



Upgrade Guide

Version 7.8 Rev E

July 2006

ORACLE®

Copyright © 2005, 2006, 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 Siebel 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 Upgrade Topics

- How the Siebel Upgrade Topics Are Organized 19
- About Siebel Upgrade Topic Applicability 20
- About Terms Used in Siebel Upgrade Topics 20
- About File Paths and Commands in Siebel Upgrade Topics 21

Chapter 3: How the Siebel Database Upgrade Works

- About Supported Siebel Upgrade Paths 25
- About Siebel Upgrade Environments 26
- About the Siebel Database Upgrade Process 28
- About the Siebel Database Server Configuration Utilities 32
- About the Siebel Upgrade Wizard and Driver Files 36
- Example of a Siebel Development Upgrade Flow 39
- About the Siebel Repository Merge 45
- About Inheriting Upgrade Behavior in a Siebel Upgrade 49
- About the Siebel Postmerge Utilities 52
- About the Siebel Incorporate Custom Layout (ICL) Upgrade Option 55
- About the Siebel Database Server 64

Chapter 4: How to Perform a Siebel Database Upgrade

- Road Map for Performing a Siebel Database Upgrade 67
- Process of Planning a Siebel Database Upgrade 70
- Process of Upgrading a Siebel Development Environment 71
- Process of Upgrading a Siebel Production Test Environment 77
- Process of Tuning Siebel Upgrade Performance 83
- Process of Upgrading a Siebel Production Environment 86

Chapter 5: Siebel Database and UI Upgrade Planning

Important Siebel Database Upgrade Planning Resources	93
Best Practices for Doing Your Siebel Database Upgrade	95
New Siebel Upgrade Features and Requirements	98
About Upgrading Your RDBMS in the Siebel Environment	98
About Siebel Multilingual Deployments	99
About Siebel Unicode Support	100
Upgrading to Siebel RC2 or AES Encryption	102
About Siebel User Interface Changes	102
Upgrade Planning for Siebel Web Template Files and Style Sheet	103
About Database Sort Order in the Siebel Environment	104
About Upgrading Siebel Access Control	105
About the Siebel Party Model	106
About Migrating Siebel HTML Attachments to Base Tables	110

Chapter 6: Application Planning for a Siebel Upgrade

Determining Your Siebel 7.5.x Release Level	111
Upgrade Planning for Migrating Siebel Address Data	113
Upgrade Planning for Siebel Employee Relationship Management (ERM)	114
Upgrade Planning for Siebel Marketing	115
Upgrade Planning for Siebel Workflow Designer	121
Upgrade Planning for Handheld Devices in the Siebel Environment	122
Upgrade Planning for Resonate Central Dispatch in the Siebel Environment	122
Upgrade Planning for Siebel String Translation	123
Upgrade Planning for Siebel Configurator	123
Upgrade Planning for Siebel Personalization	124
Upgrade Planning for Siebel Pricer and Order Management	124
Upgrade Planning: Additional Siebel Application Changes	125

Chapter 7: Basic Database Preparations for a Siebel Upgrade

Verifying Siebel Database Connectivity	127
--	-----

Preparing Siebel Tables and Views for Upgrade	127
Preparing Siebel Custom Indexes for Upgrade	128
Exporting Siebel Interface Table Data	129
Archiving Unneeded Siebel Repositories	129
Preserving Siebel Dock Objects and Visibility Rules	130
Securing AIX Memory Allocation Segment Space for the Siebel Database	130

Chapter 8: Preparing an IBM DB2 Database for a Siebel Upgrade

Verifying the IBM DB2 Client for a Siebel Upgrade	131
Verifying IBM DB2 Sort Order for a Siebel Upgrade	131
Setting IBM DB2 Parameters for a Siebel Upgrade	132
Verifying IBM DB2 Permissions for a Siebel Upgrade	134
Verifying IBM DB2 Instance Owner Permissions for a Siebel Upgrade	134
Creating IBM DB2 Temporary Tablespaces and Bufferpools for a Siebel Upgrade	135
Analyzing IBM DB2 Custom Tablespace Requirements for a Siebel Upgrade	135
Verifying the DB2 Application Development Client for a Siebel Upgrade	138
Identifying IBM DB2 Long Columns for Truncation in a Siebel Upgrade	138

Chapter 9: Preparing an Oracle Database for a Siebel Upgrade

Verifying Oracle Database Sort Order for a Siebel Upgrade	141
Verifying Oracle Database Configuration for a Siebel Upgrade	142
Verifying Oracle Database Parameters for Multiple CPUs in a Siebel Upgrade	143
Verifying the Oracle Database ODBC Definition for a Siebel Upgrade	143
Setting Oracle Database Optimizer Mode for a Siebel Upgrade	145

Chapter 10: Preparing an MS SQL Server Database for a Siebel Upgrade

Verifying MS SQL Server Sort Order for a Siebel Upgrade	147
Setting MS SQL Server Temporary Space Size for a Siebel Upgrade	148
Setting MS SQL Server Configuration Parameters for a Siebel Upgrade	148
Rebuilding MS SQL Server Clustered Indexes for a Siebel Upgrade	150

Chapter 11: Preparing Siebel Application Data for Upgrade

- Preparing Siebel Workflow Processes for Upgrade 151
- Identifying Siebel Seed Data Customizations 152
- Preparing Siebel Customized Seed Data for Upgrade 158
- Migrating Siebel Household Data 160
- Setting Up Campaign Status Values for Siebel Marketing 162
- Preserving Siebel Marketing Segment Descriptions 164
- Preparing Siebel Mobile User Data for Upgrade 164
- Setting the Value of Siebel S_SRC_PAYMENT.TYPE_CD for Upgrade 166
- Preparing Siebel Address Data for Upgrade 166
- Migrating Siebel Address Data from Custom Extension Columns 167
- Preparing Siebel Customizable Product Data for Upgrade 174

Chapter 12: Upgrading the Siebel Database

- Renaming the Siebel Repository 177
- Changing the Siebel Database Server Configuration Utilities Language 178
- Preparing to Run the Siebel Database Server Configuration Utilities 179
- Running the Siebel Database Server Configuration Utilities Under Windows 182
- Running the Siebel Database Server Configuration Utilities Under UNIX 183
- Starting the Siebel Upgrade Wizard 184
- Stopping the Siebel Upgrade Wizard 186
- Regenerating SQL Files for a Siebel Upgrade 187
- Identifying and Dropping Obsolete Indexes for a Siebel Upgrade 187
- Preparing for a No-Development-Environment Siebel Upgrade 188
- Installing New Siebel License Keys During an Upgrade 189

Chapter 13: Reviewing the Siebel Upgrade Log Files

- About the Siebel Database Upgrade Logs 191
- Summarizing Siebel Log Files Using Logparse 192
- Reviewing Siebel Upgrade Log Files for Errors 195
- Manually Archiving Siebel Upgrade Log Files 197

Chapter 14: Performing the Siebel Repository Merge

Configuring Siebel Repository Objects to Inherit Upgrade Behavior	199
Configuring Siebel Tools for the Repository Merge	200
Performing a Siebel Repository Merge	201
Determining if a Siebel Repository Merge was Successful	205
Generating Siebel EIM Temporary Columns	207
Reviewing Siebel Repository Object Property Conflicts	208
Regenerating the Siebel Repository Definition Files	209
Moving the Siebel Repository Files	211
Setting Label Alignment for Siebel Text Fields	212
Copying UI Files to a New Siebel Environment	214
Deleting Unneeded Siebel Repository Files	215
Migrating Siebel Repository Objects to the Standard UI	215
Running the Siebel Postmerge Utilities	220

Chapter 15: Reviewing the Siebel User Interface

Troubleshooting Postmerge Siebel UI Problems	223
Verifying Siebel Business Address Applet Configuration	225
Reviewing Siebel Grid-Based Applets	225
Reviewing Siebel UI Navigation	226
Reviewing Siebel Multi-Value Group (MVG) Shuttle Applets	227
Revising Siebel UI Rich Text Controls	229
Reviewing New Siebel UI Aggregate Categories	231
Revising Siebel Visibility Filters to Display Correctly	231
Assigning a Category and Type to Siebel Chart Views	232
Assigning a Category and Type to Siebel Explorer Views	233
Setting Up Navigation to Inaccessible Siebel Detail Views	234
Eliminating Obsolete Siebel UI Fields	235
Reviewing Siebel UI Objects Affected by Incorporate Custom Layout	235
Reviewing Required Fields in the Siebel UI	236
Assigning an Item Identifier to Siebel Web Template Items	237

Chapter 16: Siebel Postmerge Development Tasks

- Reviewing Objects Deleted from the Siebel Repository 239
- Reviewing Obsolete Objects in the Siebel Repository 240
- Migrating Custom Siebel Workflows 241
- Upgrading to the Siebel Symbolic String Model 242
- Dropping IBM DB2 8-KB Tablespaces and Buffers After a Siebel Merge 242
- Updating Siebel Enterprise Application Integration (EAI) 243

Chapter 17: Postupgrade Tasks for the Siebel Database and File System

- Reapplying Schema Customizations in the Siebel Database 245
- Checking for Inactivated EIM Table Columns in the Siebel Database 246
- Updating Siebel File System Attachments 246
- Validating Dock Objects and Rule Definitions in the Siebel Database 248
- Verifying an Upgraded Oracle RDBMS After a Siebel Upgrade 250
- Setting Oracle Database Parameters After a Siebel Upgrade 251

Chapter 18: Postupgrade Tasks for Siebel Applications

- Generating Siebel Reporting Relationships 254
- Setting Up Siebel Global Time Zone Support 255
- Upgrading Siebel ERM Approval Business Process Workflows 256
- Upgrading Siebel ERM Customized Microsite and Group News Pages 257
- Migrating Course Duration Information for Siebel Training 258
- Upgrading the Launch Field in Siebel Training LOV 259
- Upgrading the Test Status in Siebel Training 259
- Verifying Class and Session Times in Siebel Training 260
- Upgrading Responsibilities in Siebel Marketing 261
- Reviewing Siebel Marketing Campaign Data 262
- Reviewing Renamed Fields in Siebel Marketing 262
- Displaying Regions in Siebel Marketing 263
- Revising Program Flowchart Icons in Siebel Marketing 263
- Setting Default Campaign Execution Options in Siebel Marketing 264

Upgrading Activity Plans for Programs and Campaigns in Siebel Marketing	264
Upgrading the Newsletter Offer Type in Siebel Marketing	265
Configuring Universal Inbox in Siebel Marketing	265
Upgrading Siebel Purchase Orders	266
Configuring Siebel Asset-Based Ordering	266
Reviewing Siebel Address Data	268
Upgrading Siebel Attribute Pricing	269
Verifying Aggregate Discounts in Siebel Pricer	270
Upgrading Siebel Seeded Workflows	271
Upgrading Inbound Siebel Workflows	271
Migrating Data to the Bankruptcy Status Field in Siebel Financial Services	272

Chapter 19: Tuning the Siebel Upgrade Files

About Tuning Siebel Production Upgrade Files	273
Starting and Stopping Siebel Upgrade Tuner	278
Managing Parallel Threads Using Siebel Upgrade Tuner	280
Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner	282
Transferring UNIX Files for Use by Siebel Upgrade Tuner	285
Rolling Back Siebel Upgrade Tuner Changes	288

Appendix A: Siebel Marketing Upgrade Reference

Obsolete Siebel Marketing Data	291
Obsolete Business Objects in Siebel Marketing	297
Obsolete or Replaced Views in Siebel Marketing	298

Appendix B: Tables Modified or Seeded During a Siebel Upgrade

Important Schema Changes at Siebel 7.8	307
--	-----

Index

1

What's New in This Release

NOTE: Oracle's Siebel 7.8 does not support upgrades from Siebel 6.x, except as noted below. You must first upgrade to Siebel 7.7 and then to Siebel 7.8. For information on upgrading to Siebel 7.7, see the latest Release 7.7 version of Siebel's *Upgrade Guide* on Oracle's Siebel SupportWeb.

