and production test.

Platforms: Windows, UNIX, IBM z/OS.

Databases: All databases.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

For Siebel CRM 8.0, query features are enhanced to provide indexes that directly support case and accent insensitive (CIAI) queries on eligible text columns. The CIAI Wizard configures specified columns for CIAI queries by defining CIAI columns and CIAI indexes in the repository. The Wizard also sets the Default Insensitivity property for these columns to DB Case & Accent.

The purpose of the enhanced CIAI features is to improve query effectiveness and performance. Running the CIAI Wizard is optional.

Overview of What the CIAI Wizard Does

The CIAI Wizard performs the following functions in the repository to configure columns to support CIAI queries. No columns or indexes are created in the Siebel Database until you synchronize the repository to the Siebel Database. The columns you want to configure for CIAI queries are called *base columns*:

- Validates the syntax of all records if an input file is used.
- Validates that all specified tables and columns are eligible for CIAI configuration.
- For each eligible base column, defines a new CIAI column. The CIAI column contains the data in the base column converted to uppercase.
- If you select the Single or Copy All index strategy, the Wizard defines an index on the CIAI column.
- If you select the Copy All index strategy, defines a copy of all indexes that have the base column as a key. The new indexes reference the CIAI column instead of the base column.

- Sets the Default Insensitivity property for the base column to DB Case & Accent.
- Sets flags and performs other configuration operations in the repository to support CIAI queries.

The CIAI Wizard can also be run in a special mode to set the Default Insensitivity property on columns that do not have any indexes defined.

The main purpose of the CIAI query enhancements is to provide indexes that can be used for case insensitive searches. The database does not have to perform table scans to locate records. This allows the database to perform case insensitive searches more quickly.

For example, in S_CONTACT, you configure the column LAST_NAME for CIAI queries. The CIAI Wizard defines a column called LAST_NAME_CI. When you query for the name Smith, the Object Manager creates a query similar to the following (IBM DB2):

```
SELECT column list FROM S_CONTACT
WHERE LAST_NAME_CI = SMITH
```

Choosing the Correct Repository

If you are upgrading a development environment, run the CIAI Wizard on the Siebel Repository. Later in the upgrade process, this repository will be renamed *Prior Customer Repository*.

If you have completed an upgrade of the development environment (upgprep, merge, upgphys), run the CIAI Wizard on the Siebel Repository. You then need to generate another schema.ddl file and use it to update your production test and production environments. Typically, you run the Wizard after an upgrade is complete to revise the configuration of columns you have configured for case insensitive queries.

Running the Wizard Using an Input File

Oracle provides a recommended input file for Siebel Business Applications and for Siebel Industry Applications. The input files have a .csv extension and are located in the following directory:

Windows: *Tools\Install_dir\objects*

These files list columns that are frequently used for queries and are provided as a recommendation. You can edit these files or create new input files as desired.

Prerequisites: See [“About the Siebel Case Insensitivity Wizard” on page 75](#) for information on how the CIAI Wizard works, eligibility criteria, and how to edit input files.

To run the CIAI Wizard using an input file

- 1 Review the input file and verify the following:
 - The syntax for all records is correct.
 - The tables and columns are eligible.
 - The specified configuration options are correct.
 - The configuration defaults are acceptable if any configuration options are omitted.

2 In Siebel Tools, open the repository.

3 Lock the tables listed in the input file.

4 Select Tools > Utilities > Case Insensitivity.

The CIAI Wizard displays.

5 Select "Administer the columns listed in this file."

6 Click Browse.

The tool s\obj ects directory displays, containing the default .csv input files.

7 Select the desired .csv file, and click Open.

8 In the Wizard, click Next.

The CIAI Wizard validates the syntax of the file and validates the eligibility of all tables and columns. If the file contains errors or eligibility problems, the Wizard lists the records containing errors. If you continue, the Wizard skips records containing errors or eligibility problems.

If there are errors, export the listing to a file, correct any errors in the input file, and restart the CIAI Wizard. Click Export to export the error listing to a text file.

If there are no errors, click Next.

The Wizard displays the records in the input file.

9 Review the configuration settings, and verify that they are correct.

10 If you want to change any configuration settings, click Export.

The Wizard exports the listing to a text file in input-file format.

11 Edit the text file, restart the CIAI Wizard, and specify the edited text file as the input file.

12 If the configuration settings are correct, click Next.

The Wizard displays the changes it will make to repository tables and indexes.

13 If you want to save a record of the changes, click Export.

The Wizard writes the changes to a text file.

14 Click Finish.

The Wizard configures the columns in the repository to support CIAI queries.

Running the CIAI Wizard by Selecting Columns

This procedure assumes the Object Explorer is in Types mode where objects are displayed hierarchically. An alternate way to perform the procedure is to display Object Explorer in Flat mode. Then choose the Column object and navigate to the desired columns.

Prerequisites: Review "[About the Siebel Case Insensitivity Wizard](#)" on page 75 for information on how the CIAI Wizard works, eligibility criteria, and how to edit input files.

Running the CIAI Wizard by selecting columns

- 1 In Siebel Tools, open the repository.
- 2 In the Siebel Tools Object Explorer, select Table.
- 3 In the Tables list, lock the desired table, and highlight it.
- 4 In the Object Explorer, select Table > Column.
- 5 In the Columns list, select the desired columns.
- 6 Right-click the highlighted columns, and choose Case Insensitivity from the pop-up menu.
 The CIAI Wizard validates the eligibility of the selected columns. The Wizard lists any columns that have eligibility errors. If you continue, the Wizard skips columns containing errors.
 Export the error listing to a text file for reference, correct any errors, and restart the CIAI Wizard. Click Export to export the error listing.
 The Wizard displays the configuration settings it will use to configure the columns.
- 7 Review the configuration settings and verify they are correct.
- 8 If you want to change any configuration settings, click Export.
 The Wizard exports the listing to a text file in input-file format. Edit the text file, then run the CIAI Wizard and specify the text file as the input file.
- 9 If the configuration settings are correct, click Next.
 The Wizard displays the changes it will make to the repository tables and indexes.
- 10 If you want to save a record of the changes, click Export.
 The Wizard writes the changes to a text file.
- 11 Click Finish.
 The Wizard configures the columns in the repository to support CIAI queries.

Configuring CIAI Support for Columns That Do Not Have Indexes Defined

If a column does not have any indexes defined, the CIAI Wizard does not create new columns or indexes.

However, you can run the Wizard in special mode that changes the column's Default Insensitivity property to DB Case & Accent. The Wizard performs these steps on all eligible columns in the repository. You cannot manually select columns.

Prerequisites: Review [“About the Siebel Case Insensitivity Wizard” on page 75](#) for information on how the CIAI Wizard works, eligibility criteria, and how to edit input files.

To configure columns without indexes

- 1 In Siebel Tools, open the repository.

- 2 Select Tools > Utilities > Case Insensitivity.

The CIAI Wizard displays.

- 3 Select "Enable for all unindexed columns."

- 4 Click Next.

The Wizard locates unindexed columns that meet CIAI eligibility criteria. The Wizard then displays a list of tables that need to be locked.

- 5 Click Export to export the list of tables to a text file. Then exit the Wizard.

- 6 Lock all the tables in the text file.

- 7 Start the CIAI Wizard again, and select "Enable for all unindexed columns."

The Wizard locates unindexed columns and then displays a page listing how they will be configured.

- 8 Verify that for all the columns, method = Database and Index Strategy = None.

When the index strategy is set to None, the Wizard does not create a CIAI column or indexes.

When you click Next, the Wizard displays a page listing the repository changes it will make.

- 9 Verify that for all columns, Default Insensitivity will be changed to DB Case & Accent. Click Finish.

The Wizard makes the changes to the repository.

Related Topics

["About the Siebel Case Insensitivity Wizard" on page 75](#)

["Regenerating the Siebel Repository Definition Files" on page 259](#)

Applying Siebel Additive Schema Changes

Upgrades from: All Supported Siebel releases.

Environments: Production test, production.

All databases. This topic applies to all supported databases.

Windows and UNIX only. This topic does not apply to IBM z/OS platforms.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

As of Siebel CRM 8.0, you can run Database Configuration Utilities in Apply Additive Schema Changes mode. This generates a `schema.additive.sql` script that makes schema changes to the Siebel Database schema. You do not have to take the database offline to apply this script.

Applying this script to your production database reduces the number of upgrade steps that must be performed while the production database is offline. This reduces production database downtime during the upgrade.

The recommended method for using this feature is to first test it in the production test environment as part of upgrade tuning. After verifying that `schema.additive.sql` does not adversely affect the operation of the application, you then run the script against the production database.

Prerequisites for Applying Siebel Additive Schema

- You must have run the Database Configuration Utilities in Apply Additive Schema Changes mode to generate the `schema.additive.sql` script.
- You must have reviewed the Apply Additive Schema Changes log file and verified that `schema.additive.sql` was generated without errors.
- If you are testing the script in the production test environment, the schema of the Siebel Database should be the same as the schema of the production environment Siebel Database.

To apply additive schema changes

- 1 Navigate to the directory where `schema.additive.sql` is located.
- 2 Review the script and verify that it does not make unacceptable changes to the schema.
- 3 Production environment only. Compare the script generated in the production environment with the one generated in the production test environment. If the scripts are not identical, determine why. If you cannot determine that the differences are benign, do not apply additive schema changes in the production environment.
- 4 Using any SQL editor, run the script against the Siebel Database.

You can run `schema.additive.sql` multiple times. For example, you can run the script, revise it, and run it again.

Related Topics

[“About Siebel Additive Schema Changes Mode” on page 70](#)

Regenerating SQL Files for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development and production test.

If you enter incorrect information in the Database Server Configuration Utilities, the SQL files they generate will be incorrect and must not be used. You must run the Utilities again, enter the correct information, and regenerate the SQL files.

For example, you are upgrading a non-Unicode database. You run the Database Server Configuration Utilities and enter that you are upgrading a Unicode database. The SQL that the utility generates will be incorrect and cannot be used. You must regenerate the SQL files.

When the Database Server Configuration Utilities generate SQL files, they create a lock file that prevents regenerating the SQL files. To regenerate the SQL files, you must delete the lock file.

Regenerating SQL files with the Database Server Configuration Utilities

- 1 Verify that the SQL files must be regenerated.
Typically this is caused by entering incorrect information in the Database Server Configuration Utilities. If you are unsure if the files must be regenerated, create a service request (SR) on Oracle *Metalink* 3, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- 2 Navigate to `DBSRVR_ROOT\common` (UNIX: `DBSRVR_ROOT/common`) and locate the file `sql gen. usg`. This is the lock file that prevents the utility from regenerating the SQL files.
- 3 Delete the `sql gen. usg` file.
- 4 Rerun the Database Server Configuration Utilities.

Related Topics

["About the Siebel Database Configuration Utilities" on page 62](#)

Identifying and Dropping Obsolete Indexes for a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Production test environment only. Does not apply to production environment.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

Use this topic to identify indexes that may be obsolete in the Siebel Database and can be dropped. This topic is optional but is recommended since dropping obsolete indexes improves database performance.

When you run the Database Server Configuration Utilities in Prepare for Production mode, they do the following to identify obsolete indexes:

- Compares the repository schema definition in the development environment against the Siebel Database physical schema definition in the production test environment.
- If an index is present in the Siebel Database physical schema definition but not in the repository logical schema definition, the utility creates an SQL drop statement and adds it to that index. The utility places this SQL statement in a file called `gen_obs_idx.sql`.

You must manually review the `gen_obs_idx.sql` file. If it contains indexes you want to drop (those containing a drop statement not followed by a create statement), you must copy the corresponding SQL statements to another SQL file called `obs_idx.sql`. This file is executed by the Database Server Configuration Utilities in `upgrep` mode.

When the Upgrade Wizard is then run, all indexes, including obsolete indexes, are maintained during table rebuilds and data migration. The obsolete indexes file is executed during the Create Siebel Indexes step.

Prerequisites: You must have run the Database Server Configuration Utilities in Prepare for Production mode in the production test environment.

To identify and drop obsolete indexes

- 1 Navigate to the following file:

Windows: *DBSRVR_ROOT\platform\gen_obs_idx.sql*

UNIX: *DBSRVR_ROOT/platform/gen_obs_idx.sql*

where *platform* is the database type, for example DB2UDB.

- 2 Open the file with a text editor and review the SQL statements it contains.

The SQL statements drop indexes that are present in the Siebel Database but not in the development environment repository logical schema definition.

- 3 If you want to drop an index, copy the corresponding SQL statement(s) to the following file:

Windows: *DBSRVR_ROOT\platform\obs_idx.sql*

UNIX: *DBSRVR_ROOT/platform/obs_idx.sql*

where *platform* is the database type, for example DB2UDB.

This file will be executed when you run the Database Configuration Wizard in upgrep mode.

Preparing for a No-Development-Environment Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

If your installation does not include a development environment, you do not need to merge your Siebel Repository. Instead, you can use the repository and schema definition files included in the Siebel Database Server installation.

Before performing the upgrade, you must move and rename these files.

To prepare for a no-development-environment upgrade

- 1 Navigate to *DBSRVR_ROOT\common* (UNIX: *DBSRVR_ROOT/common*) and locate the *mstrep.dat* file.
- 2 Copy the *mstrep.dat* file and rename it *custrep.dat*.
- 3 Place the *custrep.dat* file in the *DBSRVR_ROOT\Platform* (UNIX: *DBSRVR_ROOT/Platform*) directory, where *Platform* is the database platform, for example *DBSRVR_ROOT\DB2UDB*.
- 4 In the *Platform* directory, copy the *ddl.ctf* file and paste the copy into the same directory.

- 5 Rename the copy schema. ddl.
- 6 In the production test environment create a new database, separate from the Siebel Database. Install the Siebel Database from the new release in the new database. Do not migrate any data to the new database.
This database is called the reference database.
- 7 Define an ODBC for the reference database.

Installing New Siebel License Keys During an Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

With the new release of Siebel Business Applications, you received one or more license keys. You must add all of the new license keys to enable the new release of the Siebel Business Applications, which you use in the next steps of the upgrade process.

For more information on installing license keys, see *Siebel Installation Guide* for the operating system you are using.

To add new license keys

- 1 Start the new-release version of Siebel Tools
- 2 Connect to the database server as the Siebel administrator.
- 3 Add your new license keys.

14 Reviewing the Siebel Upgrade Log Files

This chapter provides information on accessing, reviewing and archiving Siebel upgrade log files. This chapter includes the following topics:

- [“Upgrade Flow Orientation” on page 238](#)
- [“Summarizing Siebel Log Files Using the Logparse Utility” on page 239](#)
- [“Reviewing Siebel Upgrade Log Files for Errors” on page 241](#)
- [“Manually Archiving Siebel Upgrade Log Files” on page 244](#)

Related Topics

[Chapter 5, “How the Siebel Database Upgrade Works”](#)

[Chapter 4, “How to Perform a Siebel Database Upgrade”](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase Three of the Siebel Database upgrade flow, as shown in [Figure 12](#). At this phase of the upgrade you are reviewing log files generated during the running of the Siebel Upgrade Configuration Utilities.

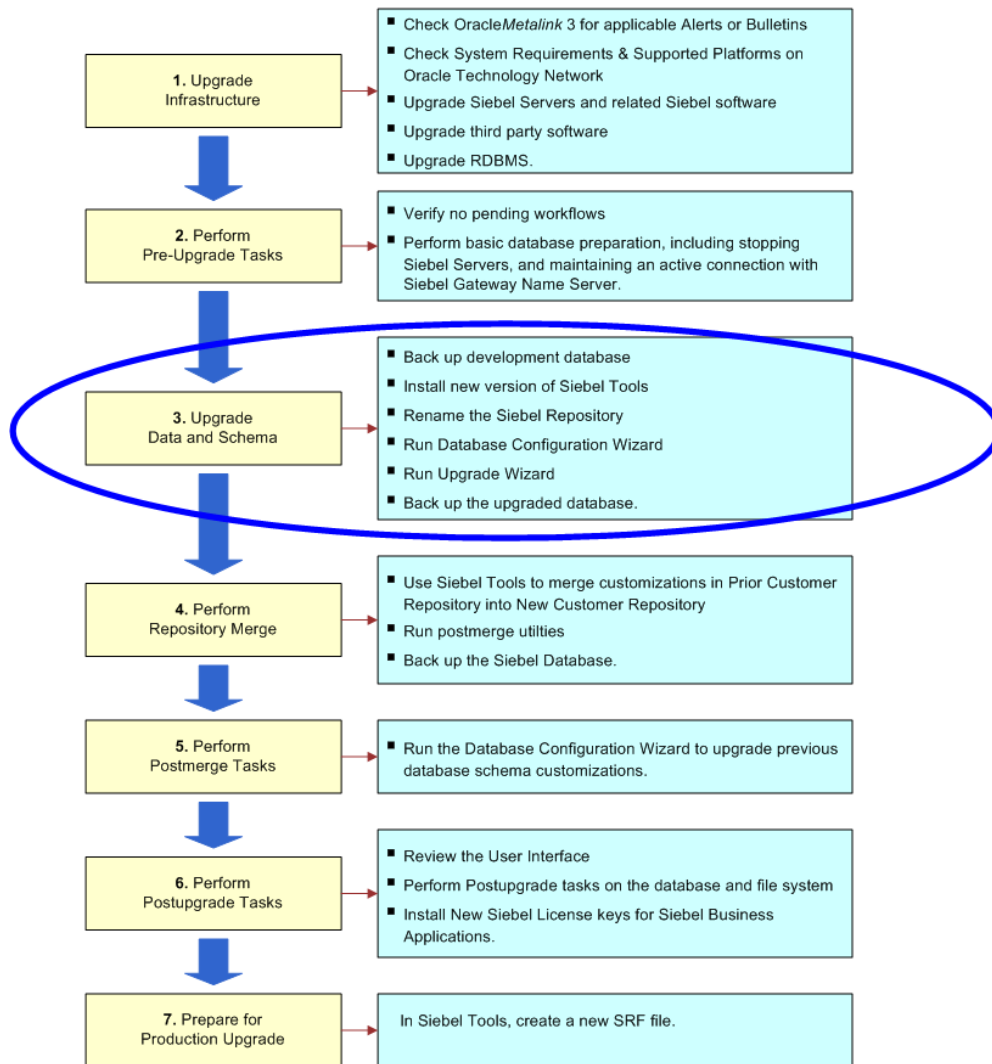


Figure 12. Phase Three of Upgrade Flow

Summarizing Siebel Log Files Using the Logparse Utility

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows and UNIX only. Does not apply to IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Use the Logparse utility to analyze and summarize the log files created when you run the Upgrade Wizard.

The Logparse utility writes its findings to a summary file:

Windows: `$SIEBEL_ROOT\log\PROCESS\summary.html`

UNIX: `$SIEBEL_ROOT/log/PROCESS/summary.html`

where:

Process is the upgrade process you want to review, for example, `upgrep_dev_77`

The Logparse utility also writes a `summary.txt` file that contains the same information as `summary.html`. Use the `summary.txt` file if you do not have a browser.

After writing the output files, the Logparse utility opens `summary.html` automatically.

[Table 33](#) lists the command line syntax for the Logparse utility.

Table 33. Command Line Syntax for the Logparse Utility

Flag	Parameter	Description	Comment
/s	install_dir	Full path to the parent directory of <i>SIEBEL_ROOT</i> (\$SIEBEL_ROOT). For example, if <i>SIEBEL_ROOT</i> is C:\sba80\si ebsrvr, then <i>install_dir</i> is C:\sba80. Enclose the path in quotes.	Required
/g	LANGUAGE_CODE	The language in which the Upgrade Wizard ran. This is called the resource language, and typically is the language in which you ran the Siebel Installation Wizard. For example, if the resource language is English, the language code is enu.	Required

Table 33. Command Line Syntax for the Logparse Utility

Flag	Parameter	Description	Comment
/r	PROCESS	Name of the schema process for which you want to generate a summary file, for example upgrep_dev_77. To obtain the schema process name, look in \siebsrvr\log. The subdirectory names in \log are the schema process names.	Required.
/l	LOGPARSE_FILENAME	Name of the Logparse log file.	Default=logparse.log
/n	MAX_NUMBER_SQL	Maximum number of longest-running SQL to display in the summary.	Default value=10
/t	THRESHOLD_TIME	Threshold time for longest running SQL, in the format hh:mm:ss.	Default value=00:10:00 (By default, SQLs that run longer than 10 minutes are displayed)
/e	MAX_NUMBER_ERRORS	Maximum number of errors to display in the summary.	Default = 8

To run the Logparse utility

- 1 Navigate to the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Run the following command. Use the syntax in [Table 33](#):

```
logparse /s install_dir /g LANGUAGE_CODE /r PROCESS /l LOGPARSE_FILENAME
/n MAX_NUMBER_SQL /t THRESHOLD_TIME /e MAX_NUMBER_ERRORS
```

Windows example (Upgrade Wizard language is English):

```
logparse /s C:\sea7xx /g enu /r upgrep_dev_704 /l logparse.log /n 10 /t 00:00:10
/e 10
```

- 3 Review the Logparse log and verify that Logparse ran without errors.

Windows: `SIEBEL_ROOT\bin\logparse.log`

UNIX: `$SIEBEL_ROOT/bin/logparse.log`

Related Topics

[“About the Siebel Database Upgrade Log Files” on page 72](#)

[“Reviewing Siebel Upgrade Log Files for Errors” on page 241](#)

Reviewing Siebel Upgrade Log Files for Errors

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows and UNIX only. Does not apply to IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Using the Logparse utility to review the log files created when you run Upgrade Wizard is the recommended way to verify that the upgrade process completed correctly and to identify errors that must be resolved.

The Logparse utility writes its findings to a summary file:

Windows: `SI EBEL_ROOT\log\process\summary.html`

UNIX: `$SI EBEL_ROOT/og/process/summary.html`

where

process is the upgrade process you want to review, for example `upgrep_dev_77`

Logparse also writes a `summary.txt` file that contains the same information as `summary.html`. Use the `summary.txt` file if you do not have a browser.

The procedures in this topic describe how to review the `summary.html` file.

Verifying the Upgrade Environment

Before reviewing the upgrade steps for errors, verify that the upgrade environment is set up correctly.

To verify the upgrade environment

- 1 If `summary.html` is not already open in your browser, navigate to `si ebsrvr\log`, and locate the subdirectory for the upgrade process you want to review.
- 2 In the desired upgrade process directory, click `summary.html` to open the file in a browser.
- 3 On the `summary.html` home page, Upgrade/Install Log File Navigation Page, locate the Parameter column in the table, and click the Parameters link.

The Parameters page displays.

- 4 Review the parameters, and verify that they are correct.

License key values are encrypted. If you find errors in the parameters, correct them, and run the upgrade process again as needed.

Determining If the Upgrade Process Completed Successfully

The upgrade processes consist of a series of steps. Each step must complete successfully.

If the Upgrade Wizard cannot complete a step, it marks the step incomplete in the state log (si_ebsrvr\log\Process\state\state.log) and exits. You must correct the error, and run the Upgrade Wizard again. When you rerun the Upgrade Wizard, it refers to the state log and resumes at the incomplete step that contained the error.

Use the following process to identify errors:

- 1 Resolve errors for steps with a Status of Incomplete.
- 2 Review all the steps with a Status of Complete. If any contain unacceptable errors, resolve these errors.
- 3 Restart the Upgrade Wizard, or if necessary, restore the database and rerun the upgrade process.

If you have any questions regarding how to resolve errors, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

To determine if the upgrade process completed successfully

- 1 If summary.html is not already open in your browser, navigate to si_ebsrvr\log, and locate the subdirectory for the upgrade process you want to review.
- 2 In the desired upgrade process directory, click summary.html to open the file in a browser.
- 3 On the summary.html home page, Upgrade/Install Log File Navigation Page, locate the Step/Error column in the table and click the Steps/Errors link.

The Steps/Errors Summary page displays.

- 4 Locate the Status column, and verify that all the steps have a status of Complete.

If the Status of all the steps is Complete, the upgrade process completed successfully.

If the Status of any step is Incomplete, the upgrade process did not complete successfully. The Upgrade Wizard has exited before completing all the steps in the upgrade process. You must identify the error, and correct it.

CAUTION: In some cases, the Upgrade Wizard can complete a step even though the step contains unacceptable errors. You must verify that all steps do not contain unacceptable errors.

Determining if Steps Contain Unacceptable Errors

There are two types of unacceptable errors:

- The Upgrade Wizard could not complete a step due to an error and exited. You can locate these errors by checking the Status field on the Steps/Errors Summary page. If the Status of a step is Incomplete, the step contains an unacceptable error.
- The Upgrade Wizard encountered an unacceptable error but was able to complete the step. You can locate these errors by examining the detailed information for each step.

To determine if steps contain unacceptable errors

- 1 If `summary.html` is not already open in your browser, navigate to `si ebsrvr\log`, and locate the subdirectory for the upgrade process you want to review.
- 2 In the desired upgrade process directory, click `summary.html` to open the file in a browser.
- 3 On the `summary.html` home page, Upgrade/Install Log File Navigation Page, locate the Step/Error column in the table, and click the Steps/Errors link
The Steps/Errors Summary page displays.
- 4 In the Name column, click the name link.
The detailed information for the step displays. An alternate method is to use the slide bar to browse down through the detailed information listing for all the steps.
- 5 IBM DB2 and Oracle: In the detailed information for each step, check the Errors item.
If the value for Errors is None, the step has no unacceptable errors.
If table lists one or more errors, these errors are unacceptable and must be corrected. Click the links to the log files to obtain more information about the errors.
- 6 MSSQL: In the detailed information for each step, check the Errors item.
If the value for Errors is None, the step has no errors.
If there is a table or a link to a table listing errors, do the following:
 - a Open the `errors.xls` spreadsheet:
Windows: `DBSRVR_ROOT\DATABASE_PLATFORM\errors.xls`
UNIX: `DBSRVR_ROOT/DATABASE_PLATFORM/errors.xls`
If an `errors.rtf` (Windows) or `errors.txt` (UNIX) file is present, you can use either of these files. They contain the same content as `errors.xls`. The errors file lists benign errors that do not need to be corrected.
 - b Compare the errors listed in the `Logparse summary.html` file to the list of acceptable errors documented in the error file.
The log files generated by Upgrade Wizard appear in the errors file as `upgwi z1.log`, `upgwi z2.log`, incrementing for additional log files.
 - c If you find the error in the errors file, the error is acceptable and no action is required.
If you find an error that is not listed in the errors file, the error is unacceptable. You must correct the condition that caused the error.
- 7 To help resolve unacceptable errors, navigate to 477324.1 (Doc ID) on Oracle *Metalink* 3. Common errors and how to resolve them are listed in this document. If the error is not listed or you cannot resolve it, contact Oracle Global Customer Support. Do not proceed with the upgrade.

Reviewing Upgrade Performance

The Logparse summary file provides the input that Upgrade Tuner uses to analyze the performance of the upgrade scripts. For example, the summary file provides Upgrade Tuner with information about SQL commands that returned 0 rows.

If you plan to use the Upgrade Tuner to tune your production upgrade, familiarize yourself with the performance information in the summary file.

To review upgrade performance in the summary file

- 1 If `summary.html` is not already open in your browser, navigate to `si_ebsrvr\log`, and locate the subdirectory for the upgrade process you want to review.
 - 2 In the desired upgrade process directory, click `summary.html` to open the file in a browser.
 - 3 On the `summary.html` home page, Upgrade/Install Log File Navigation Page, locate the following columns:
 - Performance Information for SQL
 - Performance Information for DDL
 - 4 Click the links in these columns to display performance information about the upgrade process.
- The Performance Information for SQL page displays the longest running queries and queries that returned 0 rows for SQL scripts.

The Performance Information for DDL page displays the tables created, altered, and deleted by the `ddlmp` utility and by SQL scripts.

In both pages, the Net Cost column lists the time required to perform each operation.

Related Topics

[“About the Siebel Database Upgrade Log Files” on page 72](#)

[“Summarizing Siebel Log Files Using the Logparse Utility” on page 239](#)

[“Manually Archiving Siebel Upgrade Log Files” on page 244](#)

Manually Archiving Siebel Upgrade Log Files

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows and UNIX only. Does not apply to IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

After a successful installation and upgrade, you must manually save and archive the log files located in the `SIEBEL_ROOT/log/PROCESS` (Windows) directory.

By default, only nine upgrade log files are retained for subsequent retries of the Upgrade Wizard. After nine log files have been created, when the Upgrade Wizard is rerun, it overwrites log files beginning with the earliest one created and recycles the rest as necessary. (This does not apply to the state.log file.)

The number of log files retained can be increased by resetting the `si_ebel_log_archi ve` environment variable to 20—for example, to retain twenty (20) log files.