IBM z/OS upgrades of Siebel 6.2.1 Financial Services directly to Release 7.8 of Siebel Industry Applications are supported. You do not have to upgrade to Release 7.7 first. To perform this upgrade, see version 7.8 of the *Upgrade Guide for DB2 UDB for z/OS*.

What's New in Upgrade Guide, Version 7.8 Rev. E

Table 1 lists changes in this version of the documentation to support Release 7.8.2 of the software.

Table 1. New Features in Upgrade Guide, Version 7.8 Rev. E

Topic	Description
"About the Siebel Database Upgrade Process" on page 28	Revised topic. Clarified that both the production environment upprep and upgphys are run in the production test environment and redirected to the production environment.
"About the Siebel Repository Merge" on page 45	Revised topic. Topic rewritten and expanded to address customer feedback.
"About the Siebel Postmerge Utilities" on page 52	Revised topic. Topic rewritten and expanded to address customer feedback.
"About the Siebel Incorporate Custom Layout (ICL) Upgrade Option" on page 55	Revised topic. Topic rewritten and expanded to address customer feedback.
"Process of Upgrading a Siebel Production Environment" on page 86	Revised topic. Clarified that the production upgphys is run in the production test environment and redirected to the production environment.
"Important Siebel Database Upgrade Planning Resources" on page 93	Revised topic. Added <i>Data Model Reference</i> to list of publications in the Siebel deployment documentation suite.
"Determining Your Siebel 7.5.x Release Level" on page 111	New topic. Determine your 7.5.x release-level before installing enterprise components.
"Upgrade Planning: Additional Siebel Application Changes" on page 125	Revised topic. Technical Note 511 on SupportWeb lists obsolete objects in the Siebel Repository.

Table 1. New Features in Upgrade Guide, Version 7.8 Rev. E

Topic	Description
"Verifying Oracle Database Configuration for a Siebel Upgrade" on page 142	Revised topic. Set UNDO_MANAGEMENT = MANUAL before the repository merge for Oracle 9i and later.
"Preparing Siebel Mobile User Data for Upgrade" on page 164	Revised topic. Before upgrade, check the Remote Status view and resolve any insert conflicts.
"Regenerating SQL Files for a Siebel Upgrade" on page 187	New topic. Describes how to use the Database Server Configuration Utilities to regenerate SQL files.
"Configuring Siebel Tools for the Repository Merge" on page 200	Revised topic. Clarified when to revise tools.cfg.
"Setting Label Alignment for Siebel Text Fields" on page 212	Revised topic. In main.css make changes to both the .mceLabel and .mceLabel2 sections. Also revise printmain.css so that Quick Print and Print Preview display labels correctly.
"Creating a New SRF File"	Deleted topic. To create an SRF file, refer to <i>Developing and Deploying Siebel Business Applications</i> .
"Setting Oracle Database Parameters After a Siebel Upgrade" on page 251	New topic. Lists database parameters that must be set in init.ora after the upgrade.

What's New in Upgrade Guide, Version 7.8 Rev. D

Table 2 lists changes in this version of the documentation to support Release 7.8.2 of the software.

Table 2. New Features in Upgrade Guide, Version 7.8 Rev. D

Topic	Description
"About the Siebel Upgrade Wizard and Driver Files" on page 36	Revised topic. Topic rewritten to explain role and structure of driver files.
"Example of a Siebel Development Upgrade Flow" on page 39	New topic. Explains the steps in a typical development environment upgrade.
<ul style="list-style-type: none"> ■ "Process of Upgrading a Siebel Development Environment" on page 71 ■ "Process of Upgrading a Siebel Production Test Environment" on page 77 	Revised topic. You must set up migration of address data on custom extension columns after running the Database Server Configuration Utilities in upgrep mode but before starting the Siebel Upgrade Wizard.
"Upgrade Planning for Siebel Pricer and Order Management" on page 124	Revised topic. Table added that lists how Pricer data is upgraded to the new Pricer architecture.

Table 2. New Features in Upgrade Guide, Version 7.8 Rev. D

Topic	Description
"Upgrade Planning: Additional Siebel Application Changes" on page 125	New topic. Lists additional application changes that may affect upgrade planning.
"Archiving Unneeded Siebel Repositories" on page 129	New topic. Export all repositories except the Siebel Repository before doing an upgrep.
"Preparing Siebel Workflow Processes for Upgrade" on page 151	Revised topic. For upgrades from releases prior to Release 7.7, tables S_WF_PROP_VAL and S_WF_STEP_INST must be purged before upgrade.
"Preparing Siebel Customized Seed Data for Upgrade" on page 158	New topic. Customized seed data is preserved during upgrade, even if doing so adversely affects application function. Prior to upgrade, determine whether to retain seed data customizations.
"Migrating Siebel Address Data from Custom Extension Columns" on page 167	Revised topic. Clarified which steps to perform for a development environment upgrade versus a production test environment upgrade.
"Updating Siebel File System Attachments" on page 246	Revised topic. Run chng_file_sys.bat on each obsolete table.

What's New in Upgrade Guide, Version 7.8 Rev. C

Table 3 lists changes in this version of the documentation to support Release 7.8.2 of the software.

Table 3. New Features in Upgrade Guide, Version 7.8 Rev. C

Topic	Description
"About the Siebel Database Server Configuration Utilities" on page 32	Revised topic. Expanded description of utility modes. Describes how to do a production upgrade using the SQL from the production test environment.
"Road Map for Performing a Siebel Database Upgrade" on page 67	Revised topic. There are three strategies you can use to upgrade from Release 7.8.1 to 7.8.2.
"Process of Upgrading a Siebel Development Environment" on page 71	Revised topic. Process tasks are organized into groups and placed in chapters. Some process steps now refer to all the tasks in a chapter.
"Process of Upgrading a Siebel Production Test Environment" on page 77	Revised topic. Process tasks are organized into groups and placed in chapters. Some process steps now refer to all the tasks in a chapter.
"Process of Tuning Siebel Upgrade Performance" on page 83	New topic. Provides a checklist for using Upgrade Tuner in the production test environment to tune upgrade performance.

Table 3. New Features in Upgrade Guide, Version 7.8 Rev. C

Topic	Description
"Process of Upgrading a Siebel Production Environment" on page 86	Revised topic. Process tasks are organized into groups and placed in chapters. Some process steps now refer to all the tasks in a chapter.
"Upgrade Planning for Siebel Personalization" on page 124	New topic. Primary user role is replaced by primary user responsibility at Release 7.7. To ensure applet visibility, review conditional expressions in the Administration-Personalization screen and revise conditional expressions as needed after the upgrade.
"Verifying Siebel Database Connectivity" on page 127	New topic. From the production test environment, you must be able to connect to the Siebel Database in both the development environment and the production environment.
"Verifying Oracle Database Configuration for a Siebel Upgrade" on page 142	Revised topic. The setting for db_cache_size is 394264576.
"About the Siebel Incorporate Custom Layout (ICL) Upgrade Option" on page 55	Revised topic. Clarifies how layouts are preserved in an ICL upgrade.
"About Tuning Siebel Production Upgrade Files" on page 273	Revised topic. Describes how Upgrade Tuner works, when to use Upgrade Tuner, and how it modifies driver files.
"Starting and Stopping Siebel Upgrade Tuner" on page 278	Revised topic. Describes how to exit without saving changes.
"Managing Parallel Threads Using Siebel Upgrade Tuner" on page 280	Revised topic. Describes how to create, edit, and delete parallel threads for table and index creation.
"Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner" on page 282	Revised topic. Describes how to tune an upgrade by deactivating zero-row SQL commands.
"Transferring UNIX Files for Use by Siebel Upgrade Tuner" on page 285	Revised topic. Describes the workflows for transferring UNIX upgrade files to and from a Windows host.
"Rolling Back Siebel Upgrade Tuner Changes" on page 288	New topic. Describes how to roll back changes Upgrade Tuner makes to upgrade files.

What's New in Upgrade Guide, Version 7.8 Rev. B

Table 4 lists changes in this version of the documentation to support Release 7.8.2 of the software.

Table 4. New Features in Upgrade Guide, Version 7.8 Rev. B

Topic	Description
"Important Siebel Database Upgrade Planning Resources" on page 93	Revised topic. Oracle's Siebel SupportWeb now has an installation and upgrade portal page.
"Road Map for Performing a Siebel Database Upgrade" on page 67	Revised topic. Added a Road Map for upgrading from Release 7.8.1 to Release 7.8.2.
"New Siebel Upgrade Features and Requirements" on page 98	Revised topic. You can run Oracle 9i upgrades in CBO or RBO mode.
"Best Practices for Doing Your Siebel Database Upgrade" on page 95	Revised topic. Added steps for determining your upgrade path.
"About Siebel Multilingual Deployments" on page 99	Revised topic. For multilingual deployments, you must manually import language-specific repository strings and seed data after upprep.
"Identifying Siebel Seed Data Customizations" on page 152	New topic. Seed data is overwritten when upgrading from Release 7.8.1 to 7.8.2 or later. Before upgrading, you can run a report to identify seed data customizations.
"Verifying the IBM DB2 Client for a Siebel Upgrade" on page 131	New topic. The 64-bit IBM DB2 client is not supported on the Siebel Server.
"Setting IBM DB2 Parameters for a Siebel Upgrade" on page 132	Revised topic. Topic lists only those DB2 parameters that must be reset specifically for upgrades.
"Setting Oracle Database Optimizer Mode for a Siebel Upgrade" on page 145	Revised topic. You can run Oracle 9i upgrades in either RBO or CBO mode.
"Setting the Value of Siebel S_SRC_PAYMENT.TYPE_CD for Upgrade" on page 166	New topic. S_SRC_PAYMENT.TYPE_CD is a required field. For records where this is null, set a value prior to upgrade.
"Running the Siebel Database Server Configuration Utilities Under UNIX" on page 183	Revised topic. Corrected and clarified procedure steps.
"Preparing to Run the Siebel Database Server Configuration Utilities" on page 179	Revised topic. For multilingual deployments, you must select the primary (base) language when you run the Database Server Configuration Utilities.
"Summarizing Siebel Log Files Using Logparse" on page 192	Revised topic. For steps executed in parallel, logparse adds to the process time only the duration of the longest step.
"Reviewing Siebel Upgrade Log Files for Errors" on page 195	Revised topic. Corrected file names of logs that can be ignored.

Table 4. New Features in Upgrade Guide, Version 7.8 Rev. B

Topic	Description
"About the Siebel Incorporate Custom Layout (ICL) Upgrade Option" on page 55	Revised topic: <ul style="list-style-type: none"> ■ In an ICL merge, new controls from the New Siebel Repository are not copied to applets. ■ In parent list views, you must select a record in the parent list applet to display view tabs. This is true for both ICL and non-ICL upgrades. ■ After an ICL upgrade of objects that use the Upgrade Ancestor property, the layout of descendant objects may differ from ancestor objects.
"Migrating Custom Siebel Workflows" on page 241	New topic. If you are upgrading from Release 7.7.x or 7.8.x, you must migrate custom workflows to the New Customer Repository prior to the development environment upgphys.
"Reviewing Customized Business Components"	Deleted topic. No longer applicable.
Upgrading the IBM DB2 Instance	Deleted topic. No longer applicable.

What's New in Upgrade Guide, Version 7.8 Rev. A

Table 5 lists changes in this version of the documentation to support Release 7.8.1 of the software.

Table 5. New Features in Upgrade Guide, Version 7.8 Rev. A

Topic	Description
"Upgrade Planning for Siebel Web Template Files and Style Sheet" on page 103	New topic. Guidelines for upgrading customized Web template files and stylesheet file (main.css).
"About the Siebel Party Model" on page 106	New topic. Describes how the party model is implemented in Siebel 7.x. Includes information on how Siebel Financial Services household data is handled.
"Upgrade Planning for Siebel Pricer and Order Management" on page 124	New topic. User properties for Pricer integrations are obsolete at Release 7.8. After the upgrade, you must reimplement integrations using new features.
"Verifying the DB2 Application Development Client for a Siebel Upgrade" on page 138	Revised topic. The Application Development Client must be installed on the RDBMS server.
"Migrating Siebel Household Data" on page 160	New topic. Describes how to migrate Siebel Financial Services data to the party model or how to retain the Release 6.x form of associations.

Table 5. New Features in Upgrade Guide, Version 7.8 Rev. A

Topic	Description
"Identifying and Dropping Obsolete Indexes for a Siebel Upgrade" on page 187	New topic. Optional step during the production test environment upgrade for identifying and dropping indexes that may be obsolete.
"About Inheriting Upgrade Behavior in a Siebel Upgrade" on page 49	Revised topic. Added recommendations on setting the Upgrade Ancestor property on applets.
"About the Siebel Incorporate Custom Layout (ICL) Upgrade Option" on page 55	Revised topic. Describes how Web template files and stylesheet file (main.css) are handled during an ICL repository merge. Added recommendations on using ICL with the Upgrade Ancestor property.
"Verifying Siebel Business Address Applet Configuration" on page 225	New topic. For 7.0.x Siebel Industry Application (SIA) upgrades, you must reconfigure the Business Address applet.
"Copying UI Files to a New Siebel Environment" on page 214	New topic. Describes how to move customized Web template files and stylesheet file (main.css) to the production test and production environments.
"Updating Siebel File System Attachments" on page 246	Revised topic. Corrected the description of arguments for chng_file_sys.bat.
"Safeguarding the New Custom Repository Export File"	Deleted topic. Replaced by step in "Moving the Siebel Repository Files" on page 211 .

What's New in Upgrade Guide, Version 7.8

Table 6 lists changes in this version of the documentation to support Release 7.8 of the software.

Table 6. New Features in Upgrade Guide, Version 7.8

Topic	Description
"Upgrade Planning for Migrating Siebel Address Data" on page 113	New topic. The way addresses are stored has been changed at Release 7.8. Review this topic to determine how to handle table customizations.
"Preparing Siebel Customizable Product Data for Upgrade" on page 174	New topic. To migrate products in workspaces, you must release them. Class products must have the orderable flag unchecked.
"Preparing Siebel Address Data for Upgrade" on page 166	New topic. You must check for duplicate row-IDs within and between S_ADDR_PER and S_ADDR_ORG before performing the upprep of the Siebel Database.
"Configuring Siebel Tools for the Repository Merge" on page 200	New topic. If you selected the Incorporate Custom Layout (ICL) option for the previous Release 7.x upgrade, you must return your user interface to the standard look and feel before performing the repository merge.

Table 6. New Features in Upgrade Guide, Version 7.8

Topic	Description
"About the Siebel Incorporate Custom Layout (ICL) Upgrade Option" on page 55	Revised topic. Added guidelines for deciding when to use Incorporate Custom Layout (ICL).
"Migrating Siebel Address Data from Custom Extension Columns" on page 167	Revised topic. The changes required to the ddl.ctl and preschem_sia.sql files to migrate address data in custom extension columns has been updated. Review this topic carefully before migrating address data from custom extension columns.
"Running the Siebel Postmerge Utilities" on page 220	New topic. You must start the postmerge utilities manually.
"Assigning an Item Identifier to Siebel Web Template Items" on page 237	New topic. If you select the ICL option for the upgrade and choose "Label on Top," you must edit main.css to obtain correct label alignment.
"Verifying Class and Session Times in Siebel Training" on page 260	New topic. You must verify class and session start times in Siebel Training after upgrade.
"Reviewing Siebel Address Data" on page 268	New topic. You must review address records after upgrade to eliminate duplicate and obsolete records.
"Upgrading Siebel Attribute Pricing" on page 269	New topic. You must run a business service method to upgrade attribute pricing.
"Verifying Aggregate Discounts in Siebel Pricer" on page 270	New topic. You must verify bundle factors have been correctly upgraded to the new aggregate discounts feature.
"Important Schema Changes at Siebel 7.8" on page 307	New topic. Lists important schema changes in Release 7.8.
"Configuring Siebel Tools for the Repository Merge" on page 200	New topic. Prior to the repository merge on Oracle 9i, you must set SqlStyle = Oracle in tools.cfg.
"Upgrading Siebel ERM Customized Microsite and Group News Pages" on page 257	Revised topic. Corrected error in procedure.
Chapter 15, "Reviewing the Siebel User Interface"	All topics rewritten to improve clarity. Tools procedures added where needed. Three new topics added covering "Issue 7, 8, and 9" sections of postmerge utilities log.
"Migrating Data to the Bankruptcy Status Field in Siebel Financial Services" on page 272	New topic. BK_STATUS_CD is added to S_BANKRUPTCY. This column stores bankruptcy status for use by Siebel Financial Services. If you have implemented a bankruptcy status field, you must migrate the data to BK_STATUS_CD.

2

About Siebel Upgrade Topics

This area contains the following topics:

- [“How the Siebel Upgrade Topics Are Organized” on page 19](#)
- [“About Siebel Upgrade Topic Applicability” on page 20](#)
- [“About Terms Used in Siebel Upgrade Topics” on page 20](#)
- [“About File Paths and Commands in Siebel Upgrade Topics” on page 21](#)

Related Topics

[Chapter 3, “How the Siebel Database Upgrade Works”](#)

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

How the Siebel Upgrade Topics Are Organized

Upgrades: All Siebel upgrades.

Use the road map 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 *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.
- **Road map topic.** A road map is a numbered list of processes. For example, “Road Map 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 “Road Map 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.”

About Siebel Upgrade Topic Applicability

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

The applicability of topics in the book is listed at the beginning of each topic. [Table 7](#) lists the applicability categories and their meaning.

For each topic, only the relevant categories are listed.

Table 7. Topic Applicability Categories

Applicability Category	Meaning
Upgrades	Lists the upgrades to which the topic applies, for example: Upgrades from: Siebel 7.0.x, 7.5.x, & 7.7.x.
Environments	Lists the Siebel environments to which the topic applies. For example, “Environments: Development environment only” means the topic applies only to a development environment upgrade. For more information on Siebel environments, see “About Siebel Upgrade Environments” on page 26 .
Databases	Lists the databases to which the topic applies. For example, “Databases: IBM DB2 only” means the topic applies only to IBM DB2 databases.
Platforms	Lists the operating system platform to which the topic applies. For example, “Platforms: Windows only” means the topic applies only to Windows platforms.

About Terms Used in Siebel Upgrade Topics

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

This guide follows several naming conventions:

- Current release means the currently shipping release of the Siebel Business Applications.

- Siebel 7.x refers collectively to all versions of Siebel 7 prior to the current release, for example Release 7.0.4 or 7.5.3. See the Upgrade section in *System Requirements and Supported Platforms* on Siebel SupportWeb for specific information on which release numbers are meant by Release 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 section in *System Requirements and Supported Platforms* on Siebel SupportWeb 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 upgrade to the current release. See the Upgrade section in *System Requirements and Supported Platforms* on Siebel SupportWeb for specific information on which release numbers are meant by Release 6.x.
- The term “Windows” refers to all Microsoft Windows operating systems listed as supported for this release in *System Requirements and Supported Platforms* on Siebel SupportWeb.
- The term “UNIX” refers to all forms of the UNIX operating system supported for this release in *System Requirements and Supported Platforms* on Siebel SupportWeb.

About File Paths and Commands in Siebel Upgrade Topics

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

Environment variables and path placeholders for both Windows and UNIX paths are used throughout the *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 *Upgrade Guide* topics:

- **SIEBEL_ROOT** is 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 C:\ as the installation directory for Siebel 7.x, then **SIEBEL_ROOT** is C:\sea7xx\siebsrvr.
- **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 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:\sea7xx\siebsrvr, then **DBSRVR_ROOT** is C:\sea7xx\dbsrvr.
- Throughout the Guide, examples use the path C:\ and C:\sea7xx.

UNIX Paths

The following environment variables and path conventions specify file system locations in *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 7.x, then `$SIEBEL_ROOT` is `/usr/siebel/sea7xx/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 *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/sea7xx/siebsrvr`, then **DBSRVR_ROOT** is `/usr/siebel/sea7xx/dbsrvr`.

- Throughout the Guide, examples use the path `/usr/siebel` and `/usr/siebel/sea7xx`.
- 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 filenames, 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.

3

How the Siebel Database Upgrade Works

This area contains the following topics:

- [“About Supported Siebel Upgrade Paths” on page 25](#)
- [“About Siebel Upgrade Environments” on page 26](#)
- [“About the Siebel Database Upgrade Process” on page 28](#)
- [“About the Siebel Database Server Configuration Utilities” on page 32](#)
- [“About the Siebel Upgrade Wizard and Driver Files” on page 36](#)
- [“Example of a Siebel Development Upgrade Flow” on page 39](#)
- [“About the Siebel Repository Merge” on page 45](#)
- [“About Inheriting Upgrade Behavior in a Siebel Upgrade” on page 49](#)
- [“About the Siebel Postmerge Utilities” on page 52](#)
- [“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 55](#)
- [“About the Siebel Database Server” on page 64](#)

Related Topics

[Chapter 2, “About Siebel Upgrade Topics”](#)

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

About Supported Siebel Upgrade Paths

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

Supported upgrade paths are described in *System Requirements and Supported Platforms* on Siebel SupportWeb.