15 Performing the Siebel Repository Merge

This chapter provides guidelines for performing a Siebel Repository merge. This chapter includes the following topics:

- [“Upgrade Flow Orientation” on page 248](#)
- [“Configuring Siebel Repository Objects to Inherit Upgrade Behavior” on page 249](#)
- [“Configuring Siebel Tools for the Repository Merge” on page 250](#)
- [“Performing a Siebel Repository Merge” on page 252](#)
- [“Reviewing the Siebel Repository Merge Log Files” on page 255](#)
- [“Generating Siebel EIM Temporary Columns” on page 257](#)
- [“Reviewing Siebel Repository Object Property Conflicts” on page 258](#)
- [“Regenerating the Siebel Repository Definition Files” on page 259](#)
- [“Setting Label Alignment for Siebel Text Fields” on page 262](#)
- [“Copying UI Files to a New Siebel Environment” on page 263](#)
- [“Deleting Unneeded Siebel Repository Files” on page 265](#)
- [“Migrating Siebel Repository Objects to the Standard UI” on page 265](#)
- [“Running the Siebel Postmerge Utilities” on page 270](#)

Related Topics

[Chapter 5, “How the Siebel Database Upgrade Works”](#)

[Chapter 4, “How to Perform a Siebel Database Upgrade”](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase Four of the Siebel Database upgrade flow, as shown in [Figure 13](#). At this phase of the upgrade you are performing the Siebel Repository merge and all associated tasks.

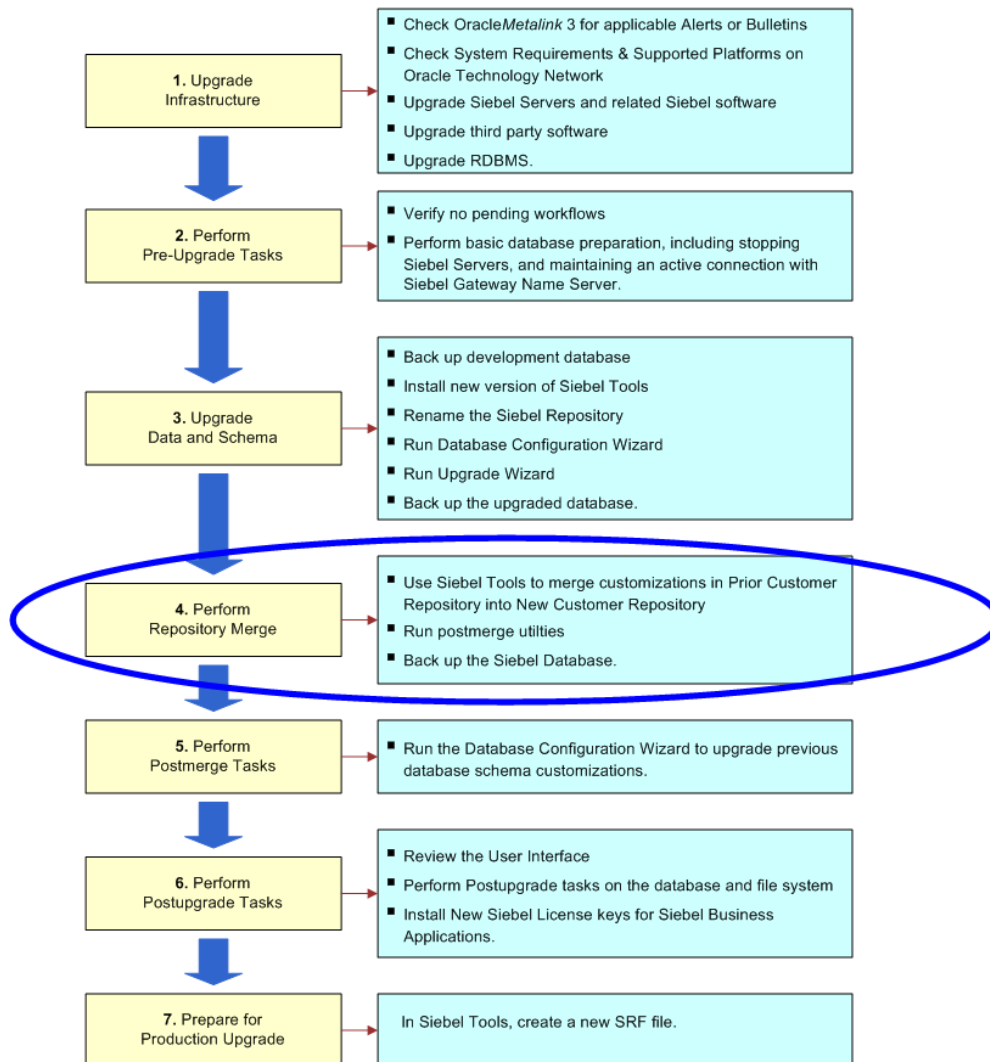


Figure 13. Phase Four of Upgrade Flow

Preparing for the Repository Merge

All upgrades. This topic applies to all upgrades.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The process of merging repositories to create a final customized repository used in the upgrade is time-intensive and resource-intensive. As a result, a merge may fail due to environmental factors, such as space constraints. When this happens, the merge process continues, even if there is a fatal database error, and the errors may go undetected.

If the merge fails, you must restore the database environment to its pre-merge state and run the merge again. Additionally, it is recommended, as another precaution, that you export the New Customer Repository to preserve existing workflows.

Recovering from a Failed Merge

It is recommended that you perform the following two tasks to preserve the pre-merge environment in the event of a failed merge.

- 1 Perform Full Database backup.** Prior to the merge, back up the entire database, then, if the merge fails, you can restore the database to its pre-merge state, and rerun the merge operation.
- 2 Export New Customer Repository.** Prior to the merge, export the New Customer Repository to create a backup copy. If the merge fails, delete the failed repository, then import the backed up copy of the New Customer Repository. See *Using Siebel Tools* for information on exporting and importing repositories using the Database Configuration Wizard.

NOTE: If you export the New Customer Repository, you must truncate the following merge log tables: S_MERGE_LOG, S_MERGE_LOG_OBJ, and S_MERGE_LOG_ATTR.

Configuring Siebel Repository Objects to Inherit Upgrade Behavior

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

As of Siebel 7.7, you can link objects together so that one object inherits the upgrade behavior of another. You do this by specifying an upgrade ancestor for an object.

You can specify an upgrade ancestor for the following object types:

- Applet
- Business component
- Integration object

■ Report

Before doing the repository merge, review new objects you have created and determine whether you want to specify an upgrade ancestor.

To specify an upgrade ancestor

- 1 Navigate to the object in Siebel Tools.
- 2 Click in the Upgrade Ancestor field.
A dialog box appears. It lists available upgrade ancestors.
- 3 Select the desired upgrade ancestor and click Pick.

To view the descendants or copies of an object

- 1 Right-click on an object.
- 2 Select View descendants from the picklist.
A dialog box appears and lists the descendants.

Related Topics

[“About the Siebel Repository Merge” on page 91](#)

[“About Inheriting Upgrade Behavior in a Siebel Upgrade” on page 96.](#)

Configuring Siebel Tools for the Repository Merge

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Databases: Oracle only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

It applies to the Siebel Tools installation that you use to perform the repository merge.

Before the Repository Merge

Perform this task if you plan to run the development environment upgrade in RBO mode.

To configure Siebel Tools before the repository merge

- 1 Exit Siebel Tools.

- 2 Navigate to the Siebel Tools configuration file. It is located in the following directory:

install_dir\bin*lang*\tools.cfg

where

- *install_dir* is the Siebel Tools installation directory
- *lang* is the language in which Siebel Tools is installed, for example enu.

- 3 Open the file with a text editor and locate the section that controls connectivity to the Oracle RDBMS.

In most cases, this is the ServerDataSrc section.

- 4 In this section, set the SqlStyle parameter as follows: SqlStyle = Oracle.
- 5 Save the file and exit.
- 6 Restart Siebel Tools.

After the Repository Merge

Perform this task to reset the SqlStyle parameter to the default, OracleCBO.

To reset Siebel Tools configuration after the repository merge

- 1 Exit Siebel Tools.
- 2 Navigate to the Siebel Tools configuration file you revised before the repository merge. It is located in the following directory:
install_dir\bin*lang*\tools.cfg
where
 - *install_dir* is the Siebel Tools installation directory
 - *lang* is the language in which Siebel Tools is installed, for example enu.
- 3 Open the file with a text editor and locate the section containing the SqlStyle parameter that you revised.
In most cases, this is the ServerDataSrc section.
- 4 In this section, set the SqlStyle parameter back to the default as follows: SqlStyle = OracleCBO.
- 5 Save the file and exit.
- 6 Restart Siebel Tools.

Related Topics

["About the Siebel Repository Merge" on page 91](#)

Performing a Siebel Repository Merge

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows and UNIX only. Does not apply to IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

This task merges customizations in the Prior Customer Repository (your current repository) into the New Customer Repository (the repository in the new Siebel release).

The repository merge typically takes five to seven hours to complete.

Prerequisite: The workstation on which the merge will occur must have a minimum of 512 MB of RAM and at least 2 GB of virtual memory or a 2-GB page file. Inadequate system resources is one of the most common causes of repository merge failures. If your repository includes a large amount of customization, additional memory may be required.

To merge the repository

- 1 Verify that all Siebel Tools projects are checked in and unlocked.
- 2 Close network connections to all repository users and exit Siebel Tools.
- 3 Open the tool s.cfg file in the new Siebel release. It is located in the following directory:

Tools_install_dir\bin\lang

Where

Tools_install_dir is the directory in which Siebel Tools is installed, and *lang* is the language, for example enu.

- 4 Locate the SIEBEL section, and verify that the parameters are set as shown in [Table 34](#).

Table 34. Siebel Tools Configuration File Parameters for Upgrade

Parameter	Value
EnableToolsConstraint	FALSE
SymStrPrefix	X_

- 5 Save the file and close it.
- 6 Navigate to Control Panel > System > Advanced > Performance Settings > Visual Effects.
- 7 Click Adjust for best performance.
- 8 Start Siebel Tools in the new Siebel release, using the following command line option:
Tools_install_dir\bin\si ebdev\editseeddata.

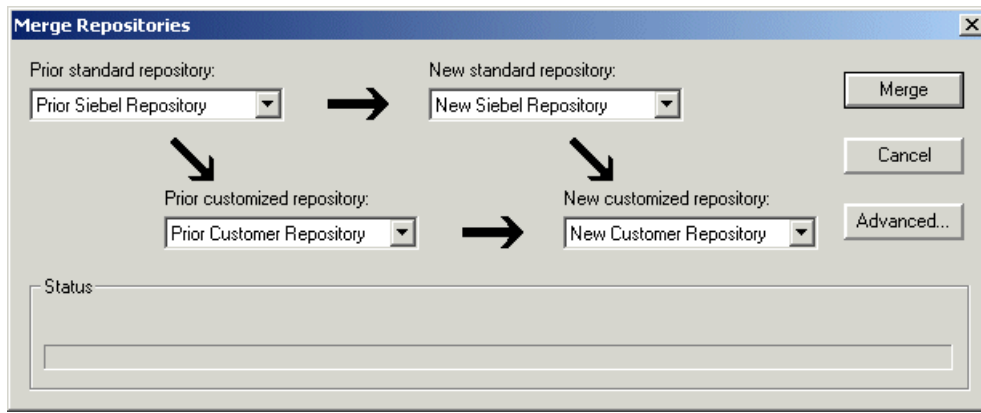
NOTE: The current install directory location for Siebel Tools should be *Tools_install_dir\bin*, as the merge0.txt file will be generated under that path.

- 9 Navigate to View > Options > Language Settings.
- 10 Verify that the language mode setting is set as desired.

This will be the user interface language for SRF files compiled from the New Customer Repository. It will also be the language used by the postmerge utilities.

- 11 Use the File > Open Repository command to open the Prior Customer Repository.
- 12 Choose Tools > Upgrade > Upgrade Application.

The Merge Repositories dialog box appears.



The Merge Repositories dialog box provides four options:

- **Merge.** This button merges the repositories you specify to produce a New Customer Repository.
 - **Cancel.** This button cancels the repository merge and exits the Merge Repositories dialog box.
 - **Advanced.** This button opens the Merge Options dialog box.
- 13 In the Merge Repositories dialog box, choose the repositories listed in the following table.

Drop-Down List Item	Value to Choose
Prior Standard Repository	Prior x.x Siebel Repository, as appropriate for the version from which you are upgrading
Prior Customized Repository	Prior Customer Repository
New Standard Repository	New Siebel Repository
New Customized Repository	New Customer Repository

- 14 Click Advanced.

The Merge Options dialog box appears.

- 15 In the Merge Options dialog box, click the check boxes to activate or deactivate the merge options:

- **Abort merge if more than x errors occur.** Activate this option to abort the repository merge automatically if more than a designated number of errors occur.

CAUTION: The typical Repository merge generates many benign errors. If you select this option, set the number of errors to a large value. This will help prevent the Repository merge from aborting due to benign errors.

- **Incorporate Custom Layout.** Activate this option to help preserve field and button placement on prior custom or modified forms, views, and screens. Select a prior release and style for label placement.

The Prior Release is the release you are upgrading from. The placement of Labels controls where labels are placed in forms. As of Siebel 7.7, label alignment of fields changed. Label vertical alignment changed from top to middle, font weight changed from bold to normal, and text alignment changed from left to right. To preserve the look and feel of releases prior to Siebel 7.7, select Label on Top in the Placement of Labels field.

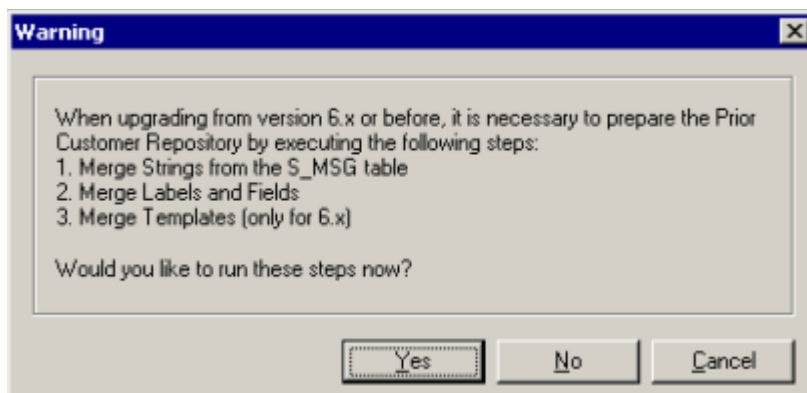
After you have made your selections, click OK.

The Upgrade Check List dialog box appears.

- 16 In the Upgrade Check List dialog box, verify that each prerequisite has been met. When they have been met, put a check mark in all the boxes and click Continue.

CAUTION: The upgrade may fail if all the items in the checklist are not completed.

- 17 If the following Warning screen appears, choose No.



After the merge completes, a dialog displays requesting that you make a backup of the New Customer Repository.

Related Topics

["About the Siebel Repository Merge" on page 91](#)

[“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 102](#)

Reviewing the Siebel Repository Merge Log Files

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

To determine if the repository merge was successful, review the merge log files. The merge is successful if it completes without unacceptable errors:

- **Acceptable errors.** If an ancestor object is specified in an object definition, and the ancestor object is not present in the New Siebel Repository, this causes a merge error. This is an acceptable error and can be ignored.

Here is an example of an acceptable error in the postmerge utilities log file, merge0.txt:

```
!!ERROR: : CANNOT upgrade objects which have Briefing Tracking Profile Appl et - Product marked as 'Upgrade Anc'
```

- **Unacceptable errors.** All other types of merge errors are unacceptable errors and mean that the merge was not successful.

For more detailed information, go to 477269.1 (Doc ID) on [OracleMetalink 3](#). This document was previously published as Siebel Troubleshooting Steps 19.

Merge errors are displayed in the Upgrade Applications Objects List view in Siebel Tools. Additional details on merge errors are located in the repository merge log:

```
Tools_install_dir\bin\merge0.txt
```

where

Tools_install_dir is the directory in which Siebel Tools is installed.

Each time you run the merge process, the name of the merge0.txt file is incremented, for example merge1.txt.

If your repository merge process terminates and is flagged as “Incomplete” under Siebel Tools, Screens > Application Upgrader, the most common reasons for its failure are:

- The number of errors (!!ERROR) exceeds the number that was predefined in Siebel Tools when the merge was started.
- The merge process has been terminated due to a local issue on the Siebel Tools workstation, such as a scheduled reboot.
- RDBMS errors caused the process to stop.
- Memory allotment issues on the workstation on which Siebel Tools is installed.

- Network failure.

If the repository merge is terminated and flagged as 'Incomplete', this should be restarted as discussed in [“Performing a Siebel Repository Merge” on page 252](#).

To determine if the repository merge was successful

- 1 In Siebel Tools, navigate to Screens > Application Upgrader > Application Upgrade Object List.
- 2 In the Application Upgrades list, select the record of the merge.
- 3 Check the entry in the Status column.

- **Completed.** This means the merge completed without errors.

- **Completed with Errors.** This means the merge contains errors.

If the Status column indicates Completed, no further action is required. The merge was successful.

If the Status column indicates Completed with Errors, you must review the errors to determine if the merge was successful. To review the errors, complete the remaining steps in this task.

- 4 In the Object Differences list, click Query.

- 5 In the Status field, enter ERROR::* .

- 6 Press Enter to run the query.

This displays all the objects where the merge process encountered errors.

- 7 Open the merge log file, merge0.txt. It is located in the following directory:

```
Tools_install_dir\bin
```

where

Tools_install_dir is the directory in which Siebel Tools is installed.

If there are multiple files, open the one with the highest number in the file name, for example merge1.txt.

- 8 To locate merge errors in the file, search for the !!ERROR string.

Additionally, informational messages are marked as !!INFO.

- 9 Use the objects displayed in the Object Differences list and the errors displayed in the log file to analyze the errors:

- If all the errors are acceptable, the merge is considered successful.

It is advisable, however, to consider the number of acceptable errors when determining whether to re-run the merge operation.

- If the log contains unacceptable errors, the merge has failed.

- 10** If the merge contains unacceptable errors, go to 477269.1 (Doc ID) on *OracleMetalink* 3. This document was previously published as Siebel Troubleshooting Steps 19.

This document explains the meaning of many of the error messages that can appear in the log file. Use this document to correct the errors. If you cannot resolve all the errors contact Oracle Global Customer Support.

- 11** Open the workflow merge log file:

```
Tools_install_dir\bin\merge0_ver.txt
```

If there are multiple files, open the one with the highest number in the file name, for example merge1_ver.txt. This log file is created by the workflow pre-merge and postmerge steps.

- 12** Review the log file. If the file contains errors, create a service request (SR) on *OracleMetalink* 3, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Related Topics

477269.1 (Doc ID) on *OracleMetalink* 3. This document was previously published as Siebel Troubleshooting Steps 19.

["About the Siebel Repository Merge" on page 91](#)

["Upgrade Planning for Siebel Workflow Designer" on page 152](#)

Generating Siebel EIM Temporary Columns

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

The repository merge process does not preserve EIM processing columns for custom mappings. You must generate the missing custom EIM processing columns again.

To generate EIM temporary columns

- 1** In Siebel Tools, select File > Open Repository, and choose New Customer Repository.
- 2** Navigate to Tools > Upgrade > Generate EIM Processing Columns.
A dialog box displays.
- 3** In the dialog box, click OK to generate EIM processing columns for custom mappings.
- 4** In the Object Explorer window, choose New Customer Repository and verify that the Comment field shows UpgEi mCol .

This indicates that the EIM temporary columns were created successfully.

Related Topics

[“About the Siebel Repository Merge” on page 91](#)

Reviewing Siebel Repository Object Property Conflicts

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

You can change how object property conflicts were resolved during the repository merge.

How Object Property Conflicts Occur

The repository merge compares repository object properties in the Prior Siebel Repository, Prior Customer Repository, and New Siebel Repository. When the value of an object property is different in all three repositories, an object property conflict exists.

This occurs when you have changed the value of an object property in the Prior Customer Repository, and the value of this property has also changed in the new release (New Siebel Repository).

An object property conflict does not occur if you changed the value of an object property in the Prior Customer Repository, and the object property value did not change in the new release. When this happens, the merge process transfers the changed value to the New Customer Repository.

The merge process resolves object property conflicts by referring to the setting of the object's Standard Win property. For about 90% of repository objects, the merge process resolves conflicts by using the object property value in the New Siebel Repository.

Do not change the setting of the Standard Win property.

Application Upgrade Attribute List View

You can review and change how object property conflicts were resolved using the Application Upgrade Attribute List view in Siebel Tools. The Attribute Differences List in the view includes the following columns:

- **Object Name.** The name of the object.
- **Attribute.** The object property name.
- **Conflict.** The merge process puts a check mark in this field if there was an object property conflict during the merge.
- **Resolution.** Displays which property value the merge process used to resolve the conflict:
 - **Standard Value.** The property value in the New Siebel Repository was used. This value is displayed in the New Standard column.

- **Custom Value.** The property value in the Prior Customer Repository was used. This value is displayed in the Prior Customized column.
- **Override.** Put a check mark in this column to change how the conflict is resolved. Overriding the resolution changes the property value in the merged repository. If the resolution was the Standard Value it is switched to the Custom Value and vice versa.

Putting a check mark in the Override column does not change the value displayed in the Resolution column. It indicates that the displayed value was manually overridden in the merged repository.
- **Prior Standard.** Displays the value of the object property in the Prior Siebel Repository.
- **Prior Customized.** Displays the value of the object property in the Prior Customer Repository. In the Resolution column, this value is called the Custom Value.
- **New Standard.** Displays the value of the object property in the New Siebel Repository. In the Resolution column, this value is called the Standard Value.

Prerequisite: The repository merge must have been successful. See [“Reviewing the Siebel Repository Merge Log Files” on page 255](#).

To review object property conflicts

- 1 In Siebel Tools, navigate to Screens > Application Upgrader > Application Upgrade Attribute List.
- 2 In the Application Upgrades list, select the record of the successful merge.
- 3 In the Attribute Differences list, click Query.
- 4 In the Attribute Differences list, click in the Conflict field so that a check mark appears.
- 5 Press Enter to run the query.

The query displays a list of all object properties for which there is a conflict.

- 6 For each record, review the entry in the Resolution field.
- 7 To change the resolution, click in the Override field.

A check mark appears. This changes the value of the object property in the merged repository.

Avoid overriding conflicts for the following object properties. Visually review these properties in the upgraded application before changing them:

- Left, right, top, height, width

Related Topics

[“About the Siebel Repository Merge” on page 91](#)

Regenerating the Siebel Repository Definition Files

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

If you have modified repository objects after the development environment upgrade (upgphys) and before upgrading the production test environment, you must regenerate the schema.ddl and custrep.dat files. These files were created during the upgphys:

■ **Schema.ddl.** This file contains the logical definition of the Siebel Database.

■ **Custrep.dat.** This file contains the definition of repository objects.

These files are used as input to the production test and production environment upgrades. If you modify the object definitions or the schema definitions in the repository after these files have been created, you must regenerate the files.

Regenerating the schema.ddl File

Use this procedure to regenerate the schema.ddl file.

To regenerate the schema.ddl file

- 1 On the Siebel Server where the Siebel Database Server files are installed, navigate to the following location:

Windows: *SIEBEL_ROOT*\bin

UNIX: *\$SIEBEL_ROOT*/bin

- 2 Run the following command:

```
ddl di ct /u DatabaseOwner /p Password /c "ODBCDataSource" /d TableOwner /f
DBSRVR_ROOT\DatabasePlatform\schema.ddl /e y /a y /l SiebelLogDir\sch_di ct. log /n
"Siebel Repository" /t dcir
```

Where

- *DatabaseOwner* is the Siebel Database administrator account name.
- *Password* is the Siebel Database administrator account password.
- *ODBCDataSource* is the ODBC name for connecting to the database. Enclose the name in quotes.
- *TableOwner* is the Siebel table owner name.
- *DBSRVR_ROOT* is the absolute path to the Siebel Database Server installation directory.
- *DatabasePlatform* is the Siebel Database Server directory name for the database, for example Oracle. The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
- *SiebelLogdir* is the path to the directory where you want the output log placed (log output directory). The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.

- 3 After the command completes, review the output log files for errors. If the log indicates there are errors, create a service request (SR) on *OracleMetalink 3*, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Regenerating the `custrep.dat` File

Use this procedure to regenerate the `custrep.dat` file.

To regenerate the `custrep.dat` file

- 1 On the Siebel Server where the Siebel Database Server files are installed, navigate to the following location:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Run the following command:

```
repi mexp /a e /u DatabaseOwner /p Password /c "ODBCDataSource" /d TableOwner  
/r "Siebel Repository" /f DBSRVR_ROOT\DatabasePlatform\custrep.dat  
/l SiebelLogDir\exprep.log
```

where:

- `DatabaseOwner` is the Siebel Database administrator account name.
 - `Password` is the Siebel Database administrator account password.
 - `ODBCDataSource` is the ODBC name for connecting to the database. Enclose the name in quotes.
 - `TableOwner` is the Siebel table owner name.
 - `DBSRVR_ROOT` is the absolute path to the Siebel Database Server installation directory. The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
 - `DatabasePlatform` is the Siebel Database Server directory name for the database, for example Oracle.
 - `SiebelLogdir` is the path to the directory where you want the output log placed (log output directory). The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
- 3 After the command completes, review the output log files for errors. If the log indicates there are errors, create a service request (SR) on *OracleMetalink 3*, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services..

Related Topics

["About the Siebel Database Upgrade Process" on page 58](#)

Setting Label Alignment for Siebel Text Fields

Upgrades from: Siebel 7.x using ICL.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

At Siebel 7.7 label alignment for fields in forms changed. The label vertical alignment changed from top to middle, font weight changed from bold to normal, and text alignment changed from left to right.

If you selected Incorporate Custom Layout (ICL) for the repository merge and chose Label On Top, you must edit the style sheet (main.css) to set label alignment to preserve the look and feel of your previous release.

To ensure Quick Print and Print Preview display correctly, you must also revise the print preview style sheet (printmain.css).

Revising main.css

Revise main.css to display field labels in pre-7.7 format.

To revise main.css

- 1 Using a text editor, open the main.css file in the Siebel Tools installation directory:

```
\publ i c\<lang>\files\mai n. css
```

Where

lang is the installed language, for example enu.

- 2 Search for mceLabel.
- 3 In the .mceLabel section, set the following values as shown:
 - vertical-align : top
 - text-align: left
 - font-weight : bold
- 4 Search for mceLabel2.
- 5 Make the same changes in the .mceLabel2 section that you did in the .mceLabel section.
- 6 Save the file and exit.

Revising printmain.css

Revise printmain.css to display field labels in pre-7.7 format in Quick Print and Print Preview.

- 1 Using a text editor, open the printmain.css file in the Developer Web Client:
`\publ i c\ \ang\files\printmai n. css`
Where
lang is the installed language, for example enu.
- 2 Search for mceLabel.
- 3 In the .mceLabel section, set the following values as shown:
 - vertical-align : top
 - text-align: left
 - font-weight : bold
- 4 Search for mceLabel2.
- 5 Make the same changes in the .mceLabel2 section that you did in the .mceLabel section.
- 6 Save the file and exit.

Related Topics

[“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 102](#)

Copying UI Files to a New Siebel Environment

Upgrades from: Siebel 7.x using ICL. Also applies to non-ICL upgrades where you have manually revised Web template files or the style sheet file (main.css).

Environments: Production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

ICL Upgrades

If you selected Incorporate Custom Layout (ICL) during the repository merge, a special set of Web template files (.swt and .swe) was copied automatically to the webtempl directory in the Siebel Tools installation used to perform the repository merge. For example, if you selected Siebel 7.5.3, new Web template files that provide the look and feel of 7.5.3 were copied to the webtempl directory.

You must copy these files to the Siebel Servers in the production test and production environments.

Customized Web Template Files

When you customize Web template definitions using Siebel Tools, these changes are stored in the Siebel Repository. Siebel Tools does not make changes to the Web template files located in the Siebel Tools installation directory.

If you have manually customized the Web template files in the Siebel Tools installation directory in the development environment, you must copy these files to the Siebel Servers in both the production test and production environments.

Copying Style Sheet Files to a Different Environment

When you customize properties of layout objects such as text fields using Siebel Tools, these changes are stored in the Siebel Repository. Siebel Tools does not make changes to the style sheet file (main.css) located in the Siebel Tools installation directory.

If you have manually customized main.css in the Siebel Tools installation in the development environment, you must copy this file to the Siebel Servers in both the production test and production environments.

You must also copy main.css to the new environment if you have done the following:

- Performed an ICL upgrade
- Selected Label on Top
- Edited the style sheet as described in [“Assigning an Item Identifier to Siebel Web Template Items” on page 291](#).

If you performed an ICL upgrade and selected Label on Top, you must also copy the style sheet for Quick Print and Print Preview (printmain.css) to the new environment.