The *Upgrade Guide* does not cover the following specific upgrade paths or infrastructure changes. Contact Oracle’s Siebel Technical Support or Oracle’s Siebel Expert Services for help with these tasks:

- At Siebel 7.8, direct upgrades from Siebel 6.x are not supported. 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 later of Siebel Industry Applications on IBM z/OS platforms are supported. To perform this upgrade, see *Upgrade Guide for DB2 UDB for z/OS*.

To upgrade from Siebel 6.x to Siebel 7.7, see the latest Siebel 7.7 version of the *Upgrade Guide* on Oracle’s Siebel SupportWeb.

- Changing database platform type during an upgrade, for example changing from Oracle 8i 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.

About Siebel Upgrade Environments

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

The *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 application migrates administrative data such as lists of values (LOVs) from the production test environment to the production environment. For details, see *Going Live with Siebel Business Applications*.
- **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 8](#) gives recommendations for mapping your environments to the ones used in the *Upgrade Guide*.

Table 8. 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.

About the Siebel Database Upgrade Process

Upgrades: All Siebel upgrades.

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. 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 1](#). This flow applies to development and production test environment upgrades. You also use this flow to upgrade your production environment, with the addition of several deployment steps.

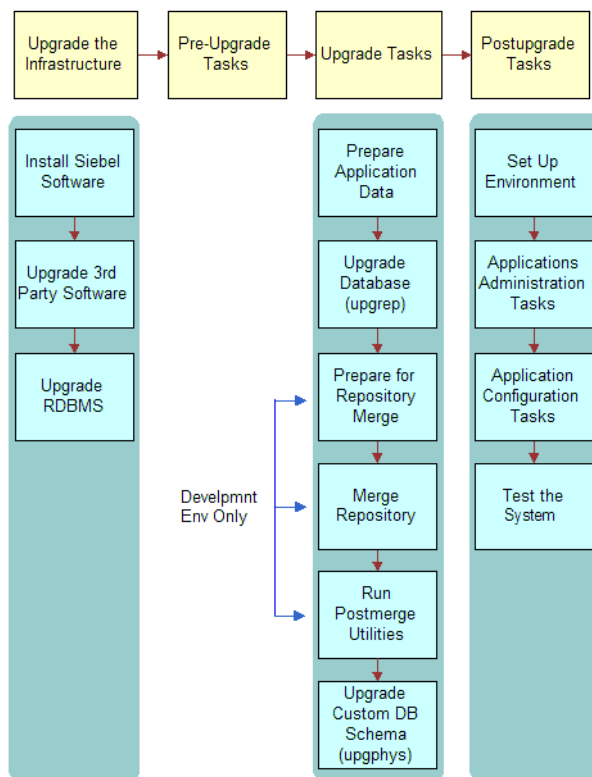


Figure 1. Flow of the Upgrade Process

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 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 Server Configuration Utilities in upgrep mode. This utility performs a basic upgrade of the Siebel Database schema and loads repositories to prepare for the repository merge.
- **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 Server Configuration Utilities in upgphys mode. It further upgrades the Siebel Database with changes resulting from the repository merge and completes the database upgrade.

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

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 Server Configuration Utilities in upgrep mode. It performs 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 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.
- **Upgrade database (upgphys).** You run the Database Server Configuration Utilities in upgphys mode. This utility completes the upgrade and performs miscellaneous administrative tasks.

Perform Upgrade Tasks (Production Environment)

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

- **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.

Upgrade database (upgphys). You run the Database Server Configuration Utilities in upgphys mode. You do this in the production test environment in the same fashion as running the upgrep. This utility completes the upgrade and performs miscellaneous administrative tasks.

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 applications and include such things 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 application function. 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 the Siebel Database Server Configuration Utilities" on page 32](#)

["About the Siebel Upgrade Wizard and Driver Files" on page 36](#)

["About the Siebel Repository Merge" on page 45](#)

["About the Siebel Database Server" on page 64](#)

About the Siebel Database Server Configuration Utilities

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

The Database Server Configuration Utilities performs the upgrep and upgphys portion of an environment upgrade. You must also run the Database Server Configuration Utilities in Prepare for Production mode during the production test environment upgrade.

The Database Server Configuration Utilities requests information from you about the upgrade environment and creates an upgrade configuration file. It then calls a driver that uses the environment information to set up SQL scripts required to upgrade your database.

After you run the Database Server Configuration Utilities, you run the Upgrade Wizard. The Upgrade Wizard opens a driver file containing the steps for the upgrade and executes the steps.

To upgrade a development environment, production test environment, and production environment, you must run the Database Server Configuration Utilities (and Upgrade Wizard) several times, as shown in [Table 9](#).

Table 9. Database Server 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	Production	upgrep
Prod. test env. upgphys	Production	upgphys
Production env. upgrep	Production	upgrep
Production env. upgphys	Production	upgphys

Figure 2 shows how the Database Server Configuration Utilities (and Upgrade Wizard) work together with the Siebel Tools repository merge to upgrade your environments.

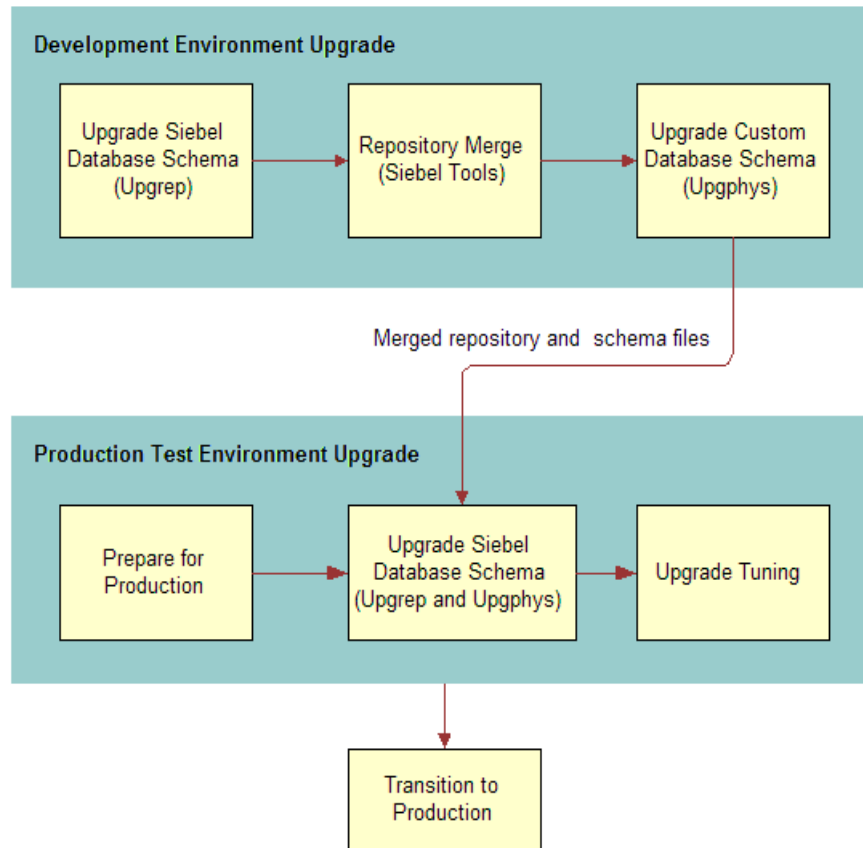


Figure 2. Database Server Configuration Utilities and Siebel Tools Repository Merge

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:
 - Drops 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

How the Siebel Database Upgrade Works ■

- 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 drop these indexes.
- **upgprep.** This mode makes the following changes:
 - Drops 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
 - Performs miscellaneous file actions
- **upgphys.** This mode makes several administrative changes to table data, including updating the schema version in `S_APP_VER`. It does not make schema changes.

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
- **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 Server Configuration Utilities. Enter information for the production environment (not the production test environment), including the new ODBC definition.

The Database Server Configuration Utilities updates 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.

- **upgphys.** Run the Database Server Configuration Utilities. Enter information for the production environment (not the production test environment), including the new ODBC definition.

The Upgrade Wizard uses the upgrade SQL commands in the production test environment to upgrade the production environment Siebel Database.

The upgphys step makes the same changes as when it ran in the production test environment.

How the Upgrade Configuration File and SQL Files Are Created

When you run the Database Server Configuration Utilities, it does the following:

- Collects configuration information
- Creates a driver upgrade configuration file (UCF). This file maps the information you entered in the Database Server Configuration Utilities to environment variables. When the Upgrade Wizard is performing the steps in a driver file, it uses these variables to execute the command contained in each step.
- Forwards 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.
- Prompts you to start the Upgrade Wizard.

In some cases, you will need to modify the generated SQL files as required by Release Notes, Technical Notes, or Alerts 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* = the upgrade process you are performing
 - Siebel Database schema upgrade = *upgrep*
 - Custom database schema upgrade = *upgphys*
 - Prepare for production upgrade = *prepare_for_production_upgrade*
- *ENVIRONMENT* = the environment that you are upgrading
 - Development environment upgrades = *dev*
 - Production environment upgrades = *prod*
- *VERSION* = the version from which you are upgrading

For example, if you are upgrading from Siebel 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 Topic

[“About the Siebel Upgrade Wizard and Driver Files” on page 36](#)

About the Siebel Upgrade Wizard and Driver Files

Upgrades: All Siebel upgrades.

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_77.ucf
- driver_upgphys_dev_77.ucf
- driver_upgrep_prod_77.ucf
- driver_upgphys_prod_77.ucf.