To copy style sheet files to a different environment

- 1 In the Siebel Tools installation directory (*installdir*) in the development environment, locate the style sheet files:

```
installdir\public\lang\files\main.css and printmain.css
```

where *lang* is the installation language, for example enu.

- 2 Copy the style sheet files to the following location on the desired Siebel Servers in the production test or production environment:

Windows: *SIEBEL_ROOT*\webmaster\files\lang

UNIX: *\$SIEBEL_ROOT*/webmaster/files/lang

To avoid overwriting the existing main.css and printmain.css file, consider changing the existing style sheet's filename before copying in the new one.

- 3 Update the Web server files.

This makes the new style sheet files available to the Web server. To update the Web server files, see *Siebel Installation Guide* for the operating system you are using. You can update the files using a URL and password or by stopping and restarting the Web server.

Related Topics

[“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 102](#)

[“Upgrade Planning for Siebel Web Template Files and Style Sheet” on page 137](#)

Deleting Unneeded Siebel Repository Files

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

When you are confident that the repository has been upgraded successfully, export the New Siebel Repository and Prior Customer Repository for safekeeping. You can also delete the following repositories:

- Prior Standard Repository
- New Standard Repository

Related Topics

[“About the Siebel Repository Merge” on page 91](#)

Migrating Siebel Repository Objects to the Standard UI

Upgrades from: All Supported Siebel releases. Perform this task if you selected Incorporate Custom Layout (ICL) on your previous upgrade.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

If you selected the Incorporate Custom Layout (ICL) merge option for your prior upgrade, UI objects in the New Customer Repository were deactivated and replaced by UI objects from the Prior Customer Repository. Before upgrading to the next release, you must reactivate the standard UI objects. You do this after the development upgrade and before the repository merge.

For example, you upgraded from Siebel 7.5 to Siebel 7.7 and selected the ICL merge option to preserve the Siebel 7.5 UI look and feel. Now you are upgrading from Siebel 7.7 to the latest Siebel Release. Before doing the repository merge, you must activate the Siebel 7.7 standard UI objects. This migrates your UI to Siebel 7.7.

The migration returns the UI to the state it would have been after a regular merge instead of an ICL merge. The following list refers to the Prior Customer Repository from your previous upgrade.

- Customer-Created screens, views and applets are retained.
- UI controls you added in the Prior Customer Repository are retained, but control placements may need to be reconfigured.

- UI controls, list columns, page tabs, charts, applet web template items, and view web template items that you deleted in the Prior Customer Repository remain deleted.
- Customer-Modified screens, views, and applets in the Prior Customer Repository that were further customized afterward are not migrated to the standard UI. Customizations are preserved, but you may need to revise object layouts to conform to the UI standard in the new release.
- Standard objects in the Prior Customer Repository that were not deleted, or modified afterward are migrated to the standard UI. No reconfiguration should be required.
- Standard objects in the Prior Customer Repository that were modified afterward are not migrated to the standard UI. Customizations are preserved, but you may need to revise object layouts to conform to the UI in the new release.

Migrating to the standard UI uses defined logic to select objects and then modifies these objects as follows:

- **ICL objects:** These are the screens, views, applets and child objects that received ICL handling in your previous upgrade. The ICL merge deactivated standard UI objects in the New Customer Repository and replaced them with ICL objects from the Prior Customer Repository. Migration to the standard UI deactivates and deletes ICL objects and activates their counterpart standard UI objects.
- **Standard UI objects.** These are the UI objects that were deactivated by the ICL merge step and have -UPG appended to their names. They contain customizations from the previous repository merge, such as control deletion and addition. They do not contain certain UI layout customizations such as control placement. Migration activates these objects. Standard UI objects replace the ICL objects for the release you are upgrading from. The layouts of some of the standard UI objects will need to be reconfigured.

Using Lag Time to Identify Postupgrade Customizations

ICL objects you have customized after the previous upgrade are not affected by the migration to the standard UI. These changes are preserved and are included in the repository merge for the new release.

Siebel Tools identifies these customizations by comparing the modification time of ICL objects with their corresponding standard UI objects. If the modification times differ by more than a specified lag time, Siebel Tools does not change the ICL object, and it is treated as a customized object in the upcoming repository merge.

For example, your previous ICL repository merge required about three days to complete. This means the modification time of an ICL object and its corresponding standard UI object did not differ by more than three days when the repository merge completed. You later modified an ICL object. Its modification time now differs by more than three days from the corresponding standard UI object.

You then use Siebel Tools to migrate the repository to the standard UI and use a lag time of three days. Since the modification time comparison for the ICL object is greater than the lag time, the ICL object is not replaced by the corresponding standard UI object. The ICL object is treated as a customized object in the upcoming repository merge.

The default lag time is 72 hours. You can specify a lag time between 24 and 120 hours. Observe the following guidelines:

- Avoid setting a lag time that is shorter than the time required to complete the previous repository merge. This can cause objects that were not customized after the merge to be treated as customized.
- Do not set a lag time that is significantly larger than the length of the previous repository merge. This increases the risk that customizations were made before the lag time expired. These customizations are lost during the upcoming repository merge.

How Repository Objects Are Changed

The migration process locates both standard UI objects and ICL objects in the repository. The process then determines whether to modify the standard UI object or the corresponding ICL object:

- Deleting the ICL object and activating the corresponding standard UI object migrates the UI object to standard.
- Deleting a standard UI object and retaining the corresponding ICL object preserves the customized ICL object.

[Table 35 on page 268](#) shows the logic used to modify these objects. Interpret the table columns as follows:

- **ICL Object Found?** Yes means Siebel Tools has located a standard UI object that has a corresponding ICL object. No means that a corresponding ICL object is not in the Siebel Repository. In most cases, this is because the object was inactive in the Prior Customer Repository during the previous merge. When this occurs, the object is appended with -UPG, and no ICL object that preserves look and feel is created.
- **Within Lag Time?** Yes means that the modification time comparison is within the specified lag time. No means the comparison is not within the specified lag time and that you have modified the ICL object after the repository merge.
- **Status in Prior Siebel Repository.** Siebel Tools checks the Active/Inactive status in the Prior Siebel Repository. This prevents activating an object that is inactive in the Prior Siebel Repository.

In the table, Siebel Repository refers to your current Siebel Repository.

Table 35. Logic Used to Modify Repository Objects

ICL object Found?	Within Lag Time?	Status in Prior Siebel Repository	How Migration Modifies Objects in the Siebel Repository
Yes	Yes	Active	<ul style="list-style-type: none"> ■ Deletes the ICL object. ■ Removes the -UPG suffix from the name of the corresponding standard UI object. ■ Changes object status from Inactive to Active in the Prior Customer Repository. ■ Removes the read-only restriction. You can delete or modify the object.
Yes	Yes	Inactive	<ul style="list-style-type: none"> ■ Deletes the ICL object. ■ Removes the -UPG suffix from the name of the corresponding standard UI object. ■ Status is not changed from Inactive to Active in the Prior Customer Repository. ■ Removes the read-only restriction. You can delete or modify the object.
Yes	No	N/A	<ul style="list-style-type: none"> ■ Makes no changes to the ICL object. (The object has been customized after the repository merge.) ■ Deletes the corresponding standard UI object and children of this object.
No	N/A	N/A	<ul style="list-style-type: none"> ■ Status of object is not changed from Inactive to Active. ■ Removes the read-only restriction. You can delete or modify the object.

How Logging Is Done

The migration process writes the changes to the repository in the following log file:

SIEBEL_ROOT\log\icl migration.log

The beginning of the log lists top-level objects that were affected by the ICL feature. For each object, the log then iteratively lists the operations performed on all child objects.

Specifying a Lag Time

The default lag time is 72 hours. You can revise this by editing the Siebel Tools .cfg file. The minimum is 24 hours, and the maximum is 120 hours.

- **Prerequisite:** You must have installed the Siebel Tools that shipped in the new release.

To specify a lag time

- 1 In the Siebel Tools installation directory, navigate to the `\bin\lang` directory, where *lang* is the installed language, for example ENU.
- 2 Using a text editor, open `tools.cfg`, and locate the [Siebel] section.
- 3 Add the following variable at the end of the section:

```
PriorICLMergeTimeLag = time
```

where *time* is an integer greater than or equal to 24 and less than or equal to 120.
- 4 Save the file.
- 5 Restart Siebel Tools.

Migrating to the Standard UI

A menu option in Siebel Tools migrates ICL objects to the standard UI. Upgrading the UI requires a minimum of three hours to complete.

Prerequisites:

- You must have installed the Siebel Tools that shipped in the new release.
- Set the lag time as desired. The default is 72 hours.
- Verify that you have backed up the upgraded database.

To migrate to the standard UI

- 1 Start Siebel Tools.
- 2 In the Tools menu, choose Upgrade, and then select Migrate ICL Objects to Standard.
- 3 In the ICL Migration dialog, make the following selections and click Continue:
 - **Prior Customer Repository:** Select Prior Customer Repository. This is the Siebel Repository in your current release, which you have renamed in preparation for the upgrade.
 - **Prior Standard Repository:** Select Prior V7.X Siebel Repository. This is the Prior Standard Repository for your currently installed release. This repository was loaded when you performed the database upgrade for the new release.

- 4 In the ICL Migration Warning dialog, click Yes to confirm you have backed up the Prior Customer Repository.

The ICL Migration Status dialog box appears. It displays log entries during the migration.

During the migration, you cannot perform other operations in Siebel Tools. You also cannot close Siebel Tools.

A pop-up message displays when the migration is complete.

- 5 To cancel the migration after it has begun, do the following:
 - a In the ICL Migration Status dialog box, click Cancel.
 - b In the dialog box that asks you to confirm you want to cancel, click Yes.

If you cancel the migration after it has begun, you must restore the saved Prior Customer Repository from backup and begin the migration again.

Related Topics

[“About the Siebel Repository Merge” on page 91](#)

[“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 102.](#)

Running the Siebel Postmerge Utilities

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

After the merge is completed successfully, run the postmerge utilities. These utilities make the following changes to user interface objects in the merged repository:

- Convert certain flow-based form applets to grid-based.
- Verify that new or customized screens, views, and applets from the Prior Customer Repository are configured correctly and can be accessed in the UI.
- Revise multi-value group (MVG) applets so that they are shuttle-enabled.
- If you selected Incorporate Custom Layout (ICL) during the repository merge, the utilities list the UI objects that were affected.
- Lists UI objects that have missing required fields.

You can rerun the postmerge utilities as needed. If the postmerge utilities encounter a problem and exit before completing, do the following, fix the problem in the merged repository, and then rerun the utilities.

To run the postmerge utilities

- 1 If you are rerunning the postmerge utilities and want to save the existing log, rename the log. The path to the log is as follows:

SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputi l i ty. l og

Where

SIEBEL_TOOLS_INSTALL_DIR is the directory where Siebel Tools is installed.

If you do not rename the log, it will be overwritten.

- 2 In Siebel Tools, navigate to Screens > Application Upgrader > Application Upgrade Object List.
- 3 In the Application Upgrades list, select the record of the merge.
- 4 In the Application Upgrades list, right-click and select Launch Post Merge Utility from the pop-up menu.

A dialog box appears and displays the postmerge utilities log. When the utilities have finished, a message showing completion displays in the log.

Do not launch more than one instance of the postmerge utilities.

Related Topics

[“About the Siebel Postmerge Utilities” on page 98](#)

[“About the Siebel Repository Merge” on page 91](#)

[“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 102](#)

16 Reviewing the Siebel User Interface

This chapter provides information on reviewing your Siebel UI following a repository merge. This chapter includes the following topics:

- ["Upgrade Flow Orientation" on page 274](#)
- ["Troubleshooting Postmerge Siebel UI Problems" on page 275](#)
- ["Verifying Siebel Business Address Applet Configuration" on page 276](#)
- ["Reviewing Siebel Grid-Based Applets" on page 276](#)
- ["Reviewing Siebel UI Navigation" on page 278](#)
- ["Reviewing Siebel Multi-Value Group \(MVG\) Shuttle Applets" on page 279](#)
- ["Revising Siebel UI Rich Text Controls" on page 282](#)
- ["Reviewing New Siebel UI Aggregate Categories" on page 283](#)
- ["Revising Siebel Visibility Filters to Display Correctly" on page 284](#)
- ["Assigning a Category and Type to Siebel Chart Views" on page 285](#)
- ["Assigning a Category and Type to Siebel Explorer Views" on page 286](#)
- ["Setting Up Navigation to Inaccessible Siebel Detail Views" on page 287](#)
- ["Eliminating Obsolete Siebel UI Fields" on page 288](#)
- ["Reviewing Siebel UI Objects Affected by Incorporate Custom Layout" on page 289](#)
- ["Reviewing Required Fields in the Siebel UI" on page 290](#)
- ["Assigning an Item Identifier to Siebel Web Template Items" on page 291](#)

Related Topics

Chapter 5, ["How the Siebel Database Upgrade Works"](#)

Chapter 4, ["How to Perform a Siebel Database Upgrade"](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase Six of Siebel Database upgrade flow, as shown in [Figure 14](#). At this phase of the upgrade you will review and resolve issues that may have arisen with your user interface during the upgrade process.

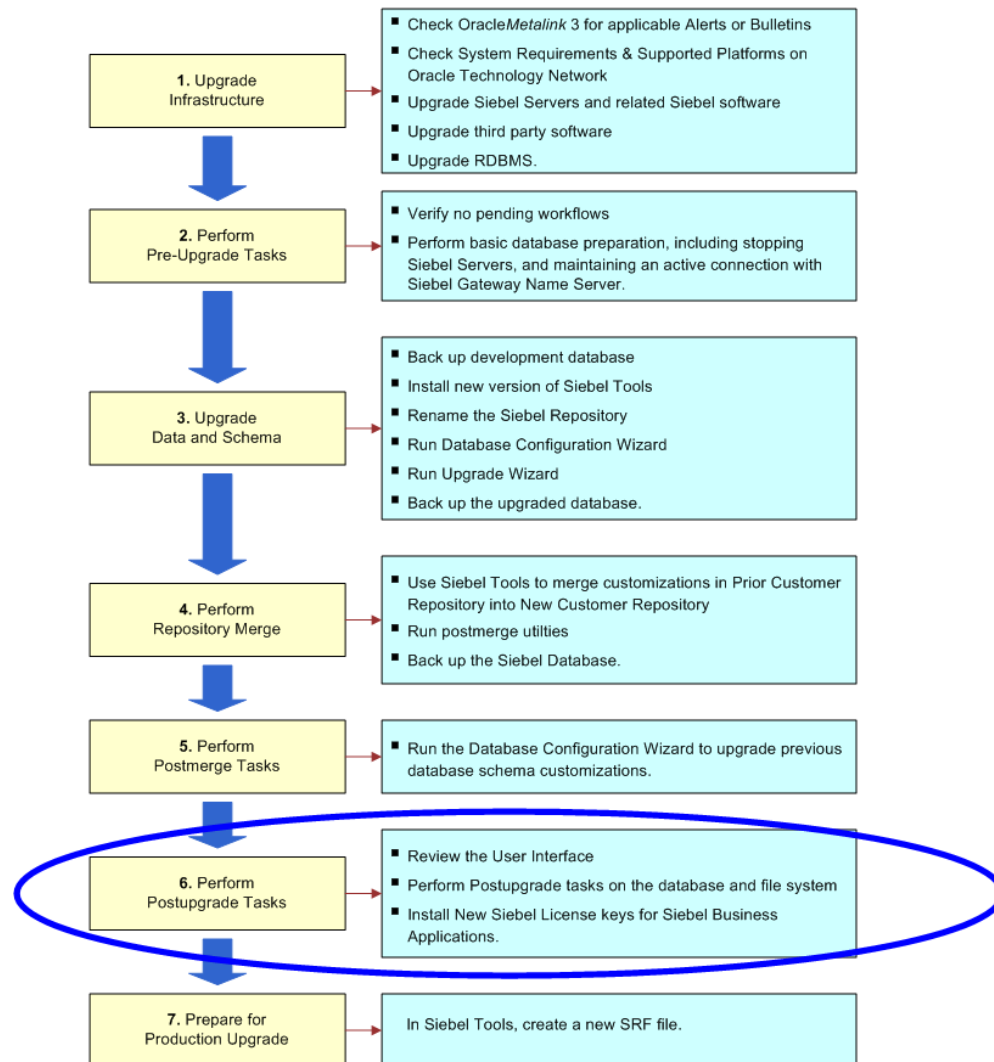


Figure 14. Phase Six of Upgrade Flow

Troubleshooting Postmerge Siebel UI Problems

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Basic Troubleshooting Guidelines

If views or screens do not display, do the following:

- Using the Developer Web Client, verify that the user has been assigned the correct responsibilities.
- Query for the view or applet in the Administration-Personalization screen. Verify the conditional expression is correct. At Siebel 7.7 primary user role was replaced by primary user responsibility. See [“Upgrade Planning for Siebel Personalization” on page 158.](#)
- In Siebel Tools, verify that the screen view and its parents have the `Display In Site Map` property, and the `Display In Page` property set to TRUE.
- Verify that the screen view and its parents have the `Viewbar Text` and `Menu Text` properties filled in.
- If an applet does not display all the fields or controls after upgrade, check in Siebel Tools for a Web template of the same name, but appended with `-Expanded` and specify this Web template for the applet. These templates provide additional placeholders for mapping fields and controls.
- In Siebel 7.5, the `Business Address` business component was replaced with the `CUT Address` business component. If you are upgrading from a release prior to 7.5, you must modify all links to `Business Address` applet so they point to `CUT Address`.

About the Postmerge Utilities Log

After the repository merge, you must run the postmerge utilities. These utilities do the following:

- Validate UI components to verify they were migrated correctly to the new repository.
- Modify UI objects to implement new UI features. For example, they modify form applets and multi-value group applets to conform to the UI standard introduced at Siebel 7.7
- Verify that customized UI objects are configured correctly.

The postmerge utilities log lists the actions performed by the postmerge utilities. The log contains the following types of messages:

- **STATUS.** These messages provide information on the specific things that the postmerge utilities did. No action is required, so you can ignore these messages.
- **INFO.** These messages provide information on the specific things that the postmerge utilities did. No action is required, so you can ignore these messages.

- **WARNING.** These messages provide information on UI objects that may be incorrectly configured, so you should review them.
- **ERROR.** These messages indicate that a problem has been found that must be corrected.

The postmerge utilities log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

Where

SIEBEL_TOOLS_INSTALL_DIR is the directory where Siebel Tools is installed.

Related Topics

[“About Siebel User Interface Changes” on page 128](#)

[“About the Siebel Postmerge Utilities” on page 98](#)

[“About the Siebel Repository Merge” on page 91](#)

[“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 102](#)

477269.1 (Doc ID) on Oracle *Metalink* 3. This document was previously published as Siebel Troubleshooting Steps 19.

Verifying Siebel Business Address Applet Configuration

Upgrades from: Siebel 7.0.x.

Platforms: Windows, UNIX, IBM z/OS.

This topic applies only to Siebel Industry Application (SIA) upgrades. It does not apply to Siebel Enterprise Application (HOR) upgrades.

At release 7.5, the Business Address business component was replaced with the CUT Address business component. You must modify all links to the Business Address applet so they point to CUT Address.

Reviewing Siebel Grid-Based Applets

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Start Invalid Web Template Item Mapping Clean-up.”

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The postmerge utilities help convert many form applets to grid-based layout as of Siebel 7.7.

If a Prior Customer Repository applet that will be converted to grid-based layout has been customized by adding new fields or controls, the utilities put these fields and controls at the bottom of the applet in the New Customer Repository. After the repository merge, you must reposition these fields and controls.

ICL upgrades only: If Incorporate Custom Layouts (ICL) was selected for the repository merge, ICL converts many form applets to grid-based layout. The postmerge utilities then deactivate the grid-based layout and activate the Prior Customer Repository, flow-based form of the applet.

The reputility.log file lists applets that were converted by the postmerge utilities. Review the applets listed in the section of the log referenced above and revise layouts as needed.

The log is located here:

SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log

Where

SIEBEL_TOOLS_INSTALL_DIR is the directory where Siebel Tools is installed.

Table 36 lists common issues and corrective actions for grid-based applets.

Table 36. Common Grid-Based Applet Issues

Log Examples	What To Do
STATUS:: Succeed Update AWTI List Mgmt Lists Entry Applet Base Status	The utility mapped a control to a different location. Action: Review the modified Applet. Verify that the new location for the control works.
STATUS:: Succeed Delete AWTI List Mgmt Lists Entry Applet Base Status ESN	To avoid possible overlapping, the utilities deleted a locale record for an applet Web template item. Action: Review the modified Applet Web Template and re-create the locale record.
WARNING:: Grid -> Grid merge/upgrade, Skip	Only flow-based form applets are converted to the grid-based Web template. Action: None required.
WARNING:: Upgrade Ancestor "Account Form Applet" Not Found in New Siebel Repository, Skip	The utilities did not find the upgrade ancestor's applet Web template in the New Siebel Repository. Action: Review the applet for invalid Web template items.
WARNING:: Applet Web Template Not Found in New Siebel Repository, Skip	The utility is trying to find the Applet Web Template from New Siebel Repository, but it is not found. Action: Review the applet for invalid Web template items.
WARNING:: No List Column Found, Applet "Program Expenditure List Applet", AWT "Edit", AWTI "Type", Control "Type", Skip	No List Column found for the Applet Web Template Item. Action: Delete the applet Web template item.

Table 36. Common Grid-Based Applet Issues

Log Examples	What To Do
WARNING:: No Control Found, Applet "Expense Item Entry Applet", AWT "Edit", AWTI "Site", Control "Site", Skip	No Control is found for the Applet Web Template Item. Action: Delete the applet Web Template.
WARNING:: Button Control, Applet "Expense Item Entry Applet", AWT "Edit", AWTI "NewQuery", Control "NewQuery", Invoke Method "NewQuery", Skip	If a control has an Invoke Method action defined on it, the utility treats it as a button and will not remap it. Action: Review the applet and verify that the button control displays correctly.

Related Topics

- ["Troubleshooting Postmerge Siebel UI Problems" on page 275](#)
- ["About Siebel User Interface Changes" on page 128](#)
- ["About the Siebel Postmerge Utilities" on page 98](#)
- ["About the Siebel Repository Merge" on page 91](#)
- ["About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option" on page 102](#)

Reviewing Siebel UI Navigation

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: "User Interface Navigation Upgrade."

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

The postmerge utilities analyze the repository and verify that objects referenced in screens, views and applets are defined correctly. The reputility.log lists objects that need to be modified. Review the section of the log referenced above and make the needed revisions.

The log is located here:

`SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log`

Where

`SIEBEL_TOOLS_INSTALL_DIR` is the directory where Siebel Tools is installed.

Table 37 shows examples of common issues and corrective actions.

Table 37. Common UI Navigation Issues

Log Examples	What To Do
WARNING:: Project <i>projname</i> is not found in this Repository	The postmerge utilities exclude certain Tools projects from UI Navigation cleanup. One of the excluded projects was not found in the repository. Action: None required.
WARNING:: Ignoring Screen View Record. View Definition Not Found [View: Account Briefing View]	The utility is unable to read the view definition to determine where it should display. Action: Remove the invalid screen view record that references the invalid view.
WARNING:: Error Writing Category Record, Ignoring Changes [Name: <i>catname</i>]	The utility could not update or insert a record in the Siebel Database. Possible causes are that the record already exists or there is a database access problem. Action: Verify that a duplicate <i>catname</i> record does not already exist, and then check database access.
WARNING:: Error Writing Screen View Record, Ignoring Changes [View: <i>viewname</i>]	The utility could not create a screen view record. Possible cause is that the category does not exist. This error is often a consequence of an error updating or inserting a category record. Action: Verify that the screen view category exists, and then check database access.

Related Topics

- [“Troubleshooting Postmerge Siebel UI Problems” on page 275](#)
- [“About Siebel User Interface Changes” on page 128](#)
- [“About the Siebel Postmerge Utilities” on page 98](#)
- [“About the Siebel Repository Merge” on page 91](#)

Reviewing Siebel Multi-Value Group (MVG) Shuttle Applets

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Multivalued Group Shuttle Applet Upgrade.”

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

As of Siebel 7.7, MVG applets are shuttle-enabled by default. The postmerge utilities shuttle-enable MVG applets in the New Customer Repository. This includes MVG applets from the Prior Customer Repository that you created or customized.

MVG applets must have a specific configuration in order to be enabled as MVG shuttle applets. For information on creating and managing MVG shuttle applets, see *Configuring Siebel Business Applications*.

In the reputility.log, the utilities list MVG applets that were converted. Review the section of the log referenced above and resolve any problems encountered during conversion.

The log is located here:

`SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log`

Where

`SIEBEL_TOOLS_INSTALL_DIR` is the directory where Siebel Tools is installed.

Table 38 lists common issues and corrective actions for MVG applets.

Table 38. Common MVG Applet Issues

Log Entry Example	What To Do
WARNING:: [APPLET: Account Address Mvg Applet (NB)] MVG Applet is Inactive. Ignoring Applet.	This applet is inactive. Action: No action required.
WARNING:: [APPLET: Primary Employee Mvg Applet] [Applet Web Template: Base] Applet Web Template is Inactive. Ignoring Applet Web Template.	The applet Web template is inactive. Action: No action required.
WARNING:: [APPLET: FINS Application Contact Mvg Applet - ACAPS] [CONTROL METHOD INVOKED: ExecuteQuery] Has a Non Standard Control Type.	The utility is trying to map an existing Go (ExecuteQuery) button to the Edit List mode to enable Popup Inline Query. However, the Go button does not have the correct attributes. Action: Revise the control definition and map it to the Edit List mode.
WARNING:: [APPLET: State Model - State Mvg Applet] [CONTROL METHOD INVOKED: UndoQuery] Has a Non Standard Control Type.	The utility is trying to map an existing Cancel (UndoQuery) button to the Edit List mode to enable Popup Inline Query. However, the Cancel button defined does not have the correct attributes. Action: Revise the control's definition and map it to the Edit List mode.

Table 38. Common MVG Applet Issues

Log Entry Example	What To Do
<p>WARNING:: [APPLET: LOY Account Address Assoc Applet]</p> <p>[CONTROL: CancelQuery] Control is Inactive. Please inspect and Reactivate.</p>	<p>The utility is trying to map an existing Cancel (UndoQuery) button to the Edit List mode to enable Popup Inline Query. However, the Cancel button defined is inactive.</p> <p>Action: Redefine a Cancel button and map it to the Edit List mode.</p>
<p>WARNING:: [APPLET: LOY Account Address Assoc Applet]</p> <p>[CONTROL: ExecuteQuery] Control is Inactive. Please inspect and Reactivate.</p>	<p>The utility is trying to map an existing Go (ExecuteQuery) button to the Edit List mode to enable Popup Inline Query. However, the Go button defined is inactive.</p> <p>Action: Redefine a Go button and map it to the Edit List mode.</p>
<p>WARNING:: [APPLET: Account Address Mvg Applet] [Applet Web Template: Edit List]</p> <p>[Applet Web Template Item: ExecuteQuery] Applet Web Template Item is Inactive. Please inspect and Reactivate.</p>	<p>The utility is trying to map an existing Go/Cancel button to the Edit List mode to enable Popup Inline Query. However, the mapping already exists, but is marked inactive.</p> <p>Action: Activate the mapping (Applet Web Template Item) in the Edit List mode and test.</p>
<p>WARNING:: [APPLET: Account Address Mvg Applet]</p> <p>[CONTROL: UndoQuery] Applet Web Template Item occupying Item Id 108. Cannot Map Control UndoQuery</p>	<p>The utilities tried to map a Cancel button to the default location 108 in Edit List mode. However, another control is already mapped to this location.</p> <p>Action: Move the control at location 108 to another location, and then map the Go button to location 108.</p>
<p>WARNING:: [APPLET: Assoc Data Type Applet] Association List</p> <p>Applet contains both Base and Edit List. Manual review needed.</p>	<p>An MVG applet definition specifies both a base and edit-list Web template. At Siebel 7.7, the UI standard is that when MVGs first display, they are editable.</p> <p>Action: For MVGs that users can edit, verify that the Web template specified for base mode and edit-list mode are the same. If not, change the base mode Web template so that it is the same as the edit-list template.</p>
<p>WARNING:: [APPLET: Activity Order Mvg Applet] [CONTROL: ExecuteQuery]</p> <p>Control is at an unexpected location. Expected Location is 107</p>	<p>The utilities tried to map a Go button to the default location 107 in Edit List mode. This is part of enabling Popup Inline Query. However, another control is already mapped to this location.</p> <p>Action: Move the control at location 107 to another location, and then map the Go button to location 107.</p>

Related Topics

["Troubleshooting Postmerge Siebel UI Problems" on page 275](#)

["About Siebel User Interface Changes" on page 128](#)

[“About the Siebel Postmerge Utilities” on page 98](#)

[“About the Siebel Repository Merge” on page 91](#)

Revising Siebel UI Rich Text Controls

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Issue 1: Rich Text Controls (RTC) That Need to Have Properties Reconfigured.”

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

As of Siebel 7.7, the configuration of rich text controls (RTCs) changed. The user properties called RTC Graphic Field and RTC Link Field were moved from the Applet User Properties to the Control User Properties of the Body field.

The postmerge utilities review the repository and verify that rich text controls are defined correctly. The reputility log lists the controls that need to be modified. Review the section of the log referenced above and make the needed revisions.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

Where

SIEBEL_TOOLS_INSTALL_DIR is the directory where Siebel Tools is installed.

To revise RTC definitions

- 1 Start Siebel Tools. Set the Object Explorer to Flat.
- 2 Navigate to Views > Options > Object Explorer. Under applet, verify that Applet User Prop and Control User Prop are check-marked.
- 3 Refer to the reputility.log and query for one of the listed applets.
- 4 In the Object Explorer, select Applet User Prop.
If the user RTC Graphic Field and RTC Link Field user properties are marked Inactive, no further action is required.
- 5 For the following user properties, write down the value—active RTC Graphic Field and RTC Link Field user property:
 - RTC Graphic Field. The value for RTC Graphic Field typically is Body Field Graphic.
 - RTC Link Field. The value for RTC Link Field typically is Body Field Link.
 - RTC Body Field. The value is the control name.
- 6 For the applet, select Control > Control User Prop.

- 7 In the Controls list, query for the value you wrote down for RTC Body Field. This is the control name.
- 8 For the control, select Control User Prop.
- 9 Define the following control user properties on the control. Assign the values you wrote down for the applet user properties:
 - RTC Graphic Field
 - RTC Link Field

Related Topics

[“Troubleshooting Postmerge Siebel UI Problems” on page 275](#)

[“About Siebel User Interface Changes” on page 128](#)

[“About the Siebel Postmerge Utilities” on page 98](#)

[“About the Siebel Repository Merge” on page 91](#)

Reviewing New Siebel UI Aggregate Categories

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Issue 2: New Aggregate Category Records That Should Be Renamed.”

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The following new properties were added to the Screen View object in Siebel Tools to support new UI navigation introduced in Siebel 7.7:

- Type (Aggregate Category, Aggregate View, Detail Category, Detail View)
- Parent Category
- Category Name
- Category Default View
- Display in Page
- Display in Site Map

For a description of these properties, see *Configuring Siebel Business Applications*.

If you have created new views or have modified existing views, the postmerge utilities create new Aggregate Category records to support the new properties. The utilities name the new aggregate category records “*busobj_name* List.” For example, a new aggregate category record for the eEvents screen would be named eEvents List.

The reputility log lists the category records that were created by the postmerge utilities. Review the section of the log referenced above and make the following revisions as needed to the listed objects:

- Revise the Viewbar Text and Menu Text properties in all installed languages as required.
- Verify that the navigation hierarchy, including sequence numbers, is correct.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

Where

SIEBEL_TOOLS_INSTALL_DIR is the directory where Siebel Tools is installed.

Related Topics

[“Troubleshooting Postmerge Siebel UI Problems” on page 275](#)

[“About Siebel User Interface Changes” on page 128](#)

[“About the Siebel Postmerge Utilities” on page 98](#)

[“About the Siebel Repository Merge” on page 91](#)

Revising Siebel Visibility Filters to Display Correctly

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Issue 3: Views that need an applet in View Web Template Item ID 1.”

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

In Siebel 7.7, the Show menu was replaced by a visibility filter. This drop-down menu lists alternative views for displaying records.

The postmerge utilities review the repository and verify that filters are defined correctly. The reputility log lists screens that have incorrectly defined filters. The most common problem is that none of the view Web template items has an Item Identifier of 1, which prevents the filter from displaying. Review the section of the log referenced above and make needed revisions.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

Where

SIEBEL_TOOLS_INSTALL_DIR is the directory where Siebel Tools is installed.

To revise visibility filters to display correctly

- 1 Query for one of the views listed in the reputility.log.
- 2 For the view, select Base > View Web Template Item.
- 3 Set the Item Identifier for the first applet in the list to 1.
- 4 Refine the query to display the Parent Category listed in the log.

Related Topics

[“Troubleshooting Postmerge Siebel UI Problems” on page 275](#)

[“About Siebel User Interface Changes” on page 128](#)

[“About the Siebel Postmerge Utilities” on page 98](#)

[“About the Siebel Repository Merge” on page 91](#)

Assigning a Category and Type to Siebel Chart Views

Upgrades from: Siebel 7.0.x and 7.5.x. Exception: this topic does not apply to 7.x upgrades that used the Incorporate Custom Layout (ICL) feature.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Issue 4: Chart Views Needing Migration to Aggregate Type.”

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The Chart menu item is relocated as of Siebel 7.7 and displays as one of the aggregate categories under the view tabs. To ensure that the Chart menu item is located correctly, all the relevant charts for a view must be assigned to the same Aggregate Category. Also, each chart view must be of type Aggregate View. If the chart view is not of type Aggregate View, the chart menu item displays as a view tab.

The postmerge utilities review the repository and verify that Chart views have been defined correctly. The reputeility log lists the screen views that require revision. Review the section of the log referenced above and make the needed revisions.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

Where

SIEBEL_TOOLS_INSTALL_DIR is the directory where Siebel Tools is installed.

To assign a category and type to chart views

- 1 Set Siebel Tools Object Explorer to Types.

- 2 Query for the screen.
- 3 In the Object Explorer, select Screen View.
- 4 Query for the screen views listed in the log.
- 5 Verify that all the views are assigned to the same category (Category Name).
The Category Name can be null.
- 6 Verify that all the views are of type Aggregate View.

Related Topics

["Troubleshooting Postmerge Siebel UI Problems" on page 275](#)

["About Siebel User Interface Changes" on page 128](#)

["About the Siebel Postmerge Utilities" on page 98](#)

["About the Siebel Repository Merge" on page 91](#)

["About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option" on page 102](#)

Assigning a Category and Type to Siebel Explorer Views

Upgrades from: Siebel 7.0.x and 7.5.x. Exception: this topic does not apply to 7.x upgrades that used the Incorporate Custom Layout (ICL) feature.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: "Issue 5: Explorer Views Needing Migration to Aggregate Type."

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

As of Siebel 7.7, explorer views are defined as Aggregate Views. In addition, explorer views use a new Web template, Tree 2.

The postmerge utilities review the repository and verify that explorer views are defined correctly. The reputility.log, lists the explorer views that require revision. Review the section of the log referenced above and make the needed revisions.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

Where

SIEBEL_TOOLS_INSTALL_DIR is the directory where Siebel Tools is installed.

To assign a new Web template to a view

- 1 In Siebel Tools, query for a view name listed in the log.

- 2 In Object Explorer, select View Web Template.
- 3 In View Web Templates, change the Web Template to Tree 2.
- 4 Repeat for all the views listed in the log section.

To assign a new type to a screen view

- 1 In Siebel Tools, query for the screen.
- 2 In Object Explorer, select Screen View.
- 3 For the screen views listed in the log, change the Type to Aggregate View.
- 4 Repeat for each screen listed in the log section.

Related Topics

["Troubleshooting Postmerge Siebel UI Problems" on page 275](#)

["About Siebel User Interface Changes" on page 128](#)

["About the Siebel Postmerge Utilities" on page 98](#)

["About the Siebel Repository Merge" on page 91](#)

["About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option" on page 102](#)

Setting Up Navigation to Inaccessible Siebel Detail Views

Upgrades from: All Supported Siebel releases. Exception: this topic does not apply to 7.x upgrades that used the Incorporate Custom Layout (ICL) feature.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: "Issue 6: Categories Where Parent Applets Are Missing Drilldowns to a Detail View."

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

The postmerge utilities verify that you can navigate from a screen to all the screen detail views listed in the parent category for the screen. If any of the screen's detail views are not accessible using normal navigation methods, the utilities list the screen name, parent category, and the Aggregate View in the log.

In many cases, the problem is caused by a missing or incorrectly defined drilldown in a list applet in the view shown in the log. The missing drilldown prevents the user from navigating to a view containing third-level view tabs that provide access to all the detail views.

Review the section of the log referenced above and make needed revisions.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputi l i t y. l o g
```

Where

`SIEBEL_TOOLS_INSTALL_DIR` is the directory where Siebel Tools is installed.

To set up navigation to inaccessible detail views

- 1 In Siebel Tools, navigate to the screen listed in the log. In Object Explorer, select Screen View.
- 2 In the Screen Views List, query for the following:
 - Type = Detail View
 - Parent Category = the Parent Category listed in the log
- 3 Start the application and navigate to the screen.
- 4 Try to navigate to the detail views listed in the query.

TIP: Use the Web Layout Editor in Tools to identify a detail view containing third-level view tabs that provide navigation to all the detail views. Verify that you can navigate to this view from a drilldown in the screen.
- 5 When you have identified the inaccessible detail view containing third-level view tabs, review the drilldown definitions in Siebel Tools for the list applet in the screen. Define a drilldown to the detail view if one does not exist.

Related Topics

[“Troubleshooting Postmerge Siebel UI Problems” on page 275](#)

[“About Siebel User Interface Changes” on page 128](#)

[“About the Siebel Postmerge Utilities” on page 98](#)

[“About the Siebel Repository Merge” on page 91](#)

[“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 102](#)

Eliminating Obsolete Siebel UI Fields

Upgrades from: All Supported Siebel releases. This topic applies only to upgrades using the Incorporate Custom Layout (ICL) feature.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Issue 7: Fields deprecated from Business Components.”

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Some of the business component fields in your installed release may be obsolete in the new release. If you have customized existing applets or created new ones, and you selected ICL during the merge, the UI may contain obsolete fields or controls. If a business component field is not available for an applet field or control after the merge, the field or control does not display.

The reputility log lists the applets that contain obsolete business component fields. Review the section of the log referenced above and revise applet definitions and layouts as required.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

Where

SIEBEL_TOOLS_INSTALL_DIR is the directory where Siebel Tools is installed.

Related Topics

[“Troubleshooting Postmerge Siebel UI Problems” on page 275](#)

[“About Siebel User Interface Changes” on page 128](#)

[“About the Siebel Postmerge Utilities” on page 98](#)

[“About the Siebel Repository Merge” on page 91](#)

[“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 102](#)

Reviewing Siebel UI Objects Affected by Incorporate Custom Layout

Upgrades from: All Supported Siebel releases. This topic applies only to upgrades using the Incorporate Custom Layout (ICL) feature.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Issue 8: List of the items affected by PCL.”

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

If you selected Incorporate Custom Layout (ICL) for the merge, the postmerge utilities list all the UI objects that were affected by the ICL feature in the reputility log. Use this list to identify the screens, views, and applets you want to review when testing the UI.

The list has two parts:

- **“UI elements changed between prior Siebel Release and current Siebel Release.”** This part lists the screens, views, and applets that have UI changes in the new release. If you selected ICL during the merge, these changes are not reflected in the UI. Instead, the UI for the release you are upgrading from has been preserved.

- **“UI elements changed between prior Siebel Release and prior customer implementation.”** This part lists your customizations. The list includes screens, views, and applets you modified as well as those you created. These customizations are included in the merged repository. If you selected ICL for the merge, the UI for the customizations is the same as the release from which you are upgrading.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

Where

SIEBEL_TOOLS_INSTALL_DIR is the directory where Siebel Tools is installed.

Related Topics

[“Troubleshooting Postmerge Siebel UI Problems” on page 275](#)

[“About Siebel User Interface Changes” on page 128](#)

[“About the Siebel Postmerge Utilities” on page 98](#)

[“About the Siebel Repository Merge” on page 91](#)

[“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 102](#)

Reviewing Required Fields in the Siebel UI

Upgrades from: All Supported Siebel releases. This topic applies only to upgrades using the Incorporate Custom Layout (ICL) feature.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility.log section: “Issue 9: List of Required Fields Missing from the UI.”

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

If you select Incorporate Custom Layout (ICL) during the repository merge, the Web templates for the UI you are upgrading from are used to display the UI after the merge. The Web templates for the new release are not used.

The Web templates used may not contain all the fields required by the new release. For example, a required field may be missing from the Web template used to display the Contacts applet. If you try to create a new record, the database will reject it because the record does not contain all the required fields.

This section of the reputility log lists all the applets that have missing required fields. Use the Web Layout Editor in Siebel Tools to add the required fields.

The log is located here:

`SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log`

Where

`SIEBEL_TOOLS_INSTALL_DIR` is the directory where Siebel Tools is installed.

Assigning an Item Identifier to Siebel Web Template Items

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

All View Web Template items must have a unique item identifier. Do the following to ensure this:

- See Bulletin 476806.1 (Doc ID) on Oracle *Metalink* 3 and resolve duplicate item identifiers. This document was previously published as Siebel Technical Note 400.
- In Siebel Tools, query View Web Templates for a null Item Identifier. Replace any null Item Identifiers with a number. If an item is part of a group of items that have the same grandparent (GParent), the number must be unique in the group.

Related Topics

["Troubleshooting Postmerge Siebel UI Problems" on page 275](#)

["About Siebel User Interface Changes" on page 128](#)

["About the Siebel Postmerge Utilities" on page 98](#)

["About the Siebel Repository Merge" on page 91](#)

["About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option" on page 102](#)

17 Siebel Postmerge Development Tasks

This chapter provides guidelines for developers following a successful Siebel Repository merge. This chapter includes the following topics:

- ["Upgrade Flow Orientation" on page 294](#)
- ["Reviewing Objects Deleted from the Siebel Repository" on page 295](#)
- ["Reviewing Obsolete Objects in the Siebel Repository" on page 295](#)
- ["Upgrading to the Siebel Symbolic String Model" on page 296](#)
- ["Dropping IBM DB2 8-KB Tablespaces and Buffers After a Siebel Merge" on page 297](#)
- ["Updating Siebel Enterprise Application Integration \(EAI\)" on page 297](#)

Related Topics

[Chapter 5, "How the Siebel Database Upgrade Works"](#)

[Chapter 4, "How to Perform a Siebel Database Upgrade"](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase Five of Siebel Database upgrade flow, as shown in [Figure 15](#). At this phase of the upgrade you are performing tasks following a successful Siebel repository merge. the Siebel Repository merge and all associated tasks.

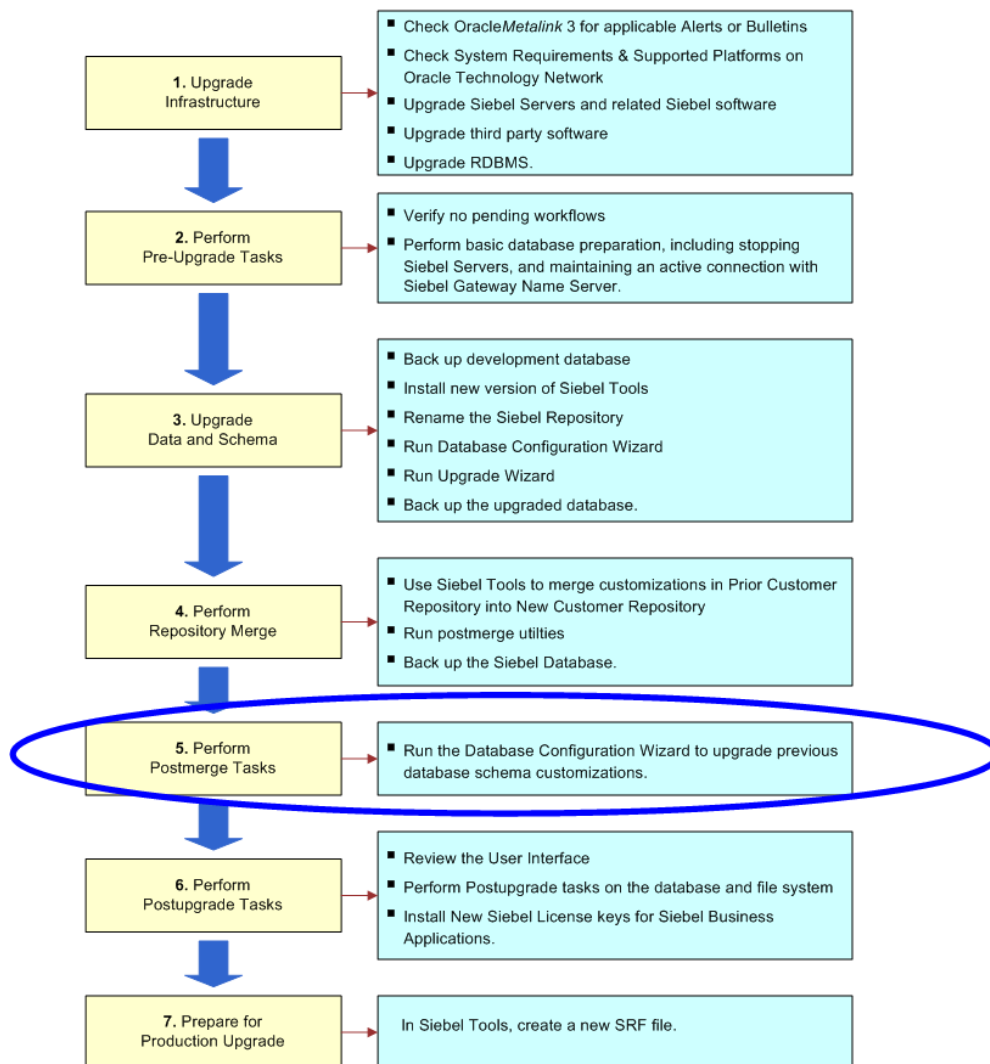


Figure 15. Phase Five of Upgrade Flow

Reviewing Objects Deleted from the Siebel Repository

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

If an object you have deleted from the Prior Customer Repository exists in the New Customer Repository, the repository merge does not delete the object from the New Customer Repository. After the merge, you must review these objects and verify that they do not adversely affect the operation of the application.

Prerequisite: The repository merge must have been successful. See ["Reviewing the Siebel Repository Merge Log Files"](#) on page 255.

To generate a list of deleted objects

- 1 In Siebel Tools, navigate to Screens > Application Upgrader > Application Upgrade Object List.
- 2 In the Application Upgrades list, select the record of the successful merge.
- 3 Click Query.
- 4 Enter your query criteria in the Object Differences list:
 - Click in the In Prior Standard field so that a check mark appears.
 - Click in the Added to New Customized field so that a check mark appears.
 - Click in the In Prior Customized field so that a check mark appears. Then click in it again so that no check mark appears.
- 5 Press Enter to run the query.

Deleted objects appear in the Object Differences list. You can filter the objects displayed by using the Top Parent Type and Object Type fields.
- 6 Review the list carefully to determine that deleted objects that have been restored to the merged repository will not have an adverse effect on upgraded applications.

Related Topics

["About the Siebel Repository Merge"](#) on page 91

Reviewing Obsolete Objects in the Siebel Repository

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Objects that were available in the Prior Siebel Repository but are not available in the New Siebel Repository are obsolete. After performing the repository merge, you can generate a list of obsolete objects using Siebel Tools. Objects that were available in the Prior Siebel Repository are compared with the objects that are available in the New Siebel Repository.

Prerequisite: The repository merge must have been successful. See [“Reviewing the Siebel Repository Merge Log Files” on page 255.](#)

To generate a list of obsolete objects

- 1 In Siebel Tools, navigate to Screens > Application Upgrader > Application Upgrade Object List.
- 2 In the Application Upgrades list, select the record of the successful merge.
- 3 Click Query.
- 4 Enter your query criteria in the Object Differences list:
 - a Click in the In Prior Standard field so that a check mark appears.
 - b Click in the In New Standard field so that a check mark appears. Then click in the field again so that no check mark appears.
 - c Click in the In Prior Customized field so that a check mark appears.
 - d Click in the Attribute field so that a check mark appears. Then click in the field again so that no check mark appears.
- 5 Press Enter to run the query.

All obsolete objects appear in the Object Differences list. You can filter the objects displayed by using the Top Parent Type and Object Type fields.

- 6 Review the list carefully to determine that obsolete objects that have been deleted will not have an adverse effect on upgraded applications.

Related Topics

[“About the Siebel Repository Merge” on page 91](#)

Upgrading to the Siebel Symbolic String Model

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Siebel 7.7 introduced a symbolic string model. It replaces the locale-based string model. In the locale-based string model, each UI text string in the Tools repository is part of a UI object definition, such as an applet. The translations of the text string are maintained as child objects in child locale records for each UI object. For commonly used text strings, this means there is a lot of redundancy between UI objects.

The new symbolic string model is object-oriented. A single symbolic string replaces the text string translations. For each language, a text string is defined and assigned to the symbolic string as an attribute. This simplifies multilingual management of text strings throughout the UI.

Some strings will not be converted to the symbolic string model during upgrade. Seed data, error messages, lists of values (LOVs), and non-translatable attributes (such as the text alignment property on a control) will continue to use locale-based strings.

You must execute a conversion utility (consoleapp) to convert and consolidate your custom locale-based strings to the new model. If you plan to install a language pack, it is recommended that you do so before you run the string conversion or consolidation process.

Instructions for converting or consolidating to the symbolic strings model are found in *Using Siebel Tools*.

Related Topics

[“About the Siebel Repository Merge” on page 91](#)

[“Upgrade Planning for Siebel String Translation” on page 157](#)

Dropping IBM DB2 8-KB Tablespaces and Buffers After a Siebel Merge

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Databases: IBM DB2 only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Drop the 8-KB tablespace, 8-KB temporary tablespace, and 8-KB bufferpool. Before dropping your 8-KB tablespace, check for the existence of any tables in it by running the following SQL statement:

```
select name from sysibm.systables where TBSPACE='TBS_8K'
```

Updating Siebel Enterprise Application Integration (EAI)

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

If you use Enterprise Application Integration (EAI), perform the following procedure to update the definitions of the Business Objects to account for changes in data type, length, edit format or other properties.

To upgrade integration objects

- 1** Determine whether you need to synchronize the integration objects, and synchronize if necessary.

To determine whether you need to synchronize integration objects, review the synchronization considerations in *Integration Platform Technologies: Siebel Enterprise Application Integration*.

- 2** Validate the integration objects.
- 3** If you receive validation errors, deactivate the user keys or fields that cause the error.
- 4** If you receive the error "List Of" in the XML Parent Element, manually remove the value "List Of" from the XML Parent Element.

18 Postupgrade Tasks for the Siebel Database and File System

This chapter provides tasks to be completed within the Siebel Database, and the Siebel File System following a successful upgrade and repository merge. This chapter includes the following topics:

- ["Upgrade Flow Orientation" on page 300](#)
- ["Reapplying Schema Customizations in the Siebel Database" on page 301](#)
- ["Checking for Inactivated EIM Table Columns in the Siebel Database" on page 301](#)
- ["Validating Dock Objects and Rule Definitions in the Siebel Database" on page 302](#)
- ["Verifying an Upgraded Oracle RDBMS After a Siebel Upgrade" on page 303](#)
- ["Setting Oracle Database Parameters After a Siebel Upgrade" on page 304](#)
- ["Updating Siebel File System Attachments" on page 305](#)

Related Topics

[Chapter 5, "How the Siebel Database Upgrade Works"](#)

[Chapter 4, "How to Perform a Siebel Database Upgrade"](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase Six of Siebel Database upgrade flow, as shown in [Figure 16](#). At this phase of the upgrade you will reapply custom schema changes made prior to the repository merge as well as review other issues following the repository merge.

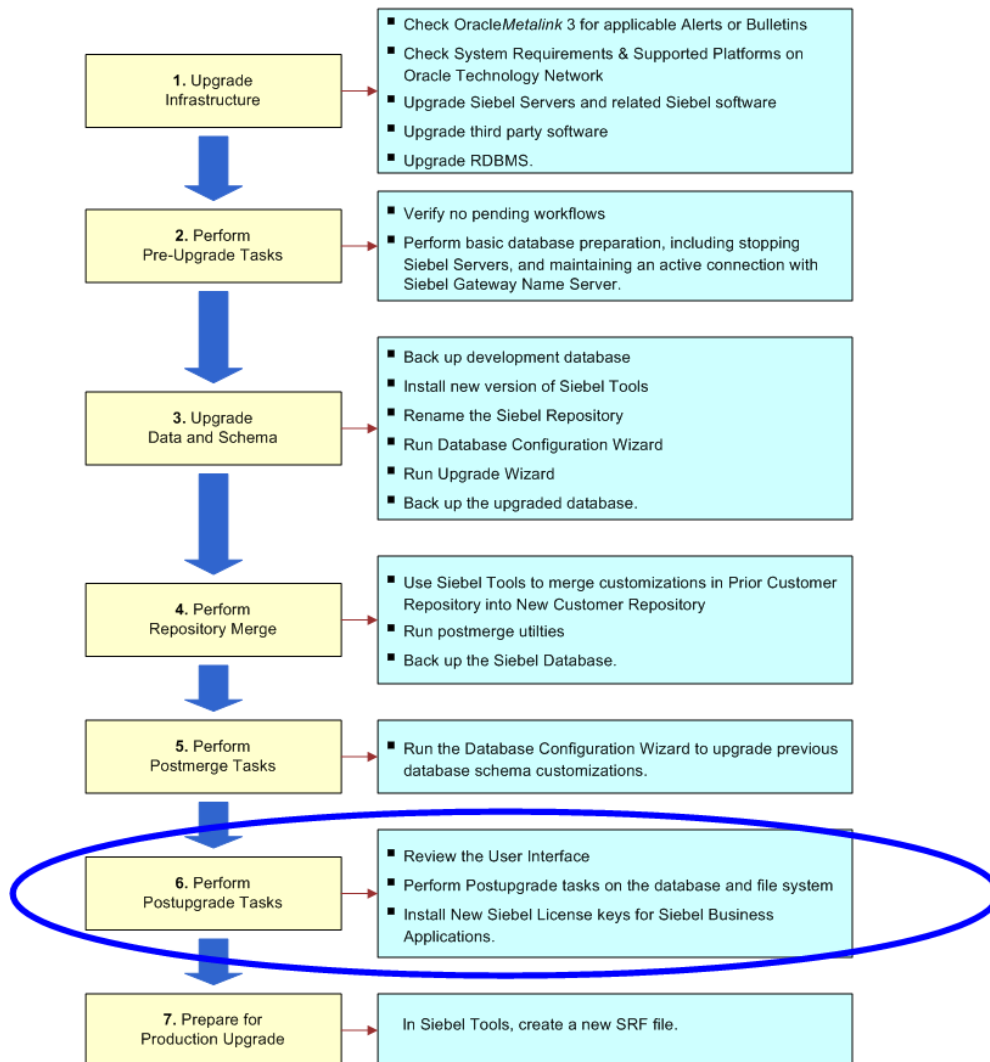


Figure 16. Phase Six of Upgrade Flow

Reapplying Schema Customizations in the Siebel Database

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

In the current release, tables are obsolete or have been replaced by new tables. If you added extension columns or foreign key (FK) columns to tables that are obsolete in the current release, you must reapply these changes to the new tables.

Reviewing Obsolete Tables

The upgrade process generates a report that you can review for information about tables that are either obsolete. This report, `xtndobtbl . txt`, lists the following:

- Custom columns in obsolete tables
- Custom foreign key columns pointing to obsolete tables
- EIM mappings for custom foreign key columns to access-control related obsolete tables
- Workflow columns by custom foreign key to obsolete tables
- Customer denormalized columns on obsolete tables
- Obsolete tables in the current release.

Each obsolete table is listed with one of three codes:

- **Not Used.** These tables are not used in the current release, but you may continue to use them. These tables are supported as is (for instance, with docking or EIM).
- **EOL (end of life).** These tables are not used in the current release, and they are not supported in future releases.
- **Inactive.** These tables have been discontinued, and are not supported in the current release. You must move extension columns and foreign key columns that reside on inactive tables to alternate tables.

If no tables are listed in `xtndobtbl . txt`, no action is required.

If this file lists any tables, reapply their custom extensions and foreign key columns to tables in the current release using Siebel Tools. See *Configuring Siebel Business Applications*.

Checking for Inactivated EIM Table Columns in the Siebel Database

Upgrades from: Siebel 7.0.x and 7.5.x. Also applies to upgrades from Siebel 6.2.1 on IBM z/OS.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

In Siebel 7.7, some columns in EIM tables were deactivated (Active = FALSE). This was done to prevent the tables from exceeding the 32-KB table size limitation. In Siebel Tools, query for inactive columns in these tables and verify that this does not affect the operation of the application.

Validating Dock Objects and Rule Definitions in the Siebel Database

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Changes to visibility rules and dock objects require the assistance of Oracle Advanced Customer Services.

If you deploy Siebel Business Applications to mobile users with local databases, you can run the DICTUTL utility to verify that all dock objects and rule definitions are correct. Dock objects allow mobile users to synchronize their local databases with the Siebel Server. Rules determine which data users synchronize. For more information about dock objects and rules, see *Siebel Tools Online Help* and *Siebel Remote and Replication Manager Administration Guide*.

To verify that all dock object and rule definitions are correct

- 1 Navigate to the following directory:

Windows: `\\SIEBEL_ROOT\bin`

UNIX: `$/SIEBEL_ROOT/bin`

- 2 Type the following command using the parameters specified in [Table 39](#):