Here is an excerpt from a driver file that controls a development environment upgrep from Siebel 7.7 on Oracle. 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 $DbsrvrRoot/$DatabasePlatform/dropif-db.sql /v
Number of 10 Second Wait Loops = 2000
Return Code = 0
```

How the Siebel Database Upgrade Works ■

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/\$DatabasePlatform/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* = the database type, for example ORACLE

■ *VERSION* = the version from which you are upgrading, for example V7.7

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

driver_upgrep_dev_77.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 28](#)

[“About the Siebel Database Server Configuration Utilities” on page 32](#)

[“Example of a Siebel Development Upgrade Flow” on page 39](#)

[“About the Siebel Repository Merge” on page 45](#)

Example of a Siebel Development Upgrade Flow

Upgrades: All Siebel upgrades.

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 10](#) 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 10. Example of Steps in a Development Environment Upgrep

Step	Script or Input File	Comment
Determine collation sequence of database.	N/A	Determines database sort order.
Verify repository name	rename_existing_repositories.sql	Renames Siebel Repository to Prior Customer Repository.
Drop interface tables	dropif-db.sql	Drops all EIM tables.

Table 10. Example of Steps in a Development Environment Upgrep

Step	Script or Input File	Comment
Drop database triggers	trigdrop-db.sql	Drops all dynamically created triggers.
Drop database-level functions and procedures	drop_db_func_proc.sql	Drops the exchange rate function exrate.
Prepare for table creation	pret.sql	Drops 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.ctl 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.
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.

Table 10. Example of Steps in a Development Environment Upgprep

Step	Script or Input File	Comment
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.
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>

Table 10. Example of Steps in a Development Environment Upgrop

Step	Script or Input File	Comment
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.
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 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, that 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>

Table 10. Example of Steps in a Development Environment Upgrade

Step	Script or Input File	Comment
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.
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 10. 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 10. Example of Steps in a Development Environment Upgrep

Step	Script or Input File	Comment
Create database-level functions and procedures	db_func_proc.sql	Creates or replaces 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 Server Configuration Utilities" on page 32](#)

["About the Siebel Upgrade Wizard and Driver Files" on page 36](#)

About the Siebel Repository Merge

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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.

How the Siebel Database Upgrade Works ■

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 11](#). These repositories are required for the repository merge.

Table 11. 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
- 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 13 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 12. 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	Standard	Customized
Standard	Customized	Standard/ Inactive	Customized/Inactive
Standard	Customized	Revised	Revised (conflict)
N/A	New	N/A	New
N/A	N/A	New	New
Standard	Deleted	Standard	Standard
Standard	Deleted	Standard/ Inactive	Standard/Inactive
Standard	Deleted	Revised	Revised

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 55](#)

["About Inheriting Upgrade Behavior in a Siebel Upgrade" on page 49](#)

["About the Siebel Postmerge Utilities" on page 52](#)

["Reviewing Siebel Repository Object Property Conflicts" on page 208](#)

About Inheriting Upgrade Behavior in a Siebel Upgrade

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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 45](#)

[“About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option” on page 55](#)

About the Siebel Postmerge Utilities

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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 `reutility.xml`:

Windows: `SIEBEL_TOOLS_INSTALL_DIR\repatch\reutility.xml`

UNIX: `SIEBEL_TOOLS_INSTALL_DIR/repatch/reutility.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.
- **CSSUINavUpgradeReposPrep**. Converts customized screens to the Siebel 7.7 UI navigation scheme.
- **CSSMVGUpgradePatch77**. Enables the shuttle feature in customized MVGs.
- **CSSUIUpgradeReports**. 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 xyyyy, 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 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 logs output to the `reputility.log` section called Invalid Applet Web Template Item Mapping Cleanup.

CSSUINavUpgradeReposPrep

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.

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 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 logs output to the `reputility.log` section called User Interface Navigation Upgrade.

CSSMVGUpgradePatch77

As of Siebel 7.7, MVG applets with a 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 inactivated in the New Customer Repository and so display as shuttle applets.

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

CSSUIUpgradeReports

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 `webtemp1` to `\temp\webtemp1`.
- The utility copies Web template files from a subdirectory of `reppatch\web_templates` to `\webtemp1`.

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 logs 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 45](#)

["About Inheriting Upgrade Behavior in a Siebel Upgrade" on page 49](#)

["About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option" on page 55](#)

About the Siebel Incorporate Custom Layout (ICL) Upgrade Option

Upgrades from: Siebel 7.x using ICL.

Environments: Development environment only.

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 inactivates 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 will 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 you should use ICL:

- ICL can only be used for upgrades from Siebel 7.x. If you are upgrading from Siebel 6.x to 7.7, you cannot use ICL.
- 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 your customizations are heaviest.

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.)

How the Siebel Database Upgrade Works ■

- 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 13 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 the table are described in a section following the table.

Table 13. 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

Table 13. 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	Customized	Revised	Revised (conflict) See 2 in Table Exceptions	Customized
N/A	New	N/A	New	New
N/A	N/A	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 13](#) are as follows:

- 1 In an ICL merge, UI navigation changes at Siebel 7.7 are implemented in screens copied-in from the Prior Customer Repository.
- 2 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.

ICL and the Upgrade Behavior Property

Certain screens, views, and applets required to administer applications and support standard application function are not eligible for ICL handling. 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 since their layout and functionality has 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

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

- If this property is set to NULL or Preserved then the UI object is preservable.
- If this property is set to ADMIN, then the UI object is not eligible for ICL handling.

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

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 to 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 to 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:

- Which web template is used for an applet is preserved. 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.
 - What controls are visible is preserved.
 - 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.
 - What columns have drilldowns is preserved
 - Destination views of drilldowns are preserved

The following things 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 Siebel 7.x to Release 7.7 or later, 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 7.7, this navigation scheme is implemented during 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. They 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 will display 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.

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

On an ICL merge, the regular merge portion propagates upgrade handling of non-ICL-object types (Business Component, etc.) 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 inactivated, 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 active.

The ICL merge step writes status and issues to the merge logs.

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 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 `webtemp1` to `\temp\webtemp1`.
- The utilities copy web template files from a subdirectory of `reppatch\web_templates` to `\webtemp1`.

These Web template files support the UI layout 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 postmerge utilities copy Web template files from the `753` and `TopLabel` subdirectories of `reppatch`.

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 New Customer Repository for the current release (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 45](#)

["About Inheriting Upgrade Behavior in a Siebel Upgrade" on page 49](#)

["About the Siebel Postmerge Utilities" on page 52](#)

["Reviewing Siebel Repository Object Property Conflicts" on page 208](#)

About the Siebel Database Server

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

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:\sea7xx\siebsrvr` (Windows), then the Siebel Database Server will be installed in `C:\sea7xx\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.

4

How to Perform a Siebel Database Upgrade

This area contains the following topics:

- ["Road Map for Performing a Siebel Database Upgrade" on page 67](#)
- ["Process of Planning a Siebel Database Upgrade" on page 70](#)
- ["Process of Upgrading a Siebel Development Environment" on page 71](#)
- ["Process of Upgrading a Siebel Production Test Environment" on page 77](#)
- ["Process of Tuning Siebel Upgrade Performance" on page 83](#)
- ["Process of Upgrading a Siebel Production Environment" on page 86](#)

Related Topics

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

[Chapter 3, "How the Siebel Database Upgrade Works"](#)

Road Map for Performing a Siebel Database Upgrade

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

Use one of the road maps listed below to guide you through the steps for upgrading your Siebel Database.

Each road map 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 road map. When you have completed all the processes in the road map, 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 the most applicable road map to perform the upgrade:

- ["Road Map for Upgrading from Siebel 6.x" on page 68](#)
- ["Road Map for Upgrading from Siebel 7.8.1 to Siebel 7.8.2" on page 68](#)
- ["Road Map for Upgrading from Siebel 7.x or Siebel 7.8.1 on IBM DB2 UDB for z/OS" on page 69](#)
- ["Road Map for Upgrading From Siebel 7.x if You Have a Development Environment" on page 69](#)
- ["Road Map for Upgrading From Siebel 7.x Without a Development Environment" on page 70](#)

Road Map 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 *Upgrade Guide* on Oracle's Siebel SupportWeb.

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 *Upgrade Guide for DB2 UDB for z/OS*.

Road Map for Upgrading from Siebel 7.8.1 to Siebel 7.8.2

If you are running Siebel 7.8.1 in your production environment, contact Oracle's Siebel Expert Services for help in upgrading to Siebel 7.8.2.

If you are running Siebel 7.8.1 only in your development or production test environment, choose one of the following strategies to upgrade from Siebel 7.8.1 to 7.8.2.

Upgrade from Siebel 7.8.1 to 7.8.2.

Perform a full upgrade, including repository merge.

- If you selected Incorporate Custom Layout (ICL) to upgrade to 7.8.1 in your development environment, you cannot choose it again when upgrading to 7.8.2. Contact Siebel Oracle's Siebel Expert Services for help if you selected ICL at 7.8.1 and need to preserve your Siebel 7.0.x or 7.5.x UI look-and-feel at 7.8.2.
- Upgrade the development and production test environments, using the 7.8.2 Maintenance Release and any related fixpacks.

You do not need to install a new Siebel Database in these environments.

To install the 7.8.2 Maintenance Release and fixpacks, see *Siebel Installation Guide* for the operating system you are using.
- Upgrade the production environment using the full 7.8.2 Release distribution plus any related fixpacks.
- Use the following road map to upgrade the development, production test, and production environments to Siebel 7.8.2:
 - a "Process of Planning a Siebel Database Upgrade" on page 70
 - b "Process of Upgrading a Siebel Development Environment" on page 71
 - c "Process of Upgrading a Siebel Production Test Environment" on page 77
 - d "Process of Tuning Siebel Upgrade Performance" on page 83
 - e "Process of Upgrading a Siebel Production Environment" on page 86

Return to Siebel 7.x, then upgrade to 7.8.2.

In this strategy, you return your development and production test environment installations to your previously-installed 7.x release. You then perform a standard upgrade to Siebel 7.8.2.