```
dictutl /C ODBC_DATASOURCE /U USERNAME /P PASSWORD /D TABLEOWNER /N  
"REPOSITORY_NAME" /A y 2> logfile.log
```

Table 39. Command Line Flags for DICTUTL

Flag	Parameter	Description	Required
/C	ODBC_DATASOURCE	ODBC datasource name	Yes
/U	USERNAME	User name to log in to database	Yes
/P	PASSWORD	User password to log in to database	Yes
/D	TABLEOWNER	User name of tableowner	Yes

Table 39. Command Line Flags for DICTUTL

Flag	Parameter	Description	Required
/N	"REPOSITORY_NAME"	Name of repository for dictionary (the parameter must be bounded within double quotes)	Yes
/A	y or n	Enter the y parameter to ignore the dictionary cache. Enter n if you do not want to ignore the dictionary cache.	Yes

3 Review the *LOGFILE*.log file:

- a** Open the file in Microsoft Excel. If the file is too large for Excel to display the whole file, break the file into two files and examine each separately.
- b** Query for the word error.
- c** If you locate errors, write down the exact error text.
- d** Query for the word syntax.
- e** If you locate syntax errors, write down the exact error text.
- f** To determine if the errors must be resolved, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

The following types of entries indicate that no errors were found:

- ❑ Errors: 0
- ❑ Syntax OK

Related Topics

["Preserving Siebel Dock Objects and Visibility Rules" on page 165](#)

Verifying an Upgraded Oracle RDBMS After a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: Oracle only.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

You can upgrade your RDBMS before or after upgrading to the current Siebel release.

If you upgrade your Oracle RDBMS after upgrading to the current Siebel release, you must run the Siebel ddlimp utility to verify the schema layout of the upgraded database.

This also converts the Siebel Database to character-based length for char and varchar data. For additional information about the following procedure, see 477966.1 (Doc ID) Oracle *Metalink* 3.

To verify the upgraded Oracle RDBMS

- 1 On the Siebel Server where the Siebel Database Server files are installed, navigate to the following location:

Windows: *SIEBEL_ROOT*\bin

UNIX: \$*SIEBEL_ROOT*/bin

- 2 Run the following command:

```
ddl imp /u TableOwner /p TablePassword /c "ODBCDataSource" /f DBSRVR_ROOT/  
DatabasePlatform/schema.ddl /t y /i n /e n /B TableSpace /X IndexSpace /G SSE_ROLE  
/R Y /I SiebelLogDir/ddlctl_veri_fy_RDBMS.log
```

where:

- *TableOwner* is the Siebel table owner name, also called the schema qualifier name.
 - *TablePassword* is the Siebel table owner password.
 - *ODBCDataSource* is the ODBC name for connecting to the database. Enclose the name in quotes.
 - *DBSRVR_ROOT* is the absolute path to the Siebel Database Server installation directory.
 - *DatabasePlatform* is the Siebel Database Server directory for the Oracle Database.
 - *Tablespace* is the Oracle tablespace name for the Siebel Database.
 - *IndexSpace* is the Oracle index space name for Siebel Database.
 - *SiebelLogdir* is the path to the directory where you want the output log placed (log output directory).
- 3 After the command completes, review the output log files for errors. If the log indicates there are errors, create a service request (SR) on Oracle *Metalink* 3. Alternatively, you can phone Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers remain the same and are listed on Oracle *Metalink* 3.

Related Topics

["About Upgrading Your RDBMS in the Siebel Environment" on page 127](#)

Setting Oracle Database Parameters After a Siebel Upgrade

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: Oracle only.

After the Siebel Database upgrade is complete, set the following parameters in `init.ora`:

- **optimizer_index_cost_adj.** Set this parameter to 1.
- **Collecting statistics.** To optimize SQL performance, use the PL/SQL package `dbms_stats` to manage statistics gathering. For additional information on optimizer settings, see Bulletin 478028.1 (Doc ID) on Oracle *Metalink* 3. This document was previously published as Siebel Technical Note 582.
- For a full list of recommended settings for your postupgrade production environment, see the chapter on configuring the RDBMS in *Siebel Installation Guide* for the operating system you are using.

Updating Siebel File System Attachments

Upgrades from: Siebel 6.x, 7.0.x, 7.5.x, 7.7.x, and 7.8.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The following changes have occurred to the structure of the Siebel File System since Siebel 6.x:

At Siebel 7.5, file names of attachments include the attachment table name. If an upgrade migrates the records in an attachment table to a new attachment table, you must run a utility to update the file system attachment names.

Perform the tasks in the following topics to update Siebel File System attachments.

Updating Attachment File Names

The task in this topic applies to all upgrades. Oracle provides a utility to update attachment file names in the Siebel File system. [Table 40](#) lists the input table names to use when you run the utility. The utility updates all files containing the table names you specify. For example, if you specify the tables `S_OLDTABLE_ATT` and `S_NEWTABLE_ATT`, the utility updates the files system attachments by copying all attachment files containing the string `S_OLDTABLE_ATT` to attachment files containing the string `S_NEWTABLE_ATT`.

Table 40. Input Arguments for the Update Utility

Upgrade Path	Old Table	New Table
Upgrades from: Siebel 6.x.	■ <code>S_EMPLOYEE_ATT</code>	■ <code>S_CONTACT_ATT</code>

Table 40. Input Arguments for the Update Utility

Upgrade Path	Old Table	New Table
Upgrades from: Siebel 6.x.	■ S_ORG_INT_ATT	■ S_ACCNT_ATT
<ul style="list-style-type: none"> ■ Upgrades from: Siebel 6.2.1 FINS on all platforms, including z/OS. ■ Upgrades from: Siebel 7.0.x, 7.5.x, 7.7.x, and 7.8.x. Siebel Insurance only. 	■ S_INSCLM_BL_ATT	■ S_INVOICE_ATT

To update file system attachments

- 1 Navigate to the following directory:

Windows: *SIEBEL_ROOT\bin*

UNIX: *\$SIEBEL_ROOT/bin*

- 2 Run the following utility:

Windows: `chg_file_sys.bat OLD_TABLE NEW_TABLE "FILE_SYSTEM"`

UNIX: `chg_file_sys.ksh -s OLD_TABLE -t NEW_TABLE -f "FILE_SYSTEM"`

where:

- *OLD_TABLE* = The name of the attachment table in the release you are upgrading from. This table is obsolete in the new release.
- *NEW_TABLE* = Attachment records in *OLD_TABLE* were migrated to *NEW_TABLE* in the new release. The utility copies file system attachments containing the string *OLD_TABLE* to attachments containing the string *NEW_TABLE*.
- "*FILE_SYSTEM*" = name of the directory where the Siebel File System attachments reside (entered inside quotation marks)

Windows example:

```
chg_file_sys.bat S_EMPLOYEE_ATT S_CONTACT_ATT
"C:\siebel\att"
```

UNIX example:

```
chg_file_sys.ksh -s S_EMPLOYEE_ATT -t S_CONTACT_ATT
-f "/usr/siebel/siebel/att"
```

- 3 Review the renamed files carefully to verify that they can be accessed by Siebel Business Applications.

For example, since *S_EMPLOYEE_ATT* is migrated to *S_CONTACT_ATT*, verify that files such as *S_EMPLOYEE_12-1ABC.SAF* are renamed to *S_CONTACT_12-1ABC.SAF*.

Updating Attachments for S_LIT

Perform the task in this topic if you are upgrading from Siebel 7.0.x. At Siebel 7.5, the records in S_LIT are migrated to S_CB_ASSET_VER. You must update the related file names of attachments from S_LIT*SAF to S_CB_ASSET_VER*.SAF.

To update attachments for S_LIT

- 1 Navigate to the following directory:

Windows: *SIEBEL_ROOT*\bin

UNIX: *\$SIEBEL_ROOT*/bin

- 2 Enter the following command:

Windows: `file_upg_mm.bat ODBC_SOURCE USER_NAME PASSWORD TABLE_OWNER
"FILE_SYSTEM_LOCATION" "SIEBEL_ROOT" "DBSRVR_ROOT"`

UNIX: `file_upg_mm.ksh ODBC_SOURCE USER_NAME PASSWORD TABLE_OWNER
FILE_SYSTEM_LOCATION $SIEBEL_ROOT DBSRVR_ROOT`

where:

- *ODBC_SOURCE* = the ODBC source of the database
- *USER_NAME* = the database user name
- *PASSWORD* = the password for the database user name
- *TABLE_OWNER* = the database tableowner (if your implementation is on iSeries, you can use the value provided for *User_Name*)
- *FILE_SYSTEM_LOCATION* = the directory where the file system resides
- *SIEBEL_ROOT* = the directory where the Siebel Server is installed
- *DBSRVR_ROOT* = the directory where Siebel Database Server files are installed

Windows example:

```
file_upg_mm.bat SEBL sadmi n sadmi npw SIEBEL "C:\siebfie" "C:\sea7xx\siebsrvr"  
"C:\sea7xx\dbsrvr"
```

UNIX example:

```
file_upg_mm.ksh SEBL sadmi n sadmi npw SIEBEL /usr/siebel/siebfie $SIEBEL_ROOT  
/usr/siebel/sea7xx/dbsrvr
```

Note that the UNIX syntax does not use quotes around *FILE_SYSTEM_LOCATION*, and *DBSRVR_ROOT*.

- 3 Review the renamed files carefully to verify that they can be accessed by Siebel Business Applications.

19 Postupgrade Tasks for Siebel Applications

This chapter provides tasks to be performed on Siebel Business Applications following a successful upgrade and repository merge. This chapter includes the following topics:

- ["Upgrade Flow Orientation" on page 311](#)
- ["Generating Siebel Reporting Relationships" on page 312](#)
- ["Setting Up Siebel Global Time Zone Support" on page 313](#)

Siebel ERM

- ["Upgrading Siebel ERM Approval Business Process Workflows" on page 313](#)
- ["Upgrading Siebel ERM Customized Microsite and Group News Pages" on page 314](#)
- ["Migrating Course Duration Information for Siebel Training" on page 316](#)
- ["Upgrading the Launch Field in Siebel Training LOV" on page 316](#)
- ["Upgrading the Test Status in Siebel Training" on page 317](#)
- ["Verifying Class and Session Times in Siebel Training" on page 318](#)

Siebel Marketing

- ["Upgrading Responsibilities in Siebel Marketing" on page 319](#)
- ["Reviewing Siebel Marketing Campaign Data" on page 319](#)
- ["Reviewing Renamed Fields in Siebel Marketing" on page 320](#)
- ["Displaying Regions in Siebel Marketing" on page 321](#)
- ["Revising Program Flowchart Icons in Siebel Marketing" on page 322](#)
- ["Setting Default Campaign Execution Options in Siebel Marketing" on page 322](#)
- ["Upgrading Activity Plans for Programs and Campaigns in Siebel Marketing" on page 323](#)
- ["Upgrading the Newsletter Offer Type in Siebel Marketing" on page 323](#)
- ["Configuring Universal Inbox in Siebel Marketing" on page 324](#)
- ["Configuring Siebel Marketing Purchase Orders for Display" on page 324](#)

Quote and Order Management

- ["Upgrading Siebel Purchase Orders" on page 325](#)
- ["Configuring Siebel Asset-Based Ordering" on page 326](#)
- ["Reviewing Siebel Address Data" on page 328](#)

- [“Upgrading Siebel Attribute Pricing” on page 329](#)
- [“Verifying Aggregate Discounts in Siebel Pricer” on page 329](#)

Workflows

[“Upgrading Inbound Siebel Workflows” on page 331](#)

Siebel Financial Services

[“Migrating Data to the Bankruptcy Status Field in Siebel Financial Services” on page 331](#)

Siebel Universal Customer Master

[“Universal Customer Master \(UCM\) Upgrade Steps After Repository Merge” on page 331](#)

Miscellaneous

[“Migrating Clinical Trip Report Data in Siebel Pharma” on page 335](#)

Related Topics

[Chapter 5, “How the Siebel Database Upgrade Works”](#)

[Chapter 4, “How to Perform a Siebel Database Upgrade”](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase Six of Siebel Database upgrade flow, as shown in [Figure 17](#). At this phase of the upgrade you will review and upgrade your specific Siebel applications, should those applications require any specific upgrade tasks.

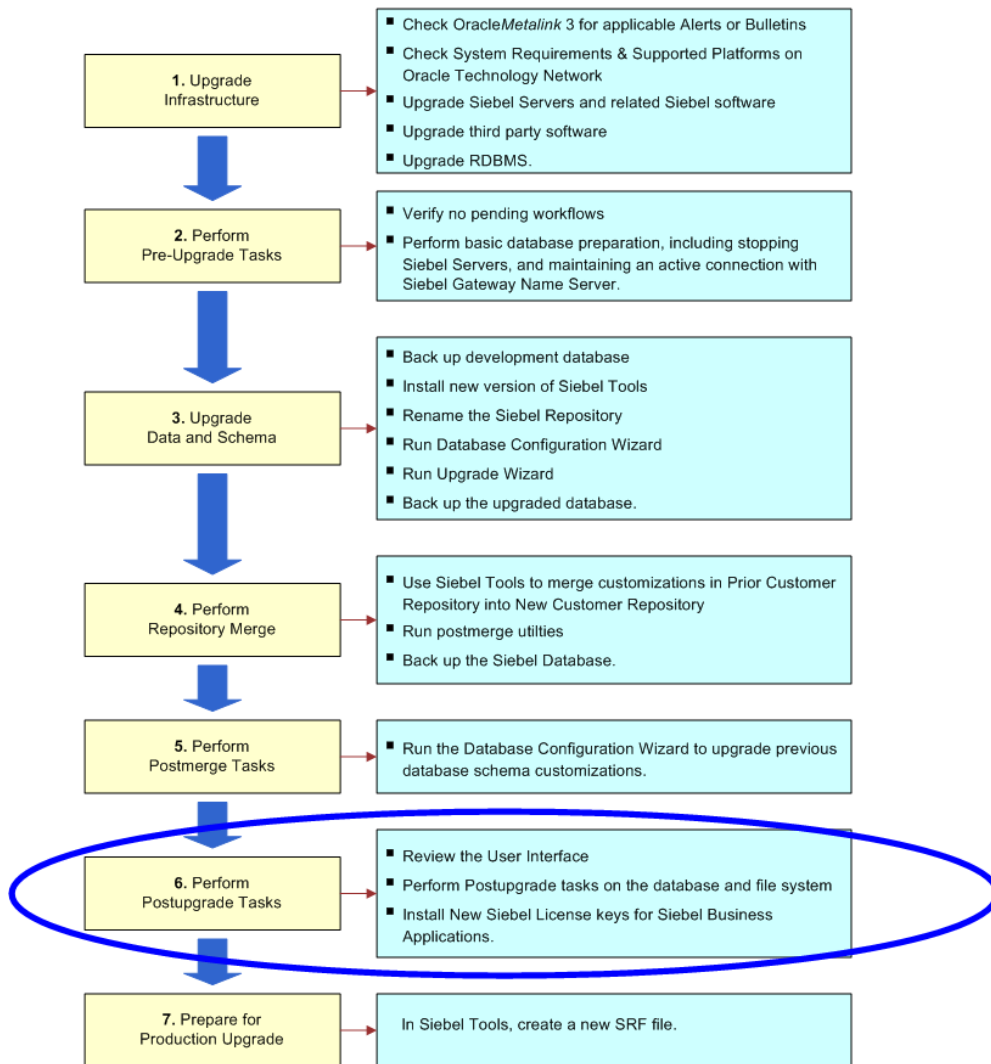


Figure 17. Phase Six of Upgrade Flow

Generating Siebel Reporting Relationships

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The Generate Reporting Relationships process needs to be executed after the upgrade process and whenever the denormalized hierarchy structure (S_PARTY_RPT_REL) becomes out of sync with the data in the normalized tables (S_PARTY). Tables can become out of sync in the following cases:

- After upgrading, the organizational hierarchy (even if there is only one organization) must be established to maintain appropriate visibility in the views cited above.
- When EIM is used to import or update any of the hierarchies (positions, organizations, or access groups).

There are three visibility hierarchies—position, organization, and access groups. These hierarchies are denormalized and maintained in the table S_PARTY_RPT_REL. These denormalized hierarchies are necessary for executing visibility modes that go up or down a hierarchy. For example:

- **Manager view mode.** My Team's Accounts View displays all accounts on which managers and their subordinates are working.
- **Suborganizations view mode.** All Contacts across My Organizations View displays all contacts that are associated to either my organization or any of my organization's suborganizations.

The Generate Reporting Relationships process rebuilds the denormalized relationships in the S_PARTY_RPT_REL table so that the hierarchical view modes display the correct information. The basic operation of the function is to empty the S_PARTY_RPT_REL table and then walk through each S_PARTY record to re-create the denormalized hierarchical structures in the table. This process generates a large number of transactions for Siebel Remote users and regional nodes.

This operation is time and CPU/memory-intensive. The process may take several minutes, depending on the size and complexity of your organizational structures. Do not perform this when you are running other memory-intensive processes.

To generate reporting relationships

- 1 If you have an active Siebel Remote environment, confer with a Oracle Administrator. The Administrator should arrange for the Transaction Processor to be paused before performing this procedure.
- 2 Choose Group Administration under Site Map and navigate to the Positions view in the Siebel Web Client application. Click the Generate Reporting Relationships button in the Position List Applet NB. Note that generating the reporting relationship may cause a large number of Siebel Remote transactions to be generated.
- 3 When this has completed, restart the Transaction Processor.

Setting Up Siebel Global Time Zone Support

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

If you previously upgraded to Siebel 7.x and you enabled your environment for global time zone, you do not need to repeat the procedure. Instead, you need to upgrade data from columns that were *not* UTC-enabled in Siebel 7.x that are UTC-enabled in the current release.

Global deployments typically span multiple time zones. The global time zone feature converts and stores date and time data using the Universal Time Coordinated (UTC) standard. This feature enables you to track dates and times in a common format across multiple time zones.

Although enabling your environment for global time zone is optional in Siebel 7.x, it is strongly recommended that you operate your production environment with global time zone enabled.

For information on setting up and managing UTC, see *Global Deployment Guide*.

Upgrading Siebel ERM Approval Business Process Workflows

Upgrades from: Siebel 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

In Siebel 7.5.3, a number of approval business processes were included in the Sample Database. These were intended for use with Employee Self-Service. If you have activated these workflows, you must manually upgrade the approval steps in them.

Also, if you have designed approval business processes for use with Employee Self-Service, you must manually upgrade the approval steps to convert them to business service operations.

The manual upgrade converts the approval steps from a standard workflow operation to a business process operation. This enables the workflow to use new Universal Inbox Business Service functionality.

Both Siebel Marketing and Siebel ERM use Universal Inbox. Application administrators can see all approval tasks across all applications.

To manually upgrade approval business process workflows

- 1 In Siebel Tools Business Process Designer, locate the desired workflow.

- 2 Change the Business Object of the workflow to UInbox Item Task.
- 3 Retrieve the Priority from the workflow's Extract Events Fields step and save the priority in the process property.
The priority is used to set the Inbox Owner Priority.
- 4 Pass the Priority to the RouteInboxItem function:
 - Input Argument: Task.OwnerInfoTaskPriority
 - Type: Process Property
 - Property Name: Item Priority
- 5 In the workflow diagram, locate all steps that check the approver's status.
- 6 Change the step from a Siebel Operation to a Business Service:
 - Type: Business Service
 - Business Service Name: Universal Inbox
 - Business Service Method: GetInboxOwnerInfoEx
 - Leave the Business Component field blank
- 7 Locate the workflow's Deactivate Inbox Owner step.
- 8 Change the step from a Siebel Operation to a Business Service:
 - Type: Business Service
 - Business Service Method: Universal Inbox
 - Business Service Method: DeactivateInboxOwner
 - Pass in InboxItemId, InboxTypeName, and OwnerPartyId to the method
- 9 Deploy and activate the revised workflow.

Related Topics

["Configuring Universal Inbox in Siebel Marketing"](#) on page 324.

["Upgrade Planning for Siebel Workflow Designer"](#) on page 152

["Upgrade Planning for Siebel Employee Relationship Management \(ERM\)"](#) on page 146

Upgrading Siebel ERM Customized Microsite and Group News Pages

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

As of Siebel 7.7, virtual business components are provided for microsite and Group News pages. These new business components include important rendering optimizations. If you have created customized page sections, you must change the underlying business component.

These old business components are no longer supported:

- ePortal MM Page Item 1
- ePortal MM Page Item 2
- ePortal MM Page Item 3
- ePortal MM Page Item 7
- ePortal MM Page Item 8

They are replaced by the following virtual business components:

- ERM Microsite Section Body VBC
- ERM Microsite Section Navigation Bar VBC
- ERM Section Page Footer VBC
- ERM Section Page Title VBC
- ERM Microsite Section Quick Picks VBC

New Virtual Business Components

In addition, two virtual business components are provided for creating new page sections:

- For microsite pages: ERM Microsite Section VBC (Copy to create a new Section)
- For Group News pages: ERM Group NewsVBC (Copy to create a new Section)

To upgrade customized microsite or Group News pages

- 1 In Siebel Tools, locate the applet for the microsite or Group News page that you want to upgrade. Write down the section code from the applet's search specification.
- 2 Locate the appropriate virtual business component:
 - For microsite pages: ERM Microsite Section VBC (Copy to create a new Section)
 - For Group News pages: ERM Group NewsVBC (Copy to create a new Section)
- 3 Copy this virtual business component and give the copy a new name.
- 4 Activate the new virtual business component.
- 5 In the new virtual business component's user properties, activate MicrositeSection. Set it equal to the section code from the applet.
- 6 Find the applet again and change its business component to the new virtual business component.
- 7 Delete the applet's search specification.
- 8 Change the applet's class to CSSWEFrameListERMPageRender.
- 9 Add the new virtual business component to the views to which the applet belongs.

10 Recompile and deploy.

Additional information on upgrading microsite pages and Group News pages can be found in Bulletins on Oracle*Metalink* 3.

Related Topics

[“Upgrade Planning for Siebel Employee Relationship Management \(ERM\)” on page 146](#)

Migrating Course Duration Information for Siebel Training

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

When upgrading from a release prior to Siebel 7.7, course duration information for Siebel Training is not migrated. To display course duration information you must move course duration to an extension column.

To migrate course duration information

- 1** In Siebel Tools, add an extension column to S_PROD_INT_CRSE. Assign the column the same length and data type as S_SRC_EVT.DURATION_DESC.
- 2** Use an SQL command to transfer existing course duration data to the extension column.
The SQL command should have the following form:

```
update S_PROD_INT_CRSE a  
  
set a.<name of new extension column> = (select b.DURATION_DESC from S_SRC_EVT b)  
  
where a.par_row = b.row_ID)
```
- 3** Expose the extension column in the user interface.
- 4** (Optional) Create corresponding information in the S_PROD_INT_CRSE.CRSE_HOUR_NUM field.

Related Topics

[“Upgrade Planning for Siebel Employee Relationship Management \(ERM\)” on page 146](#)

Upgrading the Launch Field in Siebel Training LOV

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

In Siebel Training, the Distribution Method field is renamed Launch. This field is located in Administration-Training > Training Library.

The list of values for this field was revised in Siebel 7.7. The new values include Download, Launch New Browser, and Launch In-Line.

In previous releases, this LOV contained additional values. If existing records contain these values, you must add these values to the LOV, or the records will not display.

Related Topics

[“Upgrade Planning for Siebel Employee Relationship Management \(ERM\)” on page 146](#)

Upgrading the Test Status in Siebel Training

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

As of Siebel 7.7, completed tests can have one of two statuses:

- Completed-Passed
- Completed-Failed

In previous releases, completed tests had only one status: Completed.

If you want to revise existing data to reflect the new completion statuses, you can write a script to revise the records. The script should use the following logic:

- If at least one test has reached the maximum attempts and not been passed, mark the status as CompletedFailed.
- If all tests have been passed at least once, mark the status as CompletedPassed. Not all test attempts have to be cleared. If there were three attempts allowed and the student failed on the first two but passed on the third attempt, this is acceptable.
- If at least one test has not been attempted, then there should be no change in the status of the registration. So, if a course had a Course Survey associated to it, and if the user had not attempted it yet, its status cannot go to CompletedPassed.
- Else CompletedFailed takes precedence over CompletedPassed.

Related Topics

[“Upgrade Planning for Siebel Employee Relationship Management \(ERM\)” on page 146](#)

Verifying Class and Session Times in Siebel Training

Upgrades from: Siebel 7.0.x, 7.5.x, and 7.7.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

In Siebel 7.8, the way class and session start times and end times are displayed is changed. For existing classes and sessions you must do the following:

- For classes, verify that the times displayed in the Start Date and End Date fields are correct.
- For sessions, verify that the times displayed in the Start and End fields are correct.

To verify class times are correct

- 1 Navigate to Administration-Training > Class Details.
- 2 In the Classes applet, expose the following hidden fields:
 - Start Time
 - End TimeAlternatively, these fields are displayed in the form applet.
- 3 For each class record, compare the time in the Start Time field with the time in the Start Date field.
- 4 Right-click in the Start Date field to adjust the time to match the time in Start Time field as required.
- 5 Compare the time in the End Time field with the time in the End Date field.
- 6 Right-click in the End Date field to adjust the time to match the time in the End Time field as required.
- 7 Repeat this procedure for all the displayed classes.
- 8 Hide the Start Time and End Time fields.

To verify session times

- 1 In the sessions applet, expose the following hidden fields:
 - Start Time
 - End Time

- 2 Select a class in the Classes applet.
- 3 For each session in the Sessions applet, compare the time in the Start Time field with the time in the Start field.
- 4 Right-click in the Start field to adjust the time to match the time in the Start Time field as required.
- 5 Compare the time in the End Time field with the time in the End field.
- 6 Right-click in the End field to adjust the time to match the time in the End Time field as required.
- 7 Select the next class in the Classes applet and repeat this procedure.
- 8 When you have examined all the sessions for all the classes, hide the Start Time and End Time fields.

Related Topics

[“Upgrade Planning for Siebel Employee Relationship Management \(ERM\)” on page 146](#)

Upgrading Responsibilities in Siebel Marketing

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Siebel 7.7 added many new views. In addition, many views from earlier releases are obsolete. You will need to replace any responsibilities you developed for previous releases. A new set of responsibilities are provided in seed data for all active views in Siebel Marketing.

To implement the new responsibilities, use one of the following approaches:

- Add any custom views you create to the seed data responsibilities for Siebel Marketing.
- Create a set of separate responsibilities for any custom views you create. Provide users with the seed data responsibilities as well as your custom responsibilities.

Related Topics

[“Upgrade Planning for Siebel Marketing” on page 146](#)

Reviewing Siebel Marketing Campaign Data

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

At Siebel 7.7, how marketing data is stored was changed. Some Siebel Marketing data is not automatically upgraded:

- Field values from the obsolete Campaign Occurrence header records (from S_SRC) are not merged to the surviving parent campaign. If you have important historical data that is stored on each campaign occurrence, export the list of obsolete campaign occurrence records from the database and determine which field values (if any) you need to apply to the parent campaigns.
- Exported Lists for campaign occurrences are not re-parented to campaign plans. They remain in the obsolete table S_DD_LST_DISTR.
- Campaign expense records (S_SRC_COST) for campaign occurrences are not migrated. This is to prevent campaign expenses from being double-counted after the campaign plan and any campaigns (occurrences) are merged.

During the upgrade, campaign plans and campaigns are merged. This may cause some data to appear double-counted due to re-parenting of similar objects to the same campaign:

- **Program and Campaign Activities.** If Activities are associated to the campaign plan as well as the campaign (occurrence), similar activities may appear twice in the upgraded campaign.
- **Campaign Contacts.** A contact (or prospect) may appear in the same campaign more than once if the campaign member was targeted in multiple occurrences of the same campaign plan. After upgrade, this campaign member will appear in more than one campaign load for the same campaign.

Related Topics

[“Upgrade Planning for Siebel Marketing” on page 146](#)

Reviewing Renamed Fields in Siebel Marketing

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

As of Siebel 7.7, some Marketing applet fields have been renamed. If you have written any customized user documentation, you need to update references to these fields. Renamed fields are shown in [Table 41 on page 321](#).

Table 41. Renamed Fields in Siebel Marketing

Business Component Field Name	Business Component	Original Caption	New Caption
Budget	Program Container	Budget	Assigned Budget
Period	Campaign	Period	Execution Period
Response Type	Campaign	Response Type	Enabled Follow Up Action

Related Topics

[“Upgrade Planning for Siebel Marketing” on page 146](#)

Displaying Regions in Siebel Marketing

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

As of Siebel 7.7, Region is associated with the Region hierarchy rather than the Region LOV field. For this reason, after the upgrade the Region field for Marketing Plans, Programs, and Campaigns does not display.

To display marketing regions

- 1 Create new Marketing Regions under Administration – Location to correlate to the previous LOV values for the Region field.

For more information, see the *Siebel Marketing Installation and Administration Guide*.

- 2 Rename the Region LOV field and add it to the applets in the user interface.

This provides backward compatibility.

Related Topics

[“Upgrade Planning for Siebel Marketing” on page 146](#)

Revising Program Flowchart Icons in Siebel Marketing

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

After an upgrade from releases prior to Siebel 7.7, the icons in the Marketing Program Flowchart view display larger than normal.

To revise the size of Program Flowchart icons

- 1 Right-click in the Program Flowchart applet and deselect Snap to Grid.
- 2 Select the desired icon and reduce its size.

For more information on setting up Program Flowcharts, see the *Siebel Marketing Installation and Administration Guide*.

Related Topics

[“Upgrade Planning for Siebel Marketing” on page 146](#)

Setting Default Campaign Execution Options in Siebel Marketing

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#) It is for upgrades from releases prior to Siebel 7.7. It is not mandatory but may be required by local business processes.

Campaign Execution options are important for controlling the load behavior, launch behavior, assignment behavior, and collaboration options for campaigns. Each Campaign Execution option has an assigned default value. Review your marketing business process to confirm that each Campaign Execution Option is set correctly.

For a discussion of how to set default campaign execution options, see the *Siebel Marketing Installation and Administration Guide*.

Related Topics

[“Upgrade Planning for Siebel Marketing” on page 146](#)

Upgrading Activity Plans for Programs and Campaigns in Siebel Marketing

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

This postupgrade task is not mandatory but may be required by local business processes.

As of Siebel 7.7, the Activity Template type DBM Campaign is obsolete and is not used. Change the template type to Campaign.

To upgrade Activity Plans

- 1 Navigate to Administration--Data > Activity Templates.
- 2 Query for templates of type DBM Campaign.
- 3 For each template, update the type to Campaign.

Related Topics

[“Upgrade Planning for Siebel Marketing” on page 146](#)

Upgrading the Newsletter Offer Type in Siebel Marketing

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

This postupgrade task is not mandatory but may be required by local business processes.

As of Siebel 7.7, the offer type eNewsletter is obsolete. If you have existing offers that are of type eNewsletter, revise the type to Email.

To upgrade the offer type

- 1 Navigate to Offers > All Offers.
- 2 Query for Channel = eNewsletter.
- 3 For each offer, change the offer type to Email.

Related Topics

[“Upgrade Planning for Siebel Marketing” on page 146](#)

Configuring Universal Inbox in Siebel Marketing

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

As of Siebel 7.7, Universal Inbox is implemented as a business service. It provides management of approval queues in both Siebel ERM and in Siebel Marketing. Application administrators can see all the queued tasks in Universal Inbox across all applications.

This means that the Siebel ERM administrator can see tasks queued in Siebel Marketing and vice versa. If this cross-application visibility violates data visibility policies at your site, create a copy of the Universal Inbox. Then assign the copy to the workflows in one of the applications. Use the preconfigured version of Universal Inbox in the other application. This creates physically separate approval queues.

Universal Inbox in Siebel Marketing

As of Siebel 7.7, Siebel Marketing uses a new business service, Universal Inbox. It provides centralized management of approval queues and is used by both Siebel Marketing and Siebel ERM. Application administrators can see all approval tasks across all applications.

Related Topics

[“Upgrade Planning for Siebel Marketing” on page 146](#)

Configuring Siebel Marketing Purchase Orders for Display

Upgrades from: Siebel 7.0.x, 7.5.x, 7.7.x, and 7.8.x.

Environments: Development, production test, production.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

At Siebel CRM 8.0, the relationship between campaign and order is changed from M:1 to M:M and is displayed in a new view, Marketing Purchase Order.

The upgrade process does not set up campaign-related purchase orders created in previous releases so that they display in the new view.

If you want to display purchase orders from previous releases, you must run a script after the upgrade. The script changes the order type from Purchase Order to Marketing Purchase Order for campaign-related purchase orders. You can also use the script to change the order type of campaign-related sales orders to Marketing Purchase Order. This allows these purchase orders to display in the new view.

To configure marketing purchase orders

- 1 Run the following script against the Siebel Database:

Windows: `DBSRVR_ROOT\common\MktgPurchaseOrder.sql`

UNIX: `DBSRVR_ROOT/common/MktgPurchaseOrder.sql`

For campaign-related purchase orders, this script sets order type to Marketing Purchase Order. This allows these purchase orders to display in the Marketing Purchase Order view.

- 2 To configure campaign-related sales orders to display in the new view, open the script with a text editor, and locate the following section:

```
where ORDER_TYPE_ID =
(select ROW_ID from S_ORDER_TYPE
where NAME = 'Purchase Order'
```

Make the following change:

```
where ORDER_TYPE_ID =
(select ROW_ID from S_ORDER_TYPE
where NAME = 'Sales Order'
```

- 3 After making the change, run the script again.

Related Topics

[“Upgrade Planning for Siebel Marketing” on page 146](#)

Upgrading Siebel Purchase Orders

Upgrades from: Siebel 7.0x to 7.5x *only*.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

If you are upgrading from Siebel 7.0 to 7.5, you must run a utility to update the Transaction Amount field in Payment Lines that have Purchase Order as the payment method. The utility requires the .srf that you compiled after upgrading your development environment.

The utility performs the following steps:

- 1 Creates a new Order Entry business object.
- 2 Creates an Order Entry business component and a Payments business component.

- 3 Checks all order records and looks at the Payment Method of corresponding Payment Lines.
- 4 If the Payment Method of a Payment Line is Purchase Order, it updates the Transaction Amount field to Order Total.

To update the Transaction Amount field in Payment Lines

- Enter the following command:

Windows: `SIEBEL_ROOT\bin\pmntupgd /u USERNAME /p PASSWORD /l LANG /c CFG_FILE /d DATA_SOURCE`

UNIX: `$SIEBEL_ROOT/bin/pmntupgd /u USERNAME /p PASSWORD /l LANG /c CFG_FILE /d DATA_SOURCE`

Where

- `USERNAME` is the Siebel user login name
- `PASSWORD` is the Siebel login password
- `LANG` is the language used
- `CFG_FILE` is the configuration file used to launch the application
- `DATA_SOURCE` is the data source used from the .cfg file

Configuring Siebel Asset-Based Ordering

Upgrades from: Siebel 7.0.x and 7.5.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Siebel 7.7 introduced several architectural changes to the asset-based ordering workflows and views:

- **Asset-Based Ordering workflows.** The asset-based ordering workflows have been modified. If you are upgrading from Siebel 7.0.4 or Siebel 7.5, and you have modified these workflows, you must reapply your changes to these workflows after the upgrade.
- **Quote > Orders view.** The Sales Order button now uses a named method to invoke the SIS OM Quote To Order Workflow--PMT Version. You can now pass arguments to the workflow. In Siebel 7.5, this button was hard-coded. If you have written scripts or made other changes regarding the Sales Order button, consider modifying them to take advantage of the named method.
- **Order Header applet.** The Submit button now uses a named method to invoke the SIS OM Submit Order Process. You can now pass arguments to the workflow. In Siebel 7.5, this button was hard-coded. If you have written scripts or made other changes regarding the Submit button, consider modifying them to take advantage of the named method.

- **Disconnect and Modify workflows.** The named methods used to invoke the Disconnect and Modify workflows now pass the additional properties shown in Table 42. If you have written scripts or made other changes regarding these buttons or named methods, you must manually add these after the upgrade.

Table 42. Additional Properties Passed by Named Methods

Property	Explanation
"Due Date", "Today() + 1"	Sets the Due Date on all line items.
"BC Context", "Asset"	Passed as a property to the workflow.
"Compound Product Number", "[Compound Product Number]"	The Compound Product Number is used for Network Order Entry.

- **Delta-In-Place feature.** Delta-In-Place updates the action code of a line item when a specific line item field value is updated. For example, if a user updates the Service ID value of a product in the Order Line Item applet, the action code changes from - (dash) to Update.

In Siebel 7.5, the Delta-In-Place feature was implemented through User Properties on the Quote Item and Order Entry – Line Items business components.

For example:

SIS OM On Field Update Set 6

```
"Service Id", "Action Code", "If ([Action Code] = LookupValue('DELTA_ACTION_CODE', 'Existing'), LookupValue('DELTA_ACTION_CODE', 'Modified'), [Action Code])"
```

In Siebel 7.7, the Delta-In-Place feature was moved to the SIS OM PMT Service workflow. The SIS OM PMT Delta method that is used for changes made to products within the Configurator session is now used to generate the action code for changes to line-item level fields.

If you have added new fields to the Order Management process (quote, order, asset) and then used the SIS OM Field Update Set user property, you must create new user properties in the SIS OM PMT Service to replace these.

For example, if you created a new custom field called Point of Presence, you can delete the old SIS OM On Field Update Set user properties and create a new one as follows:

- **Name:** Delta Line Item Compare Field 26
- **Value:** [Point Of Presence]: [Point of Presence]
- **Siebel Configurator runtime: Done button.** In Siebel 7.0.4 and 7.5, when a user clicks Done in a Siebel Configurator session, Siebel EAI transfers user selections into Quote and Order line items. This process used Quotes, Orders, and Assets-related business components.

As of Siebel 7.7, this process uses new, smaller business components. This improves performance. The new business components begin with "MACD." If you have modified the original business components, you must apply these changes to the MACD business components.

Reviewing Siebel Address Data

Upgrades:

- From Siebel Financial Services 7.0.x to Siebel Industry Applications (SIA) 8.0.x
- From Siebel Business Applications 8.0.x to Siebel SIA 8.0.x
- From Siebel Financial Services 6.2.1 on IBM z/OS platforms to Siebel SIA 8.0.x on IBM z/OS platforms

NOTE: This topic does not apply to Siebel Business Applications (HOR) that you are upgrading to a later release of Siebel Business Applications (HOR).

Environments: Production test, production.

Databases: All databases.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The way address data is stored changed in Siebel 7.7. The upgrade migrated data from the table S_ADDR_ORG to S_ADDR_PER. The upgrade uses the following method for preserving address data:

- If multiple records have the same value for ADDR_NAME within or across S_ADDR_ORG and S_ADDR_PER, the upgrade process preserves ADDR_NAME for one of the records and appends the ROW_ID to ADDR_NAME in the others.
- The upgrade process truncates the value of ADDR_NAME to fit the size of the column.
- The upgrade generates a report listing the records with a duplicate ADDR_NAME within and across S_ADDR_ORG and S_ADDR_PER.

After the upgrade completes, review this report and edit or delete records from S_ADDR_PER as desired.

NOTE: This report is also generated for Siebel 7.0.x Siebel Industry Solutions upgrades to Siebel 7.7 Siebel Industry Applications. Ignore this report for these upgrades.

To review address records after upgrade

- 1 Review the report generated by the upgrade:
 - Windows: `SIEBEL_ROOT\log\rpt_dup_addr_names.txt`
 - UNIX: `$SIEBEL_ROOT/1og/rpt_dup_addr_names.txt`
- 2 Use the report to identify records that are duplicates.
- 3 Use the application or EIM to delete or revise records as needed.

Related Topics

[“Upgrade Planning for Migrating Siebel Address Data” on page 144](#)

Upgrading Siebel Attribute Pricing

Upgrades from: Siebel 7.0.x, 7.5.x, and 7.7.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

At Siebel 7.8, the attribute adjustments feature replaces attribute pricing in Siebel Pricer.

You must manually upgrade attribute pricing data to attribute adjustments by running a business service method. The business service method does the following:

- Upgrades attribute pricing headers to attribute adjustment headers
- Upgrades attribute pricing attributes to attribute adjustment dimensions
- Upgrades attribute pricing values to attribute adjustment dimension domains
- Upgrades attribute pricing adjustment items to attribute adjustment rules

To upgrade attribute pricing to attribute adjustments

- 1 Launch Siebel Sales.
- 2 Verify that all attribute classes have been upgraded to product classes.
- 3 Navigate to Administration > Business Services > Simulator.
- 4 Create a new record:

Field	Value
Service Name	PSP Pricer Upgrade
Method Name	UpgradeDynamicMatrix

- 5 Click Run to start the business method.

Related Topics

[“Upgrade Planning for Siebel Pricer and Order Management” on page 159](#)

Verifying Aggregate Discounts in Siebel Pricer

Upgrades from: Siebel 7.0.x, 7.5.x, and 7.7.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

At Siebel 7.8, the aggregate discounts feature replaces bundle factors in Siebel Pricer. The bundle factor definitions are upgraded to aggregate discounts, and the sequencing of bundle factors are upgraded to aggregate discount sequences.

The name of the aggregate discount in Pricer will be set to *bundle factor name + row ID of the record*. This is because Pricer requires the aggregate discount name to be unique.

Sequencing of bundle factors within a pricing model is upgraded to aggregate discount sequencing. The name of the aggregate discount sequence is set to the pricing model name that contained the bundle factors.

The Price List and Price List Item will be stamped with the appropriate aggregate discount sequence name. In prior releases, the pricing model was specified at the Price List or, for customized products, at the Price List Line Item level. This ensured execution of the bundle factors at runtime.

At Siebel 7.8, the execution of aggregate discounts at runtime requires the association of the aggregate discount sequences at the Price List or Price List Line Item level.

The upgrade process makes the following assumptions about Pricer implementations prior to Siebel 7.8:

- Flowcharts were used to chain up bundle factors in the pricing model
- The bundle factor with the lowest sequence is connected to the "Aggregate Start" step
- Each Aggregate Start sequence contains only bundle factors and does not contain aggregate factors
- The next factor in the flowchart (when True or False) always has a larger sequence number.

If your implementation does not meet all the above criteria, the upgrade process moves the definitions to the appropriate Pricer entities (such as aggregate discounts), but the sequences will not be correct.

In such cases, you must manually verify that aggregate discount sequences chain up the aggregate discounts as intended. Use the sequence of execution that existed prior to the upgrade.

To verify upgrade to aggregate discounts

- 1** Launch Siebel Sales.
- 2** Navigate to Administration – Pricing > Aggregate Discount Sequences.
- 3** For each aggregate discount sequence, drill down to the detail view.
- 4** Locate the aggregate discount that corresponds to the first pre-upgrade bundle factor. Verify that it has the lowest sequence number. If not, revise the numbers in the Sequence, Next Discount If Used, and Next Discount If Not Used columns.
- 5** Verify that the numbers in the Next Discount If Used and Next Discount If Not Used columns are greater than the number in the Sequence column. Also verify that they point to the expected aggregate discounts. If not, revise the numbers in all three columns as required.

Related Topics

[“Upgrade Planning for Siebel Pricer and Order Management” on page 159](#)

Upgrading Inbound Siebel Workflows

Upgrades from: Siebel 7.0.x.

Environments: Development, production test, production.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Change inbound workflows that contain a “String” type process property to pass the value into type Binary; otherwise, the workflow presents the following error message:

```
Output argument '<Value>' in step 'Read from File' contains data that cannot be passed
to string type property 'InputXML'. Data type: 'MEMBLOCK'; String representation of
data body: '<?xml version="1.0" encoding="UTF-8"?><?'
```

Related Topics

[“Upgrade Planning for Siebel Workflow Designer” on page 152](#)

Migrating Data to the Bankruptcy Status Field in Siebel Financial Services

Upgrades from: Siebel 7.0.x, 7.5.x, and 7.7.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

At Siebel 7.8, a BK_STATUS_CD column is provided in the S_BANKRUPTCY table. This column stores Bankruptcy status information that is used by the Siebel Financial Services application.

If you have implemented a bankruptcy status field in the release you are upgrading from, use an SQL command to migrate the data from the custom extension column to BK_STATUS_CD.

Universal Customer Master (UCM) Upgrade Steps After Repository Merge

Upgrades from: UCM 7.8.2 to 8.0

Environments: Development environment only.

Siebel UCM 7.8.2 is shipped with .sif files in addition to the standard 7.8.2 repository. In the Siebel 7.8.2 to 8.0 upgrade process, the Prior Standard Repository is a standard Siebel 7.8.2 repository which does not contain the UCM 7.8.2 additions gained by applying the UCM .sif file. This inconsistency may cause some false conflicts and the 8.0 changes may be overridden by the repository merge conflict resolution process. Some of the mis-picked values may cause functional impact in various UCM areas, such as UI, source data and the history, batch manager and so on. To retain out of the box 8.0 functionality for UCM 8.0, you need to perform the following procedure.

To change the Resolution field in the Application Upgrade Attribute list

- 1 Log in to Siebel Tools following your repository merge.
- 2 From the application menu, click Screens > Application Upgrader > Application Upgrade Attribute List.
- 3 In the Attribute Differences list, find the objects listed in [Table 43](#) and change the value in the Resolution field from “Use Custom Value” to “Use Standard Value.”

Table 43. UCM Objects prior to Resolution field value change

Object Name	Top Parent Name	Top Parent Type	Object	Attribute	Prior Customized Value
UCM Account Address Source Data and History	UCM Account Address Source Data and History	Business Component	Business Component 'UCM Account Address Source Data and History'	Search Specification	[Object Name] = "CUT Address"
UCM Contact Address Source Data and History	UCM Contact Address Source Data and History	Business Component	Business Component 'UCM Contact Address Source Data and History'	Search Specification	[Object Name] = "Personal Address"
UCM Account History Flat List View	CIF Administration Screen	Screen	Screen View 'UCM Account History Flat List View' of Screen 'CIF Administration Screen'	Menu Text - String Reference	SBL_ACCOUNT_SOURCE_DATA_UCM

Table 43. UCM Objects prior to Resolution field value change

Object Name	Top Parent Name	Top Parent Type	Object	Attribute	Prior Customized Value
UCM Account History Flat List View	CIF Administration Screen	Screen	Screen View 'UCM Account History Flat List View' of Screen 'CIF Administration Screen'	Viewbar Text - String Reference	SBL_ACCOUNT_SOURCE_DATA_UCM
UCM Contact History Flat List View	CIF Administration Screen	Screen	Screen View 'UCM Contact History Flat List View' of Screen 'CIF Administration Screen'	Menu Text - String Reference	SBL_CONTACT_SOURCE_DATA_UCM
UCM Contact History Flat List View	CIF Administration Screen	Screen	Screen View 'UCM Contact History Flat List View' of Screen 'CIF Administration Screen'	Viewbar Text - String Reference	SBL_CONTACT_SOURCE_DATA_UCM
Yes	UCM Batch Process: 3	Workflow Process	WF Step Branch 'Yes' of WF Step 'Last Page?' of Workflow Process 'UCM Batch Process: 3'	Expression	(LastPage All Must Match (Ignore Case)('true'))

- 4 After changing the Resolution field value from “Use Custom Value” to “Use Standard Value” the objects will reflect the values displayed in [Table 44](#).

Table 44. UCM Objects after Resolution field value change

Object Name	Top Parent Name	Top Parent Type	Object	Attribute	New Standard Value
UCM Account Address Source Data and History	UCM Account Address Source Data and History	Business Component	Business Component 'UCM Account Address Source Data and History'	Search Specification	[Parent Type] = 'Account'
UCM Contact Address Source Data and History	UCM Contact Address Source Data and History	Business Component	Business Component 'UCM Contact Address Source Data and History'	Search Specification	[Parent Type] = 'Contact'
UCM Account History Flat List View	CIF Administration Screen	Screen	Screen View 'UCM Account History Flat List View' of Screen 'CIF Administration Screen'	Menu Text - String Reference	SBL_ACCOUNTS-1004224419-01T
UCM Account History Flat List View	CIF Administration Screen	Screen	Screen View 'UCM Account History Flat List View' of Screen 'CIF Administration Screen'	Viewbar Text - String Reference	SBL_ACCOUNTS-1004224419-01T
UCM Contact History Flat List View	CIF Administration Screen	Screen	Screen View 'UCM Contact History Flat List View' of Screen 'CIF Administration Screen'	Menu Text - String Reference	SBL_CONTACTS-1004224924-40K

Table 44. UCM Objects after Resolution field value change

Object Name	Top Parent Name	Top Parent Type	Object	Attribute	New Standard Value
UCM Contact History Flat List View	CIF Administration Screen	Screen	Screen View 'UCM Contact History Flat List View' of Screen 'CIF Administration Screen'	Viewbar Text - String Reference	SBL_CONTACTS-1004224924-40K
Yes	UCM Batch Process: 3	Workflow Process	WF Step Branch 'Yes' of WF Step 'Last Page?' of Workflow Process 'UCM Batch Process: 3'	Expression	

Migrating Clinical Trip Report Data in Siebel Pharma

Upgrades from: Siebel 7.0.x, 7.5.x, 7.7.x, and 7.8.x.

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

In Siebel 7.8.x and earlier releases, clinical trip report data are stored in S_EVT_ACT with joins to other tables including S_ORG_EXT, S_CONTACT for account name, contact name, and so on. Values in these joined fields can be changed, hence the trip data are dynamic. This is not compliant with government regulations that require all data in a clinical trip report to be static when the report record is locked.

In Siebel CRM 8.0, a new S_EVT_ACT extension table S_CL_TRP_RPT_LS stores the static data. Run the following script to populate S_CL_TRP_RPT_LS for existing trip report records in S_EVT_ACT:

Windows: `DBSRVR_ROOT\common\Ci ni cal Upgrade. sql`

UNIX: `DBSRVR_ROOT/common/Ci ni cal Upgrade. sql`

If you have already implemented a custom extension table or extension columns to store static data, modify the script to move the data to S_CL_TRP_RPT_LS.

20 Tuning the Siebel Upgrade Files

This chapter provides guidelines for preparing your Siebel Application data for the Siebel Database upgrade. This chapter includes the following topics:

- ["Upgrade Flow Orientation" on page 338](#)
- ["Starting and Stopping Siebel Upgrade Tuner" on page 338](#)
- ["Managing Parallel Threads Using Siebel Upgrade Tuner" on page 340](#)
- ["Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner" on page 343](#)
- ["Transferring UNIX Files for Use by Siebel Upgrade Tuner" on page 346](#)
- ["Rolling Back Siebel Upgrade Tuner Changes" on page 349](#)

Related Topics

[Chapter 5, "How the Siebel Database Upgrade Works"](#)

[Chapter 4, "How to Perform a Siebel Database Upgrade"](#)

Upgrade Flow Orientation

The tasks in this chapter belong to Phase Six of Siebel Database upgrade flow, as shown in [Figure 18](#). At this phase of the upgrade you will run the Siebel Upgrade Tuner.

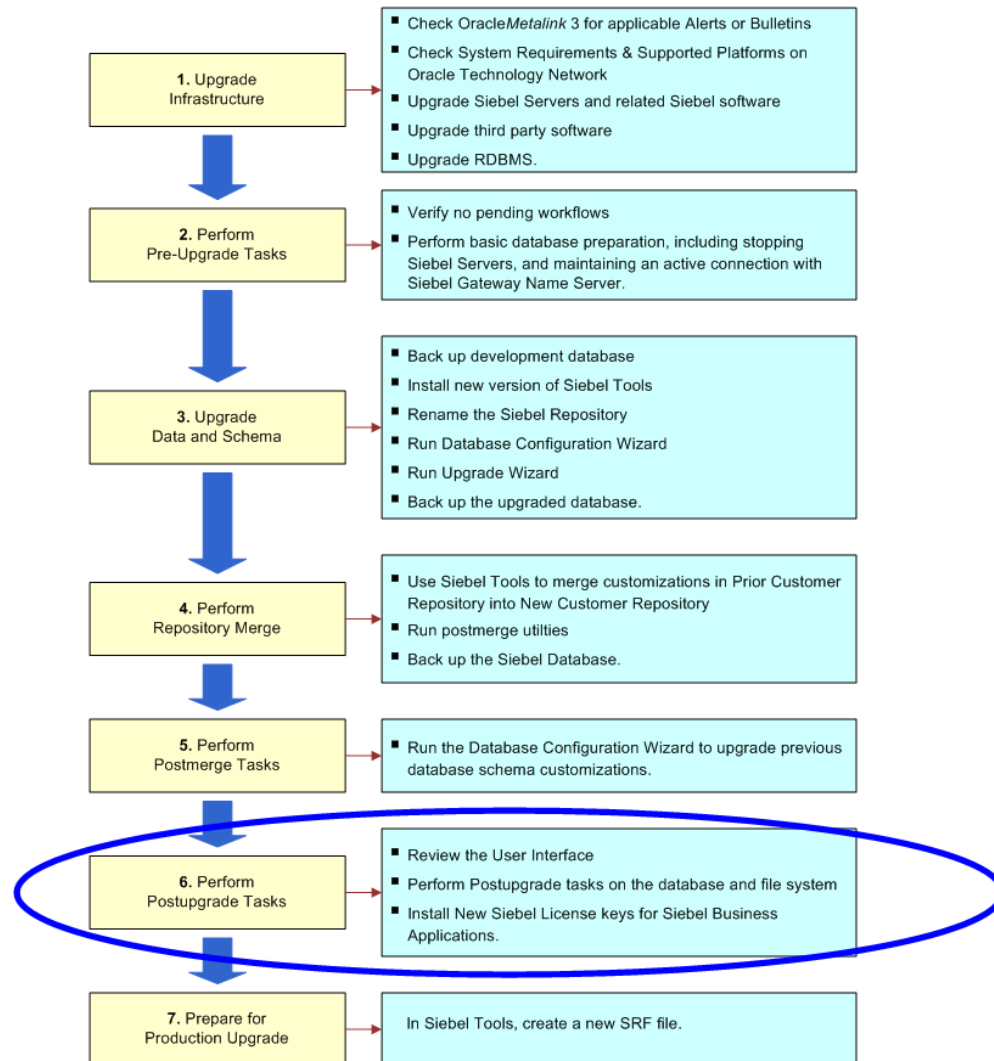


Figure 18. Phase Six of Upgrade Flow

Starting and Stopping Siebel Upgrade Tuner

Upgrades from: All Supported Siebel releases.

Environments: Production test environment only. Does not apply to production environment.

Platforms: Windows and UNIX only.

Databases: All databases.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

This topic describes how to start the Upgrade Tuner (upgtuner) from the Siebel Server installation directory on a Windows host. It also describe how to save or discard your changes when exiting Upgrade Tuner.

Use the Upgrade Tuner to improve the performance of table creation, index creation, and SQL execution during the production upgrep step in the production test environment.

Prerequisites

- In the production test environment, you must have completed the production upgrep step.
- You must have run the logparse utility on the production upgrep log files. See [“Summarizing Siebel Log Files Using the Logparse Utility” on page 239.](#)
- For UNIX platforms, you must have transferred files to a Windows host on which a Siebel Server is installed. See [“Transferring UNIX Files for Use by Siebel Upgrade Tuner” on page 346.](#)

Starting Upgrade Tuner

Use this procedure to start Upgrade Tuner.

To start Upgrade Tuner

- Enter the following command:

```
SIEBEL_ROOT\bin\upgtuner /c LOGPARSE_SUMMARY_LOCATION
```

where:

LOGPARSE_SUMMARY_LOCATION = location of summary.xml generated by the log parse utility for the production upgrep.

For example:

```
upgtuner /c SIEBEL_ROOT\log\upgrep_prod_77\summary.xml
```

The Upgrade Tuner Process Information page appears.

Alternative: Start Upgrade Tuner without the */c* option. Click OK at the error pop-up. When Upgrade Tuner displays, click Browse and navigate to the summary.xml file.

Saving Your Changes and Exiting Upgrade Tuner

Use this procedure to save your changes and exit Upgrade Tuner.

To save your changes and exit Upgrade Tuner

- 1 Evaluate the changes you have made and revise them as needed.
Upgrade Tuner does not revise the upgrade files until you exit.
- 2 Click Save+Exit.
- 3 Click Yes in the pop-up window that asks you to confirm if you want to save and exit.
Upgrade Tuner applies your changes to the upgrade files and exits.

Discarding Your Session Changes and Exiting Upgrade Tuner

Use this procedure to discard the changes you have made in the current session and exit Upgrade Tuner. No changes are made to upgrade files.

Parallelize Table or Index Creation Pages

Changes are discarded and do not display the next time you start Upgrade Tuner, if:

- You moved tables or indexes between threads, these changes are discarded.
- You created new threads, the threads are discarded.

Deactivate 0-Row SQLs

Changes are discarded and do not display the next time you start Upgrade Tuner, if:

- You deactivated a command, this change is discarded. The next time you start Upgrade Tuner, no check mark displays in the Inactive column. The command remains active in the SQL file.
- You activated a command, this change is discarded. The next time you start Upgrade Tuner, a check mark displays in the Inactive column, and the command remains inactive in the SQL file.

To discard your changes and exit Upgrade Tuner

- 1 Click Cancel.
- 2 Click Yes in the pop-up window that asks if you to confirm you want to discard your changes and exit.

Related Topics

["About Tuning Siebel Production Upgrade Files" on page 112](#)

Managing Parallel Threads Using Siebel Upgrade Tuner

Upgrades from: All Supported Siebel releases.

Environments: Production test environment only. Does not apply to production environment.

Platforms: Windows and UNIX only.

Databases: Oracle only.

Upgrade Tuner allows you to create, edit, and delete parallel thread for table and index creation. This improves upgrade performance by reducing the amount of time required to complete table and index creation.

You create, edit, and delete threads in the Parallelize Table Creation page and the Parallelize Index Creation page. These two pages have the same layout.

Prerequisites

- In the production test environment, you must have completed the production upprep step.
- You must have run the Logparse utility on the production upprep log files. See [“Summarizing Siebel Log Files Using the Logparse Utility” on page 239](#).
- For UNIX platforms, you must have transferred files to a Windows host on which a Siebel Server is installed. See [“Transferring UNIX Files for Use by Siebel Upgrade Tuner” on page 346](#).
- Start Upgrade Tuner. See [“Starting and Stopping Siebel Upgrade Tuner” on page 338](#).

Displaying Threads

You can view and sort the contents of threads or view items sorted across threads:

- **Default sort.** The default sort is all the items in the Serial Thread sorted from highest to lowest cost. The default sort displays when you start Upgrade Tuner, add a thread, or remove a thread. To reverse the sort order, click on the Serial Thread column head.
- **Contents of a thread.** Click on the column head for that thread. To reverse the sort-order, click on the column head again.
- **All items in all threads sorted by cost.** Click on the Cost per Table column head. To reverse the sort-order, click on the column head again.

Creating Parallel Threads

Use this procedure to create a parallel thread for table creation or index creation. Upgrade Tuner automatically names threads Parallel Thread 1, Parallel Thread 2, and so on. You cannot edit thread names.

You start creating parallel threads by creating Parallel Thread 1 and Parallel Thread 2 together. You must assign at least one table or index to each of these threads.

All threads you create must contain at least one table or index.

To create Parallel Thread 1 and Parallel Thread 2

- 1 In the Serial Thread, select a table or index, and move it to the right using the arrow key on the keyboard.

Upgrade Tuner creates Parallel Thread 1 and Parallel Thread 2. It then assigns the table or index to Parallel Thread 1.

- 2 Move at least one table or index to Parallel Thread 2, to populate both threads.

To create additional parallel threads

- Select a table or index and move it to the right using the arrow key on the keyboard.

When you move the table or index to the highest-numbered thread and click the arrow key again, Upgrade Tuner creates a new thread and places the table or index in the new thread.

TIP: Another way to create a new thread is to right-click in a row. In the drop-down menu, select the last thread listed.

Moving Items Between Threads

Use the left and right arrow keys to move tables or indexes between threads, including the Serial Thread.

NOTE: You cannot save and exit if any thread is empty.

Deleting a Thread

Use this procedure to delete an existing thread. You cannot delete the Serial Thread.

You must delete Parallel Thread 1 and Parallel Thread 2 together. You must delete all other threads before deleting Parallel Thread 1 and Parallel Thread 2.

To delete a thread other than Parallel Thread 1 or Parallel Thread 2

- 1 Click the column head for the desired thread.
This action sorts the list so that all items in that thread appear at the top.
- 2 Right-Click in the column head, and choose "Move all items to the serial thread" from the pop-up menu.
- 3 Right-Click in the column head, and choose "Remove thread" from the pop-up menu.
Upgrade Tuner deletes the thread and renames all higher-numbered threads.

To delete Parallel Thread 1 and Parallel Thread 2

- 1 Right-Click the column head for Parallel Thread 2, and choose "Move all items to the serial thread" from the pop-up menu.
- 2 Right-Click the column head for Parallel Thread 1, and choose "Move all items to the serial thread" from the pop-up menu.

- 3 Right-Click in the column head for Parallel Thread 2, and choose “Remove thread” from the pop-up menu.

Upgrade Tuner deletes Parallel Thread 1 and Parallel Thread 2.

Evaluating Upgrade Performance Improvement

To evaluate production upprep performance improvement, use the two fields at the top of the page:

- **Total cost of sequential table (or index) creation.** Displays the time to create tables or indexes when no parallel threads are used.
- **Total cost of parallelized table (or index) creation.** Displays the time to complete the upgrade using the parallel threads you have created. The time is computed by adding the Serial Thread time and the longest-running parallel thread time.

The difference between the sequential creation time and the parallelized creation time is an estimate of the reduction in upgrade time from using parallel threads.

You can reduce upgrade time further by performing the following actions:

- Move additional items from the Serial Thread to a parallel thread
- Move items from the longest-running parallel thread to other threads or a new thread

The goal is to reduce both the Serial Thread time and longest-running parallel thread time to a minimum. Because each new parallel thread requires additional memory and CPU cycles, you may need to experiment with the number of parallel threads to optimize upgrade performance.

Related Topics

[“About Tuning Siebel Production Upgrade Files” on page 112](#)

Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner

Upgrades from: All Supported Siebel releases.

Environments: Production test environment only. Does not apply to production environment.

Platforms: Windows and UNIX only.

Databases: All databases.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The upgrade scripts support all the tables in the Siebel data model. This support means the tables may contain SQL commands that run against tables that are not included in your Siebel Database, that are empty, or do not contain data that applies to a specific SQL command. By inactivating such SQL commands, you can reduce the time required to perform the production upprep.

The Deactivate 0-Row SQLs page displays a list of SQL files that contain commands that returned zero rows. This means the command does not affect any data and does not change the database schema. The screen displays only upgrade commands executed natively by the RDBMS. The screen does not display SQL commands executed using odbcsql.

The SQL files are located in `DBSRVR_ROOT\DBPLATFORM\upgrade\VERSION`, for example, `db2udb\upgrade\V7_7\`.

When you select a file, the command that returned zero rows displays in the lower half of the screen. You can then either deactivate or activate the command. You cannot edit the displayed command.

When you deactivate a command and save your changes, Upgrade Tuner opens the SQL file containing the command and inserts `(Execute=No)` in the command. When you activate a command, Upgrade Tuner removes `(Execute=No)` from the command.

Prerequisites

- In the production test environment, you must have completed the production upprep step.
- You must have run the logparse utility on the production upprep log files. See [“Summarizing Siebel Log Files Using the Logparse Utility” on page 239](#).
- For UNIX platforms, you must have transferred files to a Windows host on which a Siebel Server is installed. See [“Transferring UNIX Files for Use by Siebel Upgrade Tuner” on page 346](#).
- Analyze your customizations and the nature of application data. Verify that you understand the role of any new tables you have added.
- Start Upgrade Tuner. See [“Starting and Stopping Siebel Upgrade Tuner” on page 338](#).

Displaying Zero-Row SQLs

You can view and sort zero-row SQLs in several ways:

- **Default sort.** The default sort order is the order in which the zero-row commands appear in the driver files. Any inactivated SQL commands, including those inactivated in previous sessions appear at the end of the list. The default sort order displays when you start Upgrade Tuner.
- **Display items sorted by cost.** To sort commands from longest-running time to shortest, click the Net Cost column head. To reverse the sort order, click Net Cost again. Commands inactivated prior to this session appear at the end of the list.
- **Display commands activated or deactivated in the current session.** Click the Inactive column head. Items display at the beginning of the list. The word Changed displays in the Inactive column for these items. Items that have been deactivated display check marks. Items that have been activated do not.
- **Display commands inactivated in previous sessions.** Click the Net Cost column head and scroll to the end of the list. Inactivated commands do not have a check mark in the Inactive column and do not display the word Changed.
- **Display commands activated in previous sessions.** The display of SQL commands does not provide a way to identify commands activated in a previous Upgrade Tuner session. When you activate a command, write down its SQL file name and SQL tag number so you can locate the command in future sessions.

- **Display all the zero-row SQL commands in a file.** Click on the SQL File column head. This action sorts the file names alphabetically. To reverse the sort order, click on the column head again.

Deactivating Zero-Row SQL Commands

Use this procedure to deactivate SQL commands that do not affect any data.

To deactivate zero-row SQL commands

- 1 Click the Deactivate 0-Row SQLs tab in Upgrade Tuner.

The Deactivate 0-Row SQLs screen appears.

- 2 Click the Net Cost column head.

This sorts the entries so that the longest running SQL commands appear first. If they do not, click the column head again. Commands deactivated in previous Upgrade Tuner sessions display at the end of the list.

- 3 Click in a row to display a command that returned zero rows.

- 4 Carefully evaluate whether you need this command for your upgrade.

- 5 Write down the net cost of the command.

You can use a spreadsheet to keep track of net cost changes, if you prefer.

- 6 To deactivate the command, click in the check box in the Inactive column.

The following occurs:

- A check mark displays indicating the command is inactive.
- The word Changed appears next to the check mark to indicate the change was made in this session.
- The time displayed in the Net Cost column changes to N/A.
- When you save and exit, Upgrade Tuner inactivates the command in the SQL file.
- The next time you start Upgrade Tuner, a check mark displays in the Inactive column for the command, but the word Changed does not.

Activating Zero-Row SQL Commands

Use this procedure to activate SQL statements that do not affect any data.

To activate zero-row SQL commands

- 1 Click the Net Cost column head, and then scroll to the end of the list.

This sorts commands by running time. Inactive commands have a running time of N/A and always appear at the end of the list.

- 2 Click in a row to display a command that returned zero rows.

3 Carefully evaluate whether you need this command for your upgrade.

4 To activate the command, click in the check box in the Inactive column.

The following occurs:

- The check mark disappears from the check box, indicating the command is active.
- The word Changed appears next to the check box to indicate the change was made in this session.
- The time displayed in the Net Cost column remains N/A.
- When you save and exit, Upgrade Tuner activates the command in the SQL file.
- The next time you start Upgrade Tuner, N/A is replaced by the running time for the command, and the word Changed does not appear.

5 Write down the SQL file name and SQL tag number for the command.

6 The next time you run Upgrade Tuner, locate the command and write down its net cost.

You can use a spreadsheet to keep track of net cost changes, if you prefer.

Evaluating Upgrade Performance Improvement

To evaluate production upprep performance improvement, add together the net cost of all the zero-row SQLs you deactivated. Then subtract the net cost of the zero-row SQLs you activated.

The final sum is an estimate of how much you have reduced the time required for the next production upprep.

Related Topics

[“About Tuning Siebel Production Upgrade Files” on page 112](#)

Transferring UNIX Files for Use by Siebel Upgrade Tuner

Upgrades from: All Supported Siebel releases.

Environments: Production test environment only. Does not apply to production environment.

Platforms: UNIX only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Upgrade Tuner is part of the Siebel Server and runs only under Windows. To tune production upgrade files on a UNIX machine you must do the following:

- Transfer upgrade files needed by Upgrade Tuner from the UNIX host to a Windows host on which a Siebel Server is installed.
- Run Upgrade Tuner on the Windows host using the UNIX files as input.
- Transfer the modified upgrade files from the Windows host back to the UNIX host.

Scripts generated by the Logparse utility during the production upgrep on the UNIX host simplify the file transfer process:

- **upgtuner_ftp_get.txt.** This script moves upgrade files from a UNIX host to a target directory on the Windows host.
- **upgtuner_ftp_put.txt.** This script moves the upgrade files from the Windows host to a target directory on the UNIX host.

Prerequisites for the UNIX Host

- In the production test environment, you must have completed the production upgrep step.
- You must have run the logparse utility on the production upgrep log files. See [“Summarizing Siebel Log Files Using the Logparse Utility” on page 239.](#)

Prerequisites for the Windows Host

- You must have installed a Siebel Server. You do not have to install the Siebel Database Server.
- The Windows host can be a Siebel Server on which have performed upgrades. Tuning upgrade files from a UNIX host does not interfere with upgrade files already on the Windows-based Siebel Server.
- To run the upgtuner_ftp_put.txt script, you must be able to FTP from the Windows host to the UNIX host.

The following procedures use FTP to transfer files. If FTP is not available, you can use other methods for transferring files.

Transferring Files from the UNIX Host to the Windows Host

To run Upgrade Tuner on UNIX upgrade files, you must first transfer the files to a Windows host.

To transfer files from the UNIX host to the Windows host

- 1 **Windows host.** Create a target directory for the UNIX upgrade files, and share the directory.
- 2 **UNIX host.** FTP the following scripts to the Windows machine target directory:
 - upgtuner_ftp_get.txt
 - upgtuner_ftp_put.txt

The files are located in \$SIEBEL_ROOT/bin.

- 3 **Windows host.** In both scripts, replace placeholder parameters with actual values, as described in the following table.

Placeholder	Value
&HostIP	This value is the IP address of UNIX machine.
&Username	This value is the username used to open an FTP session with the UNIX machine (for example sadmin).
&WindowsTempDir	This value is the full path of the target directory on the Windows machine. The target directory does not have to be within the Siebel Server installation. Avoid using a target directory that already contains upgrade files.

- 4 **Windows host.** Use FTP and upgtuner_ftp_get.txt to move the files shown in the following table from the UNIX host to the target directory on the Windows host.

File	Location on the UNIX Host
summary.xml	\$SIEBEL_ROOT/Log/upgprep_prod_ <i>VERSION</i> / For example, \$SIEBEL_ROOT/Log/upgprep_prod_77/summary.xml
master_upgprep_prod_ <i>VERSION</i> .ucf	\$SIEBEL_ROOT/bin/ For example, \$SIEBEL_ROOT/bin/master_upgprep_prod_77.ucf
schema*.ddl	<i>DBSRVR_ROOT/DBPLATFORM</i> / For example, <i>DBSRVR_ROOT/Oracle</i> /schema.ddl , schema_t1.ddl , schema_t2.ddl
driver_upgprep_prod_ <i>VERSION</i> .ucf	<i>DBSRVR_ROOT/DBPLATFORM</i> /upgrade/ <i>VERSION</i> / For example, <i>DBSRVR_ROOT/Oracle</i> /upgrade/V7_7/ driver_upgprep_prod_77.ucf
*.sql	<i>DBSRVR_ROOT/DBPLATFORM</i> /upgrade/ <i>VERSION</i> / For example, <i>DBSRVR_ROOT/Oracle</i> /upgrade/V7_7/pret.sql , preschm.sql

- 5 **Windows host.** Navigate to the target directory containing the UNIX upgrade files, and open the summary.xml file in a text editor.
- 6 **Windows host.** Near the beginning of the file, locate the element <SIEBEL_ROOT>, and edit the value to be the absolute path to the target directory containing the UNIX files that you copied to the Windows host.
- 7 **Windows host.** Save the file, and exit.

- 8 Windows host.** Start Upgrade Tuner, and tune the UNIX upgrade files.

Specify the target directory containing the UNIX upgrade files. The `summary.xml` file contains a flag that tells Upgrade Tuner to look for all the upgrade files in the target directory. You do not have to move the files.

Transferring Files from the Windows Host to the UNIX Host

After you have tuned the UNIX upgrade files, transfer them back to the UNIX host.

To transfer files from the Windows host to the UNIX host

- 1 UNIX host.** Create a target directory for the UNIX upgrade files that will be transferred from the Windows host.

Alternative: Use the FTP upload directory for the UNIX host.

- 2 Windows host.** FTP the UNIX upgrade files from the target directory to the UNIX host.

- 3 UNIX host.** Move the upgrade files to their proper locations.

The path for `<SIEBEL_ROOT>` in the `summary.xml` file is used for the Windows host and thus is incorrect for the UNIX host. The next time you run `logparse`, it will overwrite `summary.xml` and include the path for `SIEBEL_ROOT` on the UNIX host.

Related Topics

[“About Tuning Siebel Production Upgrade Files” on page 112](#)

Rolling Back Siebel Upgrade Tuner Changes

Upgrades from: All Supported Siebel releases.

Environments: Production test environment only. Does not apply to production environment.

Platforms: Windows and UNIX only.

Databases: All databases.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Use these procedures to discard the changes you saved from the most recent Upgrade Tuner session. You do this by rolling back the upgrade files to a previous Upgrade Tuner session.

This roll-back process is particularly useful for UNIX users. You can roll back upgrade files to a previous version on the UNIX host. You do not have to transfer the files to a Windows host and rerun Upgrade Tuner.

Upgrade File Versions

Before Upgrade Tuner saves changes to the upgrade files, it does the following:

- If this your first Upgrade Tuner session, Upgrade Tuner saves the current driver and SQL files to `.orig`, for example `dri ver_upgprep_prod_77.ucf.orig`.
- If this is the second session or all later sessions, Upgrade Tuner saves the current driver and SQL files to `.old`, for example `dri ver_upgprep_prod_77.ucf.old`.

To roll back, you replace the upgrade file with the `.old` or `.orig` version.

Roll-Back Guidelines

Use the following guidelines to roll back upgrade files:

- To roll back to the previous Upgrade Tuner session, replace the driver or SQL file with the `.old` version.
- To roll back to the original version of the file, before any Upgrade Tuner modifications, replace the driver or SQL file with the `.orig` version.
- You can roll back the driver and SQL files separately. For example, you can roll-back to the original driver file while retaining the most recent changes to SQL files.
- You do not have to roll back all SQL files at once. For example, you can roll back some SQL files while retaining the most current version of others.
- You do not have to roll back all the commands in an SQL file. You can edit the file and activate or deactivate as many commands as desired.
- Manually editing the `dri ver_upgprep_prod_VERSION.ucf` file is not recommended.
- If you roll back to a session with fewer threads, you do not have to delete any `schema_t#.ddl` or `schema_i#.ddl` thread-files. The deletion is unnecessary because Upgrade Tuner removes from the driver file the steps that execute the deleted threads.

The following procedures use `dri ver_upgprep_prod_77.ucf` and `pret.sql` as examples.

Rolling Back to the Previous Session

Use this procedure to discard your most-recent session and roll back to the previous session.

To roll back to the previous session

- 1 Save a copy of `dri ver_upgprep_prod_77` and `pret.sql` to new names.
- 2 Copy `dri ver_upgprep_prod_77.old` to `dri ver_upgprep_prod_77`.
- 3 Copy `pret.sql.old` to `pret.sql`.
- 4 Restart Upgrade Tuner.

Rolling Back to the Original Upgrade Files

Use this procedure to discard all Upgrade Tuner changes and roll back to the original upgrade files.

To roll back to the original upgrade files

- 1 Save a copy of `dri ver_upgrep_prod_77` and `pret. sql .`
- 2 Copy `dri ver_upgrep_prod_77.ori g` to `dri ver_upgrep_prod_77.`
- 3 Copy `pret. sql .org` to `pret. sql .`

Activating or Deactivating SQL Commands Manually

Use this procedure to activate or deactivate individual zero-row SQL commands by editing the SQL file.

To activate or deactivate a command by editing an SQL file

- 1 Save a copy of the SQL file to a new name.
- 2 Open the `.sql` file (not the copy) and locate the desired SQL command.
Commands begin `Run_SQL_#`, for example `Run_SQL_100`.
- 3 Edit the command as follows:
 - To activate the command, delete the element `(Execute=N)`
 - To deactivate a command, add the element `(Execute=N)`
Insert the element on a line by itself after `"Run_SQL_# ="`.
- 4 Save the file.

Related Topics

["About Tuning Siebel Production Upgrade Files" on page 112](#)

A

Siebel Marketing Upgrade Reference

This appendix provides reference information for Application Developers of Siebel Marketing. It includes the following topics:

- ["Obsolete Siebel Marketing Data" on page 353](#)
- ["Obsolete Business Objects in Siebel Marketing" on page 359](#)
- ["Obsolete or Replaced Views in Siebel Marketing" on page 360](#)

Related Topics

[Chapter 4, "How to Perform a Siebel Database Upgrade"](#)

Obsolete Siebel Marketing Data

Upgrades from: Siebel 7.0.x and 7.5.x.

Platforms: Windows, UNIX, IBM z/OS.

[Table 45 on page 353](#) lists marketing data that is obsolete as of Siebel 7.7. In the table, the Comments field explains how the data is handled.

Table 45. Obsolete Siebel Marketing Data as of Siebel 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
Programs (program occurrences)	Obsolete	None	None	Activities migrated to parent program.
Stages (stage occurrences)	Obsolete	None	None	N/A

Table 45. Obsolete Siebel Marketing Data as of Siebel 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
Campaigns (campaign occurrences)	Obsolete	None	DBM Campaign Campaign Occurrences Campaign Occurrences Delete Campaign Results Campaign Segment Allocation Campaign Template Related Event Templates Campaign Wave List Distribution DBM Campaign Cost DBM Campaign Cost (Fixed) DBM Campaign Cost (Inbound) DBM Campaign Cost (Outbound) DBM Campaign Occurrence Cost DBM Campaign Occurrence Lists Campaign Occurrence Offer DBM Preview List	A campaign load record is created for each occurrence.

Table 45. Obsolete Siebel Marketing Data as of Siebel 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
eNewsletter Offers	Obsolete	None	Enewsletter Offer Enewsletter Offer Attachments Enewsletter Offer Attachments.Sequence Number (Sequence) Enewsletter Offer Comm Profile Parameter Enewsletter Offer Related URLs Enewsletter Offer Template Related Web Offers f/ Enewsletter Offer Related Web Surveys f/ Enewsletter Offer	Newsletter functionality is supported by conditional content formatting in email offers.
Segments	Usage history in programs is preserved. Segment criteria are obsolete.	S_CALL_LST_CRIT S_CALL_LST_DTL S_CALL_LST_ORY	Segment Detail-DD Segment Expression Campaign Segment Allocation Segment Campaign Allocation	Segments should be reconstructed in the Segment Designer.
Filters	Obsolete	S_DD_FILTER S_DD_FILTER_DTL	Filters Filters Detail Filters Expression Save Attribute Level Look Up	Filters should be applied using criteria in the Segment Designer or Segment Tree Designer or metadata constraints in the Oracle Business Intelligence repository.

Table 45. Obsolete Siebel Marketing Data as of Siebel 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
List Formats	Obsolete	S_DD_LST_FMT S_DD_LST_FMTDTL S_DD_PROGLSTFMT	List Columns List Columns.Sequence (Sequence) List Formats Program List Formats Dependent List Format	Create List Formats using the List Format Designer.
Source Code Formats	Obsolete	None	None	Old source code formats are obsolete. After upgrade, source code formats need to be re-created with new elements.
Tables Joins	Obsolete	S_DD_DATA_OBJ S_DD_UNION_MBR S_DD_JOIN S_DD_JOIN_SPEC	DD Field DD Join DD Table Fields One Level Join Fields Join Fields. Sequence (Sequence) Union Tables	All data source metadata is managed in the Oracle Business Intelligence repository.

Table 45. Obsolete Siebel Marketing Data as of Siebel 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
List Measures Bound Measures Custom Measures	Obsolete	S_DD_MEAS S_DD_MEASRSTRCT S_DD_SUBMEAS	Aggregation Function Available Measures Base Measure Bound Measures Custom Measures Dependent List Format Functions (Logical) Functions (Mathematical) List Measures Measures Parent Measures RowNum Measure SubMeasures Measure Aggregation (Tree) Members.Sequence (Sequence) Measure Restrict Measure Restrict (Tree)	Replaced by formulas in the Oracle Business Intelligence repository.
Buckets (Measure Base Attributes)	Obsolete	S_DD_MEASATRP S_DD_MEAS_ATTR	Measure Attribute Members Measure Attribute Members (Tree) Measure Attribute Measure Attributes Measure Attributes (Tree)	Replaced by formulas in the Oracle Business Intelligence repository.

Table 45. Obsolete Siebel Marketing Data as of Siebel 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
Attributes and Attribute Families	Obsolete	S_DD_ATTRFAMVMB	Attribute Families	Segment criteria are now created using presentation columns ("fields") in the Segment Designer.
		S_DD_ATTRFAM	Attribute Family Resync VBC	
		S_DD_ATTRFAMLVL	Attribute Family Value	
		S_DD_ATTRFAMVAL	Attribute Filters Detail	
		S_DD_FIELD	Attribute Levels	
		S_DD_HIERATR	Attribute Levels.Sequence (Sequence)	
		S_DD_HIERATRKEY	Hierarchical Attribute Value	
		S_DD_HIERATRLVL	Hierarchical Attributes	
		S_DD_HIERATRVAL	Hierarchy (Fields)	
			Hierarchy (Tree)	
			Hierarchy Levels (Tree)	
			Hierarchy Levels Search (Tree)	
			Hierarchy Search (Fields)	
			Hierarchy Search (Tree)	
	Hierarchy Values (Fields)			
	Hierarchy Values (Tree)			
	Hierarchy.Sequence (Sequence)			

Table 45. Obsolete Siebel Marketing Data as of Siebel 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
Customer Hierarchies	Obsolete	S_DD_STRFAM S_DD_SRTFAM_LVL S_DD_SRTFAM_MAP	Hierarchy Hierarchy Levels Hierarchy Levels (Fields) User Defined Levels User Defined Levels.Sequence (Sequence)	No longer necessary. Customer-level targeting is supported by Target Levels in the Oracle Business Intelligence repository.
Campaign Load Mappings (Contact Key Formats)	Obsolete	N/A	N/A	N/A
Server tables (internal use)	Obsolete	S_DD_CUBE S_DD_DIM S_DD_DIM_DTL S_DD_EXTRACT S_DD_EXTRACT_CL S_DD_PROGEXPORT S_DD_SNPSHT_FMT S_DD_PRRPERFMEAS	Snapshot Elements	These tables were used by obsolete Siebel Server Components (Data Dictionary Manager and Marketing Server).

Obsolete Business Objects in Siebel Marketing

Upgrades from: Siebel 7.0.x and 7.5.x.

Platforms: Windows, UNIX, IBM z/OS.

[Table 46 on page 359](#) lists the Siebel Repository business objects that are obsolete as of Siebel 7.7.

Table 46. Obsolete Siebel Marketing Business Objects as of Siebel 7.7

Business Object Name	Tools Project
Customer Hierarchies	Server (DD)
DBM Campaign	eMarketing - Campaign Views

Table 46. Obsolete Siebel Marketing Business Objects as of Siebel 7.7

Business Object Name	Tools Project
DBM Campaign Btn	eMarketing - Campaign Views
DBM Financial Modeler	Mktg Financial Rollup
Data Dictionary	Data Dictionary (DBM)
External Contacts Mapping?	Server (DD)
Filters	Filters (DBM)
Hierarchical Attributes	Attributes (DBM)
Measure Attributes	Attributes (DBM)