- You can select ICL for this upgrade.

- In your development and production test environments, uninstall Siebel 7.8.1, and install your previously-installed Siebel 7.x Release. For example, uninstall Siebel 7.8.1 and install Siebel 7.7. To uninstall or install a Release, see *Siebel Installation Guide* for the operating system you are using.
- In your development and production test environments, install the Siebel Database from your previously-installed release. For example, in each environment, retrieve and install the 7.7 Siebel Database.
- Upgrade the development, production test, and production environments to Siebel 7.8.2 using the full 7.8.2 Release distribution plus any related fixpacks. Use the following road map to upgrade your environments:
 - a "Process of Planning a Siebel Database Upgrade" on page 70
 - b "Process of Upgrading a Siebel Development Environment" on page 71
 - c "Process of Upgrading a Siebel Production Test Environment" on page 77
 - d "Process of Tuning Siebel Upgrade Performance" on page 83
 - e "Process of Upgrading a Siebel Production Environment" on page 86

Uninstall Siebel 7.8.1 then install 7.8.2.

In this strategy you do not perform an upgrade in the development and production test environments. Instead, you uninstall Siebel 7.8.1 and the Siebel Databases in these environments. Then you install a full Siebel 7.8.2 distribution and new Siebel Database.

- Choose this option if you are evaluating Siebel 7.8.1 and did not have a previous Siebel Release installed.
- To uninstall or install a Release, see *Siebel Installation Guide* for the operating system you are using.

Road Map for Upgrading from Siebel 7.x or Siebel 7.8.1 on IBM DB2 UDB for z/OS

Upgrading from Siebel 7.x or Siebel 7.8.1 on IBM DB2 UDB for z/OS is described in *Upgrade Guide for DB2 UDB for z/OS*.

Road Map 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 road map in the order shown:

- 1 "Process of Planning a Siebel Database Upgrade" on page 70.
- 2 "Process of Upgrading a Siebel Development Environment" on page 71.
- 3 "Process of Upgrading a Siebel Production Test Environment" on page 77.

- 4 ["Process of Upgrading a Siebel Production Environment" on page 86.](#)

Road Map 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 road map 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 70.](#)
- 2 ["Preparing for a No-Development-Environment Siebel Upgrade" on page 188.](#)
- 3 Perform a production test environment upgrade. See ["Process of Upgrading a Siebel Production Test Environment" on page 77.](#)
- 4 Upgrade your production environment. See ["Process of Upgrading a Siebel Production Environment" on page 86.](#)

Process of Planning a Siebel Database Upgrade

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

This process is part of a road map. See ["Road Map for Performing a Siebel Database Upgrade" on page 67.](#)

To plan the upgrade, read the following:

- 1 [Chapter 3, "How the Siebel Database Upgrade Works."](#)
- 2 [Chapter 5, "Siebel Database and UI Upgrade Planning."](#)
- 3 If you are planning to migrate to Unicode, see ["About Siebel Unicode Support" on page 100.](#)
- 4 [Chapter 6, "Application Planning for a Siebel Upgrade."](#)
- 5 [Appendix B, "Tables Modified or Seeded During a Siebel Upgrade"](#)
- 6 [Appendix A, "Siebel Marketing Upgrade Reference"](#)

Remote Upgrade Planning

See the following resources to deploy the upgrade to remote users:

- 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: All Siebel upgrades.

Environments: Development environment only.

This process is part of a road map. See ["Road Map for Performing a Siebel Database Upgrade" on page 67](#).

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.

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 on Oracle's Siebel SupportWeb under Product Documentation > Maintenance Release Guides.

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.

- 1** 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.
- 2** 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.
- 3** Install language packs for your currently deployed languages and any new languages.

- 4 If you have customized the configuration of Enterprise components, such as Siebel Servers, you must manually enter the customizations in the upgraded environment. See *Going Live with Siebel Business Applications*.

Upgrade Third-Party Software

- 5 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

- 6 If required, upgrade the RDBMS version. Refer to the vendor's documentation to perform the upgrade. For information on supported RDBMS systems, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

Pre-Upgrade Tasks for the Siebel Database

These steps apply to all database types.

- 7 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 8 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 9 Stop the Siebel Server and the Siebel Gateway Name Server.
- 10 Verify that there are no open database connections.
- 11 Perform the tasks in [Chapter 7, "Basic Database Preparations for a Siebel Upgrade."](#)

Pre-Upgrade Tasks for IBM DB2

- 12 Perform the tasks in [Chapter 8, "Preparing an IBM DB2 Database for a Siebel Upgrade."](#)
- 13 Execute runstats on the Siebel Database. This will improve upgrade performance.

Pre-Upgrade Tasks for Oracle

- 14 Perform the tasks in [Chapter 9, "Preparing an Oracle Database for a Siebel Upgrade."](#)
- 15 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

- 16** Perform the tasks in [Chapter 10, "Preparing an MS SQL Server Database for a Siebel Upgrade."](#)
- 17** Run MS SQL Server statistics. This will improve upgrade performance.

Pre-Upgrade Tasks for Application Data

- 18** Perform the tasks in Technical Note 521 on Oracle's Siebel SupportWeb. [Table 16](#) shows the applicability of the tasks in the Technical Note.

Table 14. Tasks in Technical Note 521

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

- 19** Perform the tasks in the Technical Notes and Alerts on Oracle's Siebel SupportWeb shown in [Table 17](#).

Table 15. Technical Notes and Alerts

Title	Environment
Technical Note 481	Development only

- 20** Perform the tasks in [Chapter 11, "Preparing Siebel Application Data for Upgrade."](#)

Upgrade Siebel Database Schema (upgrep)

- 21** Verify that all developers have checked in their projects.
- 22** Install Siebel Tools on development workstations. Keep at least one copy of the previous version of Siebel Tools. You will need it to perform repository operations before the repository merge.
- 23** ["Renaming the Siebel Repository" on page 177.](#)
- 24** Back up the development database. (If you backed up the database as part of an RDBMS upgrade, ignore this step.)

- 25** (Optional) [“Changing the Siebel Database Server Configuration Utilities Language” on page 178.](#)
- 26** Run the Database Server Configuration Utilities:
- [“Preparing to Run the Siebel Database Server Configuration Utilities” on page 179.](#)
 - [“Running the Siebel Database Server Configuration Utilities Under Windows” on page 182.](#)
 - [“Running the Siebel Database Server Configuration Utilities Under UNIX” on page 183.](#)
- Choose the following settings:
- **Upgrade Options:** Upgrade Siebel Database Schema (upgrep)
 - **Environment Type:** Development
- 27** Review the following topic and perform the procedure if applicable: [“Identifying and Dropping Obsolete Indexes for a Siebel Upgrade” on page 187.](#)
- 28** Edit generated SQL files as required by Siebel Technical Notes, Alerts, Release Notes or other publications. For a complete list, see the Installation and Upgrade Portal page on Oracle’s Siebel SupportWeb.
- 29** Resume the upgrade: [“Starting the Siebel Upgrade Wizard” on page 184.](#)
- 30** Review the upgrade logs and resolve errors:
- [“Summarizing Siebel Log Files Using Logparse” on page 192.](#)
 - [“Reviewing Siebel Upgrade Log Files for Errors” on page 195.](#)
- 31** 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 Server Configuration Utilities, see [“Regenerating SQL Files for a Siebel Upgrade” on page 187.](#)
 - d** Rerun the Database Server Configuration Utilities.
- 32 Multilingual deployments:** Perform the steps in Technical Note 447 on Oracle’s Siebel SupportWeb to import language-specific repository strings and seed data into the upgrade repositories.
- 33** Back up the upgraded database.

Prepare for Repository Merge

- 34** Set the Upgrade Ancestor property for copied objects. See [“Configuring Siebel Repository Objects to Inherit Upgrade Behavior” on page 199.](#)
- 35** [“Configuring Siebel Tools for the Repository Merge” on page 200.](#)
- 36** (IBM DB2 and MSSQL 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 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.

- 37 [“Performing a Siebel Repository Merge” on page 201.](#)
- 38 [“Determining if a Siebel Repository Merge was Successful” on page 205.](#)
- 39 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.
- 40 [“Running the Siebel Postmerge Utilities” on page 220.](#)
- 41 [“Generating Siebel EIM Temporary Columns” on page 207.](#)
- 42 Reset Siebel Tools configuration. See [“Configuring Siebel Tools for the Repository Merge” on page 200.](#)
- 43 Back up the Siebel Database.

Upgrade Custom Database Schema (upgphys)

- 44 Run the Database Server Configuration Utilities:
 - [“Preparing to Run the Siebel Database Server Configuration Utilities” on page 179.](#)
 - [“Running the Siebel Database Server Configuration Utilities Under Windows” on page 182.](#)
 - [“Running the Siebel Database Server Configuration Utilities Under UNIX” on page 183.](#)Choose the following settings:
 - **Upgrade Options:** Upgrade Custom Database Schema (upgphys)
 - **Environment Type:** Development
- 45 Review the upgrade logs and resolve errors:
 - [“Summarizing Siebel Log Files Using Logparse” on page 192.](#)
 - [“Reviewing Siebel Upgrade Log Files for Errors” on page 195.](#)
- 46 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.
- 47 [“Manually Archiving Siebel Upgrade Log Files” on page 197.](#)
- 48 Back up the upgraded database.

Initialize and Extract Developers' Local Databases

- 49** Install the new release of Siebel Tools on developer's machines.
- 50** Initialize and extract the local database on development machines. For more information, see *Developing and Deploying Siebel Business Applications*.

Review the User Interface

- 51** "Reviewing Siebel Repository Object Property Conflicts" on page 208
- 52** Review the following topics on UI upgrade before proceeding:
 - "About Inheriting Upgrade Behavior in a Siebel Upgrade" on page 49
 - "About the Siebel Postmerge Utilities" on page 52
 - For ICL upgrades: "About the Siebel Incorporate Custom Layout (ICL) Upgrade Option" on page 55
- 53** 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 212.
- 54** Optional. Create a new SRF file to assist in UI testing. For information on creating an SRF file see *Developing and Deploying Siebel Business Applications*.
- 55** Perform the tasks in Chapter 15, "Reviewing the Siebel User Interface."
- 56** 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

- 57** Perform the tasks in Chapter 16, "Siebel Postmerge Development Tasks."
- 58** Resolve any business component and join conflicts.

Postupgrade Tasks for Database and File System

- 59** Perform the tasks in Chapter 17, "Postupgrade Tasks for the Siebel Database and File System."
- 60** 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.
- 61** If you exported data from interface tables before the upgrade, review the database and import the data as desired.
- 62** "Upgrading to Siebel RC2 or AES Encryption" on page 102.
- 63** Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide*.
- 64** (Oracle database 9i and later). Set Optimizer Mode to CBO.

65 Run database statistics.

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

Postupgrade Tasks for Applications Configuration

66 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 106](#).

67 Perform the tasks in [Chapter 18, "Postupgrade Tasks for Siebel Applications."](#)

68 Verify the function of interfaces in integrated applications.

69 Activate and deploy workflows.

70 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 eBusiness Application Integration Volume I*.

71 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

72 Use available test data to perform unit testing. Validate application function in the following areas:

- User interface
- Data interfaces
- Integrity of migrated data
- Workflow function

Prepare for Transition to Production Test Environment

73 Create a new SRF file. For information on creating an SRF file see *Developing and Deploying Siebel Business Applications*.

74 Regenerate the custrep.dat and schema.ddl files. See ["Regenerating the Siebel Repository Definition Files" on page 209](#).

Process of Upgrading a Siebel Production Test Environment

Upgrades: All Siebel upgrades.

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

This process is part of a road map. See ["Road Map for Performing a Siebel Database Upgrade"](#) on page 67.

This topic lists the steps required to upgrade to 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.

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 on Oracle's Siebel SupportWeb under Product Documentation > Maintenance Release Guides.

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.

- 1 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.
- 2 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.
- 3 Install language packs for your currently deployed languages and any new languages.
- 4 If you have customized the configuration of Enterprise components, such as Siebel Servers, you must manually enter the customizations in the upgraded environment. See *Going Live with Siebel Business Applications*.

Upgrade Third-Party Software

- 5 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