Measures	Measures (DBM)
Segment	Segment (DBM)

Obsolete or Replaced Views in Siebel Marketing

Upgrades from: Siebel 7.0.x and 7.5.x.

Platforms: Windows, UNIX, IBM z/OS.

Table 47 on page 360 lists the Siebel 7.x Siebel Repository views that are obsolete or have been replaced as of Siebel 7.7.

Table 47. Siebel 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Siebel 7.x Screen	Siebel 7.x View Caption	Outcome	Replacement View
All Campaigns (DBM) View	Campaign Management Screen	All Campaign Plans across Organizations	Replaced	Campaign Administration List
All Decisions List View	Decision Administration	All Decisions	Obsolete	N/A
All Measures View	Marketing Administration	All Measures	Obsolete	N/A
All Segments across My Organizations	Segment Screen	All Segments	Replaced	All Marketing Segments View
All Tables View	Marketing Administration	Tables	Obsolete	N/A
Attribute Families View	Marketing Administration	Attribute Families	Obsolete	N/A

Table 47. Siebel 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Siebel 7.x Screen	Siebel 7.x View Caption	Outcome	Replacement View
Bound Measures View	Marketing Administration	Bound Measures	Obsolete	N/A
Campaign (DBM) Activities View	Campaign Occurrence Screen	Activities	Replaced	Campaign Activity View Admin
Campaign (DBM) Offers View	Campaign Management Screen	Offers	Replaced	Campaign Literature View Admin
Campaign (DBM) Team View	Campaign Management Screen	Team	Inactivated	N/A
Campaign (DBM) Template Activities View	Campaign Management Screen	Activity Plans	Replaced	Campaign Activity Plans View
Campaign Admin Lists (SCW)	Campaign Screen (SCW)	Lists	Replaced	Campaign Admin Lists
Campaign Admin Objectives (SCW)	Campaign Screen (SCW)	Quotas	Replaced	Campaign Admin Objectives
Campaign Administration Detail (SCW)	Campaign Screen (SCW)	Contacts/Prospects	Replaced	Campaign Contacts Admin View - Org
Campaign Administration Detail - Owner Audit Trail View (SCW)	Campaign Screen (SCW)	Owner Audit Trail	Replaced	Campaign Administration Detail - Owner Audit Trail View
Campaign Administration Explorer View (SCW)	Campaign Screen (SCW)	Campaign Explorer	Replaced	Campaign Administration Explorer View
Campaign Administration List (SCW)	Campaign Screen (SCW)	Campaigns	Replaced	My Campaigns View
Campaign Administration List-More Info (SCW)	Campaign Screen (SCW)	More Info	Replaced	Campaign Administration List-More Info
Campaign Forecast (DBM) View-Cost Allocation Inputs	Campaign Management Screen	Cost Allocation Inputs	Replaced	Campaign Expenses View
Campaign Generated Lists View	Campaign Occurrence Screen	Exported Lists	Replaced	Campaign List Distribution View

Table 47. Siebel 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Siebel 7.x Screen	Siebel 7.x View Caption	Outcome	Replacement View
Campaign Groups View - Admin (SCW)	Campaign Screen (SCW)	Groups	Inactivated	N/A
Campaign List	Campaign Screen (SCW)	Campaign List	Replaced	My Campaigns View
Campaign Literature View Admin (SCW)	Campaign Screen (SCW)	Offers	Replaced	Campaign Literature View Admin
Campaign Occurrence Contacts/ Prospects	Campaign Occurrence Screen	Campaigns > All Contacts/ Prospects across Organizations	Replaced	Campaign Contacts Admin View - All
Campaign Occurrence Contacts/ Prospects - Organization	Campaign Occurrence Screen	Campaigns > All Contacts/ Prospects	Replaced	Campaign Contacts Admin View - Org
Campaign Occurrence Contacts/ Prospects - Position	Campaign Occurrence Screen	Campaigns > My Contacts/ Prospects	Obsolete	N/A
Campaign Occurrence Contacts/ Prospects Owner Audit View	Campaign Occurrence Screen	Campaigns > Owner Audit Trail	Replaced	Campaign Administration Detail - Owner Audit Trail View
Campaign Occurrence Email Status	Campaign Occurrence Screen	Campaigns > Email Status	Replaced	Campaign System Task View
Campaign Occurrence Offers View	Campaign Occurrence Screen	Offers	Replaced	Campaign Execution History View
Campaign Occurrence Timeline Gantt Chart View	Campaign Management Screen	Timeline	Replaced	Marketing Calendar - Campaigns Ax Gantt Chart View - My
Campaign Occurrences Responses View	Campaign Occurrence Screen	Campaigns > Responses	Replaced	Campaign Responses View
Campaign Occurrences View	Campaign Management Screen	Status	Replaced	Campaign Execution History View

Table 47. Siebel 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Siebel 7.x Screen	Siebel 7.x View Caption	Outcome	Replacement View
Campaign Plan Activities View	Campaign Management Screen	Activities	Replaced	Campaign Activity View Admin
Campaign Responses View (SCW)	Campaign Screen (SCW)	Responses	Replaced	Campaign Responses View
Campaign Results (DBM) Input Summary View	Campaign Management Screen	Results > Results Summary	Obsolete	N/A
Campaign Results (DBM) View- Cost Inputs	Campaign Management Screen	Results > Cost Results	Replaced	Campaign Expenses View
Campaign Results (DBM) View-Cost Allocation Results	Campaign Management Screen	Results > Cost Allocation Results	Replaced	Campaign Expenses View
Campaign Results (DBM) View-List Results	Campaign Management Screen	Results > List Results	Replaced	Campaign Segment/ List Assumptions View
Campaign Results (DBM) View-Revenue Results	Campaign Management Screen	Results > Revenue Results	Obsolete	N/A
Campaign Results (DBM) View-Segment Results	Campaign Management Screen	Results > Segment Inputs	Replaced	Campaign Segment/ List Assumptions View
Campaign Skill View (SCW)	Campaign Screen (SCW)	Assignment Skills	Replaced	Campaign Skill View
Campaign Team View - Admin (SCW)	Campaign Screen (SCW)	Team	Inactivated	N/A
Campaign Waves View	Campaign Management Screen	Waves	Replaced	Program Schedule Detail View
Columns View	Marketing Administration	Fields	Obsolete	N/A
Custom Measure Aggregation View	Marketing Administration	Custom Measure > Aggregation	Obsolete	N/A
Custom Measure Details View	Marketing Administration	Custom Measure > More Info	Obsolete	N/A

Table 47. Siebel 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Siebel 7.x Screen	Siebel 7.x View Caption	Outcome	Replacement View
Custom Measure Restriction View	Marketing Administration	Custom Measure > Restriction	Obsolete	N/A
Customer Hierarchies View	Marketing Administration	Customer Hierarchies	Obsolete	N/A
DBM Campaign Timeline Gantt Chart View	Campaign Management Screen	Timeline	Replaced	Marketing Calendar - Campaigns Ax Gantt Chart View - My
DD All Segments View	Segment Screen	All Segments across Organizations	Replaced	Marketing Segments Across All Organizations View
DD Segment Detail View	Segment Screen	Edit Segment	Replaced	SSO Mktg Segments Entry View
Database Synchronization	Marketing Administration	Data Retrieval	Obsolete	N/A
Decision Wizard Player View	Decisions Administration Screen	N/A	Obsolete	N/A
Decisions Detail Input Parameters View	Decisions Administration Screen	Identifiers	Obsolete	N/A
Decisions Detail Output Parameters View	Decisions Administration Screen	Outputs	Obsolete	N/A
Decisions Detail Session Parameters View	Decisions Administration Screen	Real Time Inputs	Obsolete	N/A
Decisions Detail View	Decisions Administration Screen	More Info	Obsolete	N/A
Enewsletter Attachments View	Offer Screen	eNewsletter > Attachments	Obsolete	N/A
Enewsletter Email Profile View	Offer Screen	eNewsletter > Profile	Obsolete	N/A
Enewsletter Offer Detail View	Offer Screen	eNewsletter > More Info	Obsolete	N/A
Enewsletter Offer Related URLs View	Offer Screen	eNewsletter > Related URLs	Obsolete	N/A

Table 47. Siebel 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Siebel 7.x Screen	Siebel 7.x View Caption	Outcome	Replacement View
Enewsletter Offer Sections List View	Offer Screen	eNewsletter > Sections	Obsolete	N/A
Enewsletter Offer View	Offer Screen	eNewsletter Offers	Obsolete	N/A
Enewsletter Rules View	Offer Screen	eNewsletter Rules	Obsolete	N/A
External Contacts Mapping	Marketing Administration	Campaign Load Mapping	Obsolete	N/A
Filters Detail View	Marketing Administration	Filters	Obsolete	N/A
Filters View	Marketing Administration	Filters	Obsolete	N/A
Financial Modeler (DBM) View	Campaign Management Screen	Forecast > Input Summary	Obsolete	N/A
Financial Modeler (DBM) View-Fixed Costs	None	Financial Modeler	Obsolete	N/A
Financial Modeler (DBM) View-Inbound Costs	None	Financial Modeler	Obsolete	N/A
Financial Modeler (DBM) View-Input Costs	Campaign Management Screen	Forecast > Cost Inputs	Replaced	Campaign Expenses View
Financial Modeler (DBM) View-List Inputs	Campaign Management Screen	Forecast > List Inputs	Replaced	Campaign Segment/List Assumptions View
Financial Modeler (DBM) View-Outbound Costs	None	Financial Modeler	Obsolete	N/A
Financial Modeler (DBM) View-Revenue Inputs	None	Financial Modeler	Obsolete	N/A
Financial Modeler (DBM) View-Segment Inputs	Campaign Management Screen	Forecast > Segment Inputs	Replaced	Campaign Segment/List Assumptions View
Hierarchical Attribute List View	Marketing Administration	Hierarchical Attributes	Obsolete	N/A

Table 47. Siebel 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Siebel 7.x Screen	Siebel 7.x View Caption	Outcome	Replacement View
Joins View	Marketing Administration	Joins	Obsolete	N/A
List Formats View	Marketing Administration	Output List Layouts	Obsolete	N/A
List Measures View	Marketing Administration	List Measures	Obsolete	N/A
Marketing Plans Funds	Marketing Plans	Funds	Inactivated	N/A
Measure Based Attributes View	Marketing Administration	Buckets	Obsolete	N/A
My Campaigns (DBM) View	Campaign Management Screen	My Campaign Plans	Replaced	My Campaigns View
My Campaigns (DBM) View - More Info	Campaign Management Screen	My Campaign Plans	Replaced	Campaign Administration List-More Info
My Decisions List View	Decision Administration	My Decisions	Obsolete	N/A
My Segments View	Segments	My Segments	Replaced	My Marketing Segments View
My Team's Decisions List View	Decision Administration	My Team's Decisions	Obsolete	N/A
Program (DBM) Activities View	Program Occurrence Screen	Activities	Replaced	Program Plan Activities View
Program Container Timeline Gantt Chart View	Program Screen	Timeline	Replaced	Program Schedule Timeline View
Program Lists View	Program Stages Screen	Preview List	Replaced	Campaign List Distribution View
Program Occurrence Timeline Gantt Chart View	Program Screen	Timeline	Replaced	Marketing Calendar - Programs Ax Gantt Chart View - My
Program Occurrences View	Program Screen	Status	Obsolete	N/A
Program Responses View	Program Occurrence Screen	Responses	Obsolete	N/A

Table 47. Siebel 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Siebel 7.x Screen	Siebel 7.x View Caption	Outcome	Replacement View
Program Snapshots View	Program Stages Screen	Snapshot Elements	Obsolete	N/A
Program Stages View	Program Screen	Stages	Inactivated	N/A
Related Events View for eNewsletter Offers	Offer Screen	Related Events	Obsolete	N/A
Related Web Offers View f/ Enewsletter	Offer Screen	Related Web Offers	Obsolete	N/A
Related Web Surveys View f/ Enewsletter	Offer Screen	Related Web Surveys	Obsolete	N/A
Response Detail View (SME) - More Info	Response Screen	More Info	Replaced	Response Detail View (Detail)
Segment Criteria View	Segment Screen	More Info	Obsolete	N/A
Segment Program Stages View	Segment Screen	Program Plans	Obsolete	N/A
eChannel All Programs View (DBM)	eChannel Program Screen - OLD	All Program Plans across Organizations	Replaced	All Programs View (DBM)
eChannel All Programs across My Organizations	eChannel Program Screen	All Programs	Replaced	All Programs across My Organizations
eChannel Campaign Administration Detail	eChannel Campaign Management Screen - OLD	Contacts/ Prospects	Replaced	Campaign Contacts Admin View - Org
eChannel My Programs View (DBM)	eChannel Program Screen - OLD	My Program Plans	Replaced	My Programs View (DBM)
eChannel Program Container Timeline Gantt Chart View	eChannel Program Screen - OLD	Timeline	Replaced	Marketing Calendar - Programs Ax Gantt Chart View - My
eChannel Programs Detail View (DBM)	eChannel Program Screen - OLD	More Info	Replaced	Programs Detail View (DBM)
eChannel Segment Program Stages View	eChannel Segment Screen	Program Plans	Obsolete	N/A

B

Tables Modified or Seeded During a Siebel Upgrade

This appendix provides information related to schema changes following the upgrade process. It includes the following topic: [“Important Schema Changes at Siebel 7.8” on page 369](#).

Related Topics

[Chapter 4, “How to Perform a Siebel Database Upgrade”](#)

Important Schema Changes at Siebel 7.8

Upgrades from: Siebel 7.0.x, 7.5.x, and 7.7.x.

Platforms: Windows, UNIX, IBM z/OS.

In Siebel 7.8.x, the upgrade includes the following important schema changes.

Customer and Order Management Applications

The following schema changes affect customer and order management applications. This includes Product Administration, Order Management, Quotes, Pricer, and Siebel Configurator.

New Party Payment Profile Table

The new Party Payment Profile table replaces the old Contact Payment Profile table. The new schema supports account and contact payment profiles. The upgrade migrates the data from the old Contact Payment Profile table to the new table.

Quote Teams

Quotes now support multiple sales representatives, or other positions, associated to a single quote. The existing reference to the sales representative for the quote is migrated to the new intersection table and reused as a primary child column.

Multiple Price Types

Products now support multiple price types (Siebel Business Applications only). The new Price Type Code column for existing products is set to One-Time. The new One-Time Charges Sub-Total column in Quote, Order, and Agreement line items is calculated and stored. The new Recurring Charges Sub-Total column in Quote, Order, and Agreement line items is set to 0.

Revision to the Line Item Discount Amount Field

The upgrade process changes how line item discounts are stored. If the data source for a discount amount is Pricer, the upgrade moves the value stored in the Discount Amount Field to the Pricing Adjustment Field.

If the Keep Discount flag is checked, the upgrade process does not make this change. Instead, all items that have this flag checked are treated as manual discounts.

The upgrade process determines the correct Header Discount Amount when both of the following are true:

- Before the upgrade, the line item did not have a Manual Discount Amount, Manual Discount %, or Manual Price Override specified
- The line item has a Header Discount % specified

Attribute Pricing Matrices in Pricer

Attribute pricing matrices in Siebel Pricer are replaced by attribute adjustments. After the database upgrade is complete, users must run a business service method to convert attribute pricing data to attribute adjustments. Obsolete tables and corresponding new tables are shown in [Table 48](#).

Table 48. Attribute Adjustment Tables

Obsolete Table	New Table
S_PRI_MTRX	S_DYN_MTRX
S_PRI_MTRX_ATTR	S_DYN_MTRX_DIM
S_PRI_MTRX_ITEM	S_DYN_MTRX_RL
S_PRI_MTRX_VAL	S_DYN_MTRX_DOM

Net Price, Currency Code, Exchange Date

The calculated Net Price field in Quote, Order, and Agreement line items is replaced by a new column. The upgrade calculates the value based on the basic configuration in the prior release. Quote, Order, and Agreement line items now support currency code and exchange date. The relevant data is migrated from the associated header record.

Effective Dates on Price List Line Items and CP Adjustments

Price List line items and Price List CP adjustments now support effective dates in the user key. The effective start and end date are copied from the associated header record where applicable. If this data is not available, the start date is set to 01-01-1980 and the end date is left as NULL.

Volume Discounts

Volume discounts now support discount amounts as well as the existing method of discount percentage. The new price adjustment type code column is used to specify the type. The existing discount amount column stores the associated value. Both columns are updated to support the existing data.

Bundle Discounts

Bundle factors in pricing models are converted to bundle sequences and discounts in Siebel 7.8. Obsolete tables and corresponding new tables are shown in [Table 49](#).

References to pricing models in price lists and price list items are converted to refer to bundle sequences. All other pricing models must be manually reimplemented as PPS Procedures after the database upgrade.

Table 49. Bundle Discount Tables

Obsolete Tables	New Tables
S_PRIMDL	S_BUNDLE_SEQ
S_PRIMDL_FCTR	<ul style="list-style-type: none"> ■ S_BDL_SEQ_ITEM ■ S_BUNDLE_DISCNT
S_PRIFCTR_ITM	S_BDL_DISC_ITEM
S_PRIMDLFCTRVAL	NA
S_PRIMDL_OBJ	NA
S_PRIMDL_OBJCRT	NA
S_PRIFCTITM_ATR	NA

Siebel Configurator

Siebel 7.8 introduces a new infrastructure for Siebel Configurator. The primary new table for configurator is S_VOD. This table stores the header information for products, classes, and attributes.

Other important changes are as follows:

- S_VOD_VER replaces S_PROD_CFGVER and stores the version information for product, class and attribute objects.
- S_ISS_OBJ_DEF stores the product and class definitions.
- S_ISS_ATTR_DEF stores the global (abstract) attribute definitions.
- S_ISS_ATTR_VAL replaces the concept of S_XA_ATTR.VLDTN_LOV_TYPE_CD and stores the enumerated values for the global attribute.
- S_ISS_OBJ_ATTR stores the relationship between local attributes and both classes and products. This replaces the relationships previously stored in S_XA_ATTR.CLASS_ID and S_PROD_INT_XA.

Obsolete tables and corresponding new tables are shown in [Table 50](#).

Table 50. Siebel Configurator Tables

Obsolete Tables	New Tables
S_XA_ATTR	<ul style="list-style-type: none"> ■ S_VOD ■ S_ISS_ATTR_DEF ■ S_ISS_ATTR_VAL ■ S_ISS_OBJ_ATTR
S_XA_ATTR_LANG	S_ISS_OBJ_ATTR_LANG
S_XA_CLASS	S_VOD, S_ISS_OBJ_DEF
S_XA_CLASS_LANG	S_VOD_LANG
S_PROD_CFGVER	<ul style="list-style-type: none"> ■ S_VOD_VER ■ S_ISS_UI_OPTION
S_PROD_INT_XA	S_ISS_OBJ_ATTR
S_PROD_ITEM	S_ISS_SUB_OBJ
S_PROD_ITEM_LANG	S_ISS_SOBJ_LANG
S_CFG_VAR_DEF	S_ISS_OBJ_LITEM
S_CFG_PROP_DEF	S_ISS_OBJ_RSRC
S_CFG_SCRIPT_DEF	S_ISS_OBJ_SCRIPT
S_CFG_RULE_DEF	S_ISS_OBJ_CFGRL
S_CFGRLDEF_LANG	S_ISS_CFRL_LANG
S_CFG_RULENODE	S_ISS_CFRL_NODE
S_CFG_UIGROUP	S_ISS_UIOPT_GRP
S_CFGUIGRP_LANG	S_ISS_UIGR_LANG
S_CFG_UIGRP_IT	S_ISS_UIGR_ITEM
S_PROD_CFG_PROP	S_ISS_OBUI_PROP
S_PRDCFGPR_LANG	S_ISS_UIPR_LANG

Like the obsolete tables, all of the new tables have versions. The upgrade migrates data from the obsolete tables to the new ones and creates new records for objects that were previously not versioned (particular classes, attributes, and products).

During the upgrade, additional data migration occurs as follows:

- S_PROD_INT to S_VOD and S_ISS_OBJ_DEF
- S_PROD_INT_LANG to S_VOD_LANG

S_PROD_INT and S_PROD_INT_LANG continue to be used in Siebel 7.8. Foreign key references to products still point to S_PROD_INT. The S_PROD_INT.CFG_MODEL_ID column is reused to refer to the associated S_VOD record in the Siebel Configurator infrastructure.

All the S_*_XA tables, for example S_QUOTE_ITEM_XA, now use the ATTR_NAME column to refer to the attribute associated to the object, for example Quote Item.

ATTR_ID is no longer used. In previous releases, it was used as a reference to S_XA_ATTR, which is obsolete in 7.8.

Captive Finance

Before Siebel 7.8, a contact was associated with only one bankruptcy. In Siebel 7.8, the relationship between contact and bankruptcy has been changed to M:M. The model also allows capturing bankruptcies for companies. The upgrade migrates data in two steps:

- Migrates bankruptcy information from S_CONTACT_FNX to S_BANKRUPTCY and S_BK_PARTY.
- Migrates associated attorney information for bankruptcies from S_PARTY_REL to S_BK_PARTY. In S_BK_PARTY, attorney type is Bank Attorney, Trustee Attorney, Debtor Attorney or Other Attorney.

The tables S_CONTACT_FNX and S_PARTY_REL continue to be used for other purposes.

eTraining

The upgrade makes the following schema changes:

- Siebel 7.8 introduces an automatic wait-list feature. Since existing wait-list records were created manually, the upgrade updates the S_SRC_EVT.AUTO_WAITLIST_FLG to N for these records.
- Moves Max Waitlist Num from S_PROD_INT_CRSE to S_SRC_EVT. This moves wait-list support from the course level to the class level.
- Moves the Allow Waitlist Flag from S_PROD_INT_CRSE to S_SRC_EVT. This moves wait-list support from the course level to the class level.

For existing class records, Max Waitlist Num and Allow Waitlist Flag are set to the values that were present for the corresponding course.

Siebel Field Service

Quote, Order, and Agreement line items now support multiple covered assets associated to a single line item. The existing reference to the covered asset for the line item is migrated to an intersection table and reused as a primary child column.

The upgrade creates the following new intersection tables:

- S_AGREE_ITM_REL from S_AGREE_ITEM
- S_ORDER_ITM_REL from S_ORDER_ITEM
- S_QUOTE_ITM_REL from S_QUOTE_ITEM

The S_AGREE_ITEM, S_ORDER_ITEM and S_QUOTE_ITEM tables continue to be used.

Consumer Goods Advanced Planning

Oracle's Siebel Applications Siebel 7.8 introduces an Account Promotion Category in promotion planning. The new account promotion hierarchy is as follows:

- Account Plan
- Account Promotion
- Account Promotion Category
- Account Promotion Product
- Account Promotion Product Baseline/Shipment

To support this, the upgrade inserts account promotion records in S_SRC. The identifier for each record is SUB_TYPE = PLAN_ACCT_PROMOTION_CATEGORY.

The upgrade makes the following changes:

- For every Account Promotion record in S_SRC, the upgrade inserts a record for Account Promotion Category. The parent of the new record is the Account Promotion record.
- Populates S_SRC_CHNL, an extension table for S_SRC, for Account Promotion Category. S_SRC_CHNL.PAR_ROW_ID points to the new record created in S_SRC. S_SRC_CHNL stores certain attributes for Promotion, Promoted Products, and so on.
- Reparents Account Promotion Product records by pointing S_SRC.PAR_SRC_ID to Account Promotion Category.
- Revises Deals records stored in S_MDF_ALLOC that point to Account Promotion records in S_SRC to point also to Account Promotion Category records.

Index

A

access control

about 139

S_CONTACT and S_ORG-GROUP, about multi-org visibility for 139

aggregate categories (new), reviewing 283

AIX memory allocation segment space, securing 166

applet functionality, reviewing 275

attachments

file system attachments, updating 305

audience for guide 123

B

binary sort order

Microsoft SQL Server, verifying database was created using 185

business component definitions, changes to party model 131

C

clustered indexes

rebuilding 188

configuration parameters, about and table 187

copied objects, automatic upgrade of
upgrade ancestor, choosing 97

custom database schema

license keys, adding new 236

Customer Repository, preparing for merge

inheriting upgrade behavior, about 96

New Customer repository, preparing to import 227, 259

upgrade inheritance, configuration steps 249

D

data migration

party model, support of 131

database server configuration, verifying

Oracle database server configuration 180

database sort order, verifying

about 129

IBM DB2 UDB, verifying database created using identity sort 170

Microsoft SQL Server, verifying sort order on 185

database, preparing for upgrade

AIX memory allocation segment space, securing 166

binary sort order, verifying database created using 185

database sort order, verifying 129

Microsoft SQL Server, verifying sort order on 185

mobile and dedicated users, preparing for upgrade 198

New Customer Repository, preparing to import 227, 259

RDBMS software, upgrading 127

dedicated users, preparing for upgrade 198

E

Encryption Upgrade Utility

RC2 encryption, about upgrading to 137

F

file system

attachments, updating 305

G

Generate Reporting Relationships button, exposing 312

global deployment, postupgrade tasks

global time zone, about setting up environment to support 313

global time zone, enabling after an upgrade 313

Unicode, migrating to 135

grid-based applets, reviewing 276

guide

audience for 123

naming and typographical conventions 17

I

IBM DB2 UDB, preparing database for upgrade

16-KB temporary tablespace, creating 173

32-KB temporary tablespace, creating 173

database configuration parameters, about and table 172

DB2 permission, verifying on AIX and Solaris 172

- DB2 UDB Application Development Client, verifying installation of 176
- identity sort order, verifying database created using 170
- instance owner permission, verifying for AIX and Solaris 173
- RDBMS software, upgrading 127
- sort order, verifying on 170

ICL, about 102**identity sort order, verifying database created using** 170**Incorporate Custom Layouts, about indexes** 102

- clustered indexes, rebuilding 188

inheriting upgrade behavior, about 96**item identifier, Web template** 291**L****license keys**

- custom database schema, adding new keys 236

log files

- archiving manually 244
- Siebsrvr log files, about saving 235

M**Microsoft SQL Server**

- clustered indexes, rebuilding 188
- configuration parameters, about and table 187
- database sort order, verifying 185
- Siebel Software Configuration utility, changing language utility runs 214
- temporary database space, about 186

mobile users, preparing for upgrade 198**multilingual deployment, installing additional languages** 134**multilingual seed data**

- Siebel Database Server for DB2, installing 134

multi-value group shuttle applets, reviewing 279**N****naming conventions** 17**O****objects**

- descendents or copies, viewing 249

Oracle database

- server configuration, before upgrading tasks 180

- sort order, verifying on 179
- upgrading, steps before upgrading 180

P**party model**

- about 130
- business component definitions 131
- data migration, support of 131
- data model, changes to (diagram) 131
- obsolete tables in 7.x 131

planning

- Configurator 158, 159
- Employee Relationship Management (ERM), about 146
- guidelines 123
- handheld devices 156
- Resonate Central Dispatch 157
- Workflow Designer 152

postupgrade tasks

- asset-based ordering 326
- dock object and rule definitions, verifying are correct 302
- file system attachments, updating 305
- Generate Reporting Relationships button, exposing 312
- inbound workflows, updating 331
- log files, manually archiving 244
- previously unused tables now used in 7.x, examples of 301
- Purchase Orders 325
- reporting relationships, generating (procedure) 312
- reporting relationships, generating, about 312
- symbolic string model, upgrading to 296
- tables not used in 7.x, examples of 301
- Transaction Amount field in Payment lines, updating 325
- Universal Inbox 324

pre-upgrade tasks

- AIX memory allocation segment space, securing 166
- binary sort order, verifying database created using 185
- database sort order, verifying 129
- Microsoft SQL Server, verifying sort order on 185
- mobile and dedicated users, preparing for upgrade 198
- New Customer Repository, preparing to import 227, 259
- RDBMS software, upgrading 127

production environment upgrade

- development environment, upgrading
 - without 235
- global time zone, about operating in 313
- New Customer Repository, preparing to
 - import 227, 259
- preparing production upgrade on UNIX 221
- Transaction Amount field in Payment Lines,
 - updating 325

R

RC2 encryption, upgrading to

- about 137

RDBMS software, preparing for upgrade 127

reporting relationships

- about generating 312
- Generate Reporting Relationships button,
 - exposing 312
- generating (procedure) 312

repositories, upgrading

- about 91
- EIM temporary columns, generating 257
- Incorporate Custom Layouts (ICL) 102
- performing merge 252
- postmerge utilities, about 98
- renaming 213
- upgrade behavior, inheriting 96

repository merge results

- determining success 255
- reviewing conflicts 258
- reviewing deleted objects 295
- reviewing obsolete objects 295

rich text controls, revising 282

RTCs, revising 282

S

S_CONTACT, multi-org visibility for 139

S_LIT files, upgrading to S_CB_ASSET_VER files 305

S_ORG_GROUP, multi-org visibility for 139

S_SRC_PAYMENT table, about columns added 200

seed data, installing multilingual seed data 134

shuttle applets, reviewing 279

Siebel Application Integration (EAI), about using SAP and upgrading 55

Siebel Business application, upgrading - multilingual deployment, installing additional languages 134

Siebel data model

- access control, about 139
- party model, business component
 - definitions 131

- party model, changes to data model
 - (diagram) 131
- party model, data migration in support
 - of 131
- party table, about 130
- party table, obsolete tables in 7.x 131
- S_SRC_PAYMENT Table, columns added
 - to 200

Siebel Database Schema, upgrading

- restarting upgrade 224

Siebel Database Server software

- multilingual deployment, installing additional languages 134

Siebel Financial Services

- household data integrity, verifying 193

Siebel Marketing

- obsolete 7.x business objects 359
- obsolete 7.x data 353
- obsolete 7.x views 360

Siebel Purchase Orders, updating

- Transaction Amount field 325

Siebel Repository, preparing for upgrade

- development environment repositories,
 - preparing 213
- dock objects and visibility rules,
 - recording 165

Siebel Software Configuration utility

- language, changing 214
- The Siebel Upgrade Wizard, about 68

Siebel Upgrade Wizard

- about 68

Siebel Workflow

- inbound workflow, updating 331

SiebSrvr log files, about saving 235

sort order

- binary sort order, verified database created
 - using 185
- Microsoft SQL Server, verifying on 185

T

tables, preparing for upgrade

- DB2 UDB long columns, identifying for
 - truncation 127, 129, 164, 177, 180, 198, 227, 259

Tools repository, renaming 213

Transaction Amount field in Payment lines, updating 325

typographical conventions 17

U

Unicode

- migrating to, about and procedure 135
- migration, planning considerations 136

- upgrade inheritance, configuration**
 - steps 249
- upgrade, planning for**
 - resources 121
- upgrading ancestor, choosing an** 97
- upgrading inheritance**
 - descendent or copies of an object,
 - viewing 249
 - parent to descendents, propagating changes
 - from 98

V

- view tabs, verifying display** 287
- visibility filters, reviewing** 284

W

- Web template item identifier** 291
- workflows**
 - inbound, updating 331