- 6 If required, upgrade the RDBMS version. Refer to the vendor's documentation to perform the upgrade. For information on supported RDBMS systems, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

Pre-Upgrade Tasks for the Siebel Database

These steps apply to all database types.

- 7** Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 8** Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 9** Stop the Siebel Server and the Siebel Gateway Name Server.
- 10** Verify that there are no open database connections.
- 11** Perform the tasks in [Chapter 7, “Basic Database Preparations for a Siebel Upgrade.”](#)

Pre-Upgrade Tasks for IBM DB2

- 12** Perform the tasks in [Chapter 8, “Preparing an IBM DB2 Database for a Siebel Upgrade.”](#)
- 13** Execute runstats on the Siebel Database. This will improve upgrade performance.

Pre-Upgrade Tasks for Oracle

- 14** Perform the tasks in [Chapter 9, “Preparing an Oracle Database for a Siebel Upgrade.”](#)
- 15** 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

- 16** Perform the tasks in [Chapter 10, “Preparing an MS SQL Server Database for a Siebel Upgrade.”](#)
- 17** Run MS SQL Server statistics. This will improve upgrade performance.

Pre-Upgrade Tasks for Application Data

- 18** Perform the tasks in Technical Note 521 on Oracle’s Siebel SupportWeb. [Table 16](#) shows the applicability of the tasks in the Technical Note.

Table 16. Tasks in Technical Note 521

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

- 19** Perform the tasks in the Technical Notes and Alerts on Oracle's Siebel SupportWeb shown in [Table 17](#).

Table 17. Technical Notes and Alerts

Title	Environment
Technical Note 481	Development only

- 20** Perform the tasks in [Chapter 11, "Preparing Siebel Application Data for Upgrade."](#)

Prepare the Siebel Database for Upgrade

- 21** Refer to Technical Note 586 on Oracle's Siebel SupportWeb for instructions on how to set up the Siebel Database and Siebel Servers in the production test environment.
- 22** Copy upgrade files to the environment:
- a** ["Moving the Siebel Repository Files" on page 211.](#)
 - b** Custom SRF file.
 - c** Reports files.
 - d** ["Copying UI Files to a New Siebel Environment" on page 214.](#)
- 23** If you revise repository objects or schema definitions, regenerate the schema.ddl and custrep.dat files. See ["Regenerating the Siebel Repository Definition Files" on page 209.](#)
- 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.
- 26** (Optional) ["Changing the Siebel Database Server Configuration Utilities Language" on page 178.](#)
- 27** Verify that you have defined ODBC's for connecting to the development environment database and the production test environment database. See ["Verifying the Oracle Database ODBC Definition for a Siebel Upgrade" on page 143.](#)
- 28** Run the Database Server Configuration Utilities:
- ["Preparing to Run the Siebel Database Server Configuration Utilities" on page 179.](#)
 - ["Running the Siebel Database Server Configuration Utilities Under Windows" on page 182.](#)
 - ["Running the Siebel Database Server Configuration Utilities Under UNIX" on page 183.](#)
- Choose the following settings:
- **Upgrade Options:** Prepare for Production Upgrade

- **Environment Type:** Production

29 (Optional) [“Identifying and Dropping Obsolete Indexes for a Siebel Upgrade” on page 187.](#)

Upgrade Siebel Database Schema (upgrep)

30 Run the Database Server Configuration Utilities:

- [“Preparing to Run the Siebel Database Server Configuration Utilities” on page 179.](#)
- [“Running the Siebel Database Server Configuration Utilities Under Windows” on page 182.](#)
- [“Running the Siebel Database Server Configuration Utilities Under UNIX” on page 183.](#)

Choose the following settings:

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

31 Review the following topic and perform the procedure if applicable: [“Identifying and Dropping Obsolete Indexes for a Siebel Upgrade” on page 187.](#)

32 Edit generated SQL files as required by Siebel Technical Notes, Alerts, Release Notes or other publications. For a complete list, see the Installation and Upgrade Portal page on Oracle’s Siebel SupportWeb.

33 Resume the upgrade: [“Starting the Siebel Upgrade Wizard” on page 184.](#)

34 Review the upgrade logs and resolve errors:

- [“Summarizing Siebel Log Files Using Logparse” on page 192.](#)
- [“Reviewing Siebel Upgrade Log Files for Errors” on page 195.](#)

35 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 Server Configuration Utilities, see [“Regenerating SQL Files for a Siebel Upgrade” on page 187.](#)
- d** Rerun the Database Server Configuration Utilities.

36 **Multilingual deployments:** Perform the steps in Technical Note 447 on Oracle’s Siebel SupportWeb to import language-specific repository strings and seed data into the upgrade repositories.

37 Back up the upgraded database.

Upgrade Custom Database Schema (upgphys)

38 Run the Database Server Configuration Utilities:

- [“Preparing to Run the Siebel Database Server Configuration Utilities” on page 179.](#)
- [“Running the Siebel Database Server Configuration Utilities Under Windows” on page 182.](#)

- ["Running the Siebel Database Server Configuration Utilities Under UNIX" on page 183.](#)

Choose the following settings:

- **Upgrade Options:** Upgrade Custom Database Schema (upgphys)
- **Environment Type:** Production

39 Review the upgrade logs and resolve errors:

- ["Summarizing Siebel Log Files Using Logparse" on page 192.](#)
- ["Reviewing Siebel Upgrade Log Files for Errors" on page 195.](#)

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

- a** Restore the backup of the database.
- b** Correct the errors.
- c** Rerun the Database Server Configuration Utilities.

41 ["Manually Archiving Siebel Upgrade Log Files" on page 197.](#)

42 ["Installing New Siebel License Keys During an Upgrade" on page 189.](#)

43 Back up the upgraded database.

Postupgrade Tasks for Database and File System

44 Perform the tasks in [Chapter 17, "Postupgrade Tasks for the Siebel Database and File System."](#)

45 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.

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

47 ["Upgrading to Siebel RC2 or AES Encryption" on page 102.](#)

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

49 (Oracle database 9i and later). Set Optimizer Mode to CBO.

50 Run database statistics.

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

Postupgrade Tasks for Applications Configuration

51 Perform the tasks in [Chapter 18, "Postupgrade Tasks for Siebel Applications."](#)

52 Verify the function of interfaces in integrated applications.

53 Activate and deploy workflows.

- 54** 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 eBusiness Application Integration Volume I*.
- 55** 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

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

Process of Tuning Siebel Upgrade Performance

Upgrades: All Siebel upgrades.

Environments: Production test, production.

This process is optional.

This process is part of a road map. See [“Road Map for Performing a Siebel Database Upgrade” on page 67](#).

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 Oracle’s Siebel Expert Services.

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:

- **Technical Note 616.** This Technical Note describes strategies for minimizing production environment downtime during an upgrade. The steps in this process are intended primarily for use with the Baseline best practices described in Technical Note 616.
- [Chapter 19, “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 Upgrade Tuner. To obtain a Windows version of Siebel Server, contact your account manager or Oracle.

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 8, "Preparing an IBM DB2 Database for a Siebel Upgrade."
 - Chapter 9, "Preparing an Oracle Database for a Siebel Upgrade."
 - Chapter 10, "Preparing an MS SQL Server Database for a Siebel Upgrade."
- 5** "Preparing Siebel Tables and Views for Upgrade" on page 127.
- 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 130.
- 8** Perform the tasks in Chapter 11, "Preparing Siebel Application Data for Upgrade."

Upgrade the Target Database Schema (upgrep)

- 9** In the production test environment, run the Database Server Configuration Utilities:
 - "Preparing to Run the Siebel Database Server Configuration Utilities" on page 179.
 - "Running the Siebel Database Server Configuration Utilities Under Windows" on page 182.
 - "Running the Siebel Database Server Configuration Utilities Under UNIX" on page 183.Choose the following settings:
 - **Upgrade Options:** Upgrade Siebel Database Schema (upgrep)
 - **Environment Type:** Production
- 10** In the production test environment, run Siebel Upgrade Wizard. See "Starting the Siebel Upgrade Wizard" on page 184.
Note the time required to upgrade the database.
- 11** Review the upgrade logs for errors:
 - "Summarizing Siebel Log Files Using Logparse" on page 192.
 - "Reviewing Siebel Upgrade Log Files for Errors" on page 195.
- 12** If the upgrade contains errors that prevented completion or adversely affected performance, correct the errors and rerun the upgrade

Tune the Upgrade Files

- 13** Evaluate upgrade performance, particularly the time required to complete the upgrade.
- 14** 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 section to continue tuning upgrade performance.
 - If the time required to complete the upgrade is too long and you cannot tune further, contact Oracle's Siebel Expert Services to apply advanced tuning.
- 15** Carefully review target database and RDBMS server configuration. Adjust as needed to further improve upgrade performance.
- 16** Run Upgrade Tuner to tune the upgrade files:
 - **UNIX users.** ["Transferring UNIX Files for Use by Siebel Upgrade Tuner" on page 285.](#)
 - ["Starting and Stopping Siebel Upgrade Tuner" on page 278.](#)
 - ["Managing Parallel Threads Using Siebel Upgrade Tuner" on page 280.](#)
 - ["Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner" on page 282.](#)
 - **UNIX users.** ["Transferring UNIX Files for Use by Siebel Upgrade Tuner" on page 285.](#)

To roll back changes you have made in previous Upgrade Tuner sessions, see ["Rolling Back Siebel Upgrade Tuner Changes" on page 288.](#)

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.

- 17** 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.
- 18** In the production test environment, perform another test upgrade and evaluate upgrade performance.
- 19** 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.

- 20** In the production test environment, delete and remove the target database.
- 21** In the production test environment, restore the production-test final database from backup.

Process of Upgrading a Siebel Production Environment

Upgrades: All Siebel upgrades.

Environments: Production environment.

Prerequisites: 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 28](#). If you cannot do this, contact Oracle's Siebel Technical Support.

This process is part of a road map. See ["Road Map for Performing a Siebel Database Upgrade" on page 67](#).

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.

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

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 on Oracle's Siebel SupportWeb under Product Documentation > Maintenance Release Guides.

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.

- 1** 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.
- 2** 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.
- 3** Install language packs for your currently deployed languages and any new languages.
- 4** If you have customized the configuration of Enterprise components, such as Siebel Servers, you must manually enter the customizations in the upgraded environment. See *Going Live with Siebel Business Applications*.

Upgrade Third-Party Software

- 5** 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

- 6 If required, upgrade the RDBMS version. Refer to the vendor's documentation to perform the upgrade. For information on supported RDBMS systems, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

Pre-Upgrade Tasks for the Siebel Database

These steps apply to all database types.

- 7 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 8 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 9 Stop the Siebel Server and the Siebel Gateway Name Server.
- 10 Verify that there are no open database connections.
- 11 Perform the tasks in [Chapter 7, "Basic Database Preparations for a Siebel Upgrade."](#)

Pre-Upgrade Tasks for IBM DB2

- 12 Perform the tasks in [Chapter 8, "Preparing an IBM DB2 Database for a Siebel Upgrade."](#)
- 13 Execute runstats on the Siebel Database. This will improve upgrade performance.

Pre-Upgrade Tasks for Oracle

- 14 Perform the tasks in [Chapter 9, "Preparing an Oracle Database for a Siebel Upgrade."](#)
- 15 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

- 16 Perform the tasks in [Chapter 10, "Preparing an MS SQL Server Database for a Siebel Upgrade."](#)
- 17 Run MS SQL Server statistics. This will improve upgrade performance.

Pre-Upgrade Tasks for Application Data

18 Perform the tasks in Technical Note 521 on Oracle's Siebel SupportWeb. [Table 16](#) shows the applicability of the tasks in the Technical Note.

Table 18. Tasks in Technical Note 521

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

19 Perform the tasks in the Technical Notes and Alerts on Oracle's Siebel SupportWeb shown in [Table 17](#).

Table 19. Technical Notes and Alerts

Title	Environment
Technical Note 481	Development only

20 Perform the tasks in [Chapter 11, "Preparing Siebel Application Data for Upgrade."](#)

Pre-Upgrade Tasks for the UI

21 Copy UI files to the production environment:

- a** Custom SRF file. Verify that this is the most current version.
- b** Reports files.
- c** ["Copying UI Files to a New Siebel Environment" on page 214.](#)

Upgrade the Siebel Database (upgrep)

You upgrade the production database by using the SQL scripts in the production test environment. In the production test environment, you run the Database Configuration Utility but enter environment information for the production database. Then you start the Siebel Upgrade Wizard in the production test environment. The Siebel Upgrade Wizard uses the production database environment information and the SQL in the production test environment, including any changes you have made, to upgrade the production database.

22 Verify you have a current backup of the production environment database.

23 On the Siebel Server you used to upgrade the production test environment, create an ODBC to connect to the production environment database.

24 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 Server Configuration Utilities, if this file exists, no SQL commands are generated.

CAUTION: If this file does not exist, do not run the Database Server Configuration Utilities. It will overwrite the SQL files used to upgrade your production test database. Contact Oracle's Siebel Technical Support for guidance on proceeding with the upgrade.

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

25 Run the Database Server Configuration Utilities:

- ["Preparing to Run the Siebel Database Server Configuration Utilities" on page 179.](#)
- ["Running the Siebel Database Server Configuration Utilities Under Windows" on page 182.](#)
- ["Running the Siebel Database Server Configuration Utilities Under UNIX" on page 183.](#)

a Choose the following settings when you run the utility:

- **Upgrade Options:** Upgrade Siebel Database Schema (upgrep)
- **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.

26 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 Oracle's Siebel Technical Support.

27 Perform the production database upgrade. See ["Starting the Siebel Upgrade Wizard" on page 184.](#)

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.

28 Review the upgrade logs and resolve errors:

- ["Summarizing Siebel Log Files Using Logparse" on page 192.](#)
- ["Reviewing Siebel Upgrade Log Files for Errors" on page 195.](#)

29 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.

30 Multilingual deployments: Perform the steps in Technical Note 447 on Oracle's Siebel SupportWeb to import language-specific repository strings and seed data into the upgrade repositories.

31 ["Manually Archiving Siebel Upgrade Log Files" on page 197.](#)

Upgrade Custom Database Schema (upgphys)

You upgrade the production database by using the SQL scripts in the production test environment. In the production test environment, you run the Database Configuration Utility but enter environment information for the production database. Then you start the Siebel Upgrade Wizard in the production test environment. The Siebel Upgrade Wizard uses the production database environment information and the SQL in the production test environment, including any changes you have made, to upgrade the production database.

32 Verify you have a current backup of the production environment database.

33 On the Siebel Server you used to upgrade the production test environment, verify you have created an ODBC to connect to the production environment database.

34 Run the Database Server Configuration Utilities:

- ["Preparing to Run the Siebel Database Server Configuration Utilities" on page 179.](#)
- ["Running the Siebel Database Server Configuration Utilities Under Windows" on page 182.](#)
- ["Running the Siebel Database Server Configuration Utilities Under UNIX" on page 183.](#)

Choose the following settings:

- **Upgrade Options:** Upgrade Custom Database Schema (upgphys)
- **Environment Type:** Production

35 Review the upgrade logs and resolve errors:

- ["Summarizing Siebel Log Files Using Logparse" on page 192.](#)
- ["Reviewing Siebel Upgrade Log Files for Errors" on page 195.](#)

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

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

37 ["Manually Archiving Siebel Upgrade Log Files" on page 197.](#)

38 ["Installing New Siebel License Keys During an Upgrade" on page 189.](#)

39 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*.

- 40** Back up the upgraded production environment database.

Postupgrade Tasks for Database and File System

- 41** Perform the tasks in [Chapter 17, "Postupgrade Tasks for the Siebel Database and File System."](#)
- 42** 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.
- 43** If you exported data from interface tables before the upgrade, review the database and import the data as desired.
- 44** ["Upgrading to Siebel RC2 or AES Encryption" on page 102.](#)
- 45** Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide*.
- 46** (Oracle database 9i and later). Set Optimizer Mode to CBO.
- 47** Run database statistics.

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

Postupgrade Tasks for Applications Configuration

- 48** 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 106.](#)
- 49** Perform the tasks in [Chapter 18, "Postupgrade Tasks for Siebel Applications."](#)
- 50** Verify the function of interfaces in integrated applications.
- 51** Activate and deploy workflows.
- 52** 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 eBusiness Application Integration Volume I*.
- 53** 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

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

5

Siebel Database and UI Upgrade Planning

This area contains the following topics:

- [“Important Siebel Database Upgrade Planning Resources” on page 93](#)
- [“Best Practices for Doing Your Siebel Database Upgrade” on page 95](#)
- [“New Siebel Upgrade Features and Requirements” on page 98](#)
- [“About Upgrading Your RDBMS in the Siebel Environment” on page 98](#)
- [“About Siebel Multilingual Deployments” on page 99](#)
- [“About Siebel Unicode Support” on page 100](#)
- [“Upgrading to Siebel RC2 or AES Encryption” on page 102](#)
- [“About Siebel User Interface Changes” on page 102](#)
- [“Upgrade Planning for Siebel Web Template Files and Style Sheet” on page 103](#)
- [“About Database Sort Order in the Siebel Environment” on page 104](#)
- [“About Upgrading Siebel Access Control” on page 105](#)
- [“About the Siebel Party Model” on page 106](#)
- [“About Migrating Siebel HTML Attachments to Base Tables” on page 110](#)

Related Topics

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

Important Siebel Database Upgrade Planning Resources

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

This topic lists important publications and resources for performing an upgrade. Review these as part of the upgrade planning process.

Product Documentation

Siebel Product documentation is collectively called the Siebel Bookshelf. The Siebel Bookshelf is available on CD-ROM. It is also available on Oracle’s Siebel SupportWeb under Product Documentation.

Siebel Deployment Documentation Suite

The following publications are part of the *Siebel Deployment Documentation Suite*, which is part of the 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* for information about how to migrate customizations from the development environment to the production environment
- *Siebel System Administration Guide* for details on how to administer, maintain, and expand your Siebel Servers
- *Security Guide for Siebel Business Applications*
- *Performance Tuning Guide*
- *Configuring Siebel Business Applications* for 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 account representative or Oracle's Siebel Technical Support for information on ordering DMRs.

Oracle's Siebel SupportWeb

This is Oracle's Siebel products technical support Web site. It provides search engine access to the Siebel Bookshelf, Technical Notes, Alerts, troubleshooting information and other important information. SupportWeb is located at <https://ebusiness.siebel.com/supportweb/>.

- *Installation and Upgrade Portal Page*. This SupportWeb page lists important installation and upgrade publications, including Bookshelf titles, Technical Notes, and Alerts. Use this page as your entry point for obtaining installation and upgrade information. It is located on Oracle's Siebel SupportWeb > Installation and Upgrade.
- *System Requirements and Supported Platforms* on Siebel SupportWeb. This document is the definitive list of system requirements and supported third-party products. It is located on Oracle's Siebel SupportWeb at Product Documentation > System Requirements and Supported Platforms & Miscellaneous Documentation.
- *Release Notes*. *Release Notes contain* late-breaking information that the *Upgrade Guide* does not yet include. Release Notes regarding upgrade are located on Oracle's Siebel SupportWeb at Product Documentation > Release Notes > Siebel Business, Industry, Midmarket Edition, and CRM Applications 7.x and later > Categories > General > Upgrade.
- *Maintenance Release Guides*. Maintenance Release Guides contain important information about updates to applications in maintenance releases. Maintenance Release Guides are located on Oracle's Siebel SupportWeb at Product Documentation > Maintenance Release Guides.

- *Documentation Updates.* Typically, the Bookshelf is updated monthly. During the month, PDF updates are posted to Oracle's Siebel SupportWeb > Product Documentation > Documentation Updates.
- *Technical Notes and Alerts.* Technical notes and Alerts provide important information on specific upgrade issues. Technical Notes related to upgrade are located on Oracle's Siebel SupportWeb at Technical Notes > Upgrade. Alerts related to upgrade are located on Oracle's Siebel SupportWeb at Alerts > Upgrade.

Alert 1002 and Alert 1179 are master Alerts that list Technical Notes and Alerts particularly important to all upgrades. References to these Alerts and Technical Notes are integrated throughout the *Upgrade Guide*.
- *Troubleshooting Steps.* Troubleshooting Steps contain information about how to troubleshoot common error messages and unwanted behavior in Oracle's Siebel applications. Troubleshooting Steps are for upgrade list error messages found in upgrade logs and describe how to resolve them. Troubleshooting Steps are located on Oracle's Siebel SupportWeb at Troubleshooting Steps > Product Areas > Upgrade.
- *Siebel Weekly Content Notification Service.* This service notifies you weekly by email of important content changes on Oracle's Siebel SupportWeb. This includes new product documentation, technical notes, alerts, and troubleshooting steps. To subscribe, see the Siebel Weekly Content Notification banner on the SupportWeb search page.

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.

Expert Services

Oracle's Siebel Expert Services offers detailed implementation planning and technical consulting services. They also provide rapid response and resolution for critical technical issues affecting Siebel deployments.

Contact Oracle's Siebel Expert 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

Best Practices for Doing Your Siebel Database Upgrade

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

This section provides an overview of 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 *System Requirements and Supported Platforms* on Siebel SupportWeb 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 Maintenance Release Guides on Oracle's Siebel SupportWeb 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 fixpacks and also include new features. Because fixes to the upgrade process may exist in fixpacks 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 road map 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 *System Requirements and Supported Platforms* on Siebel SupportWeb, Release Notes, Maintenance Release Guide, and Alerts related to upgrades to verify your upgrade path. These documents are available on Oracle's Siebel SupportWeb at <https://ebusiness.siebel.com/supportweb/>.
- 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 135.](#)

New Siebel Upgrade Features and Requirements

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

Databases: All databases.

The following new upgrade features are introduced at 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.

- **Oracle 9i upgrades can be run in CBO or RBO mode.** Development environment, production test environment, and production environment upgrades on Oracle 9i can be run in CBO or RBO mode.

About Upgrading Your RDBMS in the Siebel Environment

Upgrades: All Siebel upgrades.

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 *System Requirements and Supported Platforms* on Siebel SupportWeb.

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 *System Requirements and Supported Platforms* on Siebel SupportWeb.

Oracle

- See Alert 1053 on Oracle's Siebel SupportWeb 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 250](#).
- Development, production test, and production upgrades on both Oracle 9i and 10g can 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 Multilingual Deployments

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

Databases: All databases.

The upgrep process imports repository strings and seed data in only the primary language. The primary language, also called the base language, is the language in which data is stored in the Siebel Database.

If you have multilingual deployments, you must import multilingual repository strings and seed data after performing the upgrep. This applies to development, production test, and production upgrades.

See Technical Note 447 on Oracle's Siebel SupportWeb for instructions on how to perform these imports.

See the *Siebel Installation Guide* for the operating system you are using for instructions on installing and configuring multiple language packs.

For a list of supported code pages and encoding strategies, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

Unicode Databases

In a Unicode-enabled database environment, you can install any available language pack. In a non-Unicode database environment, you must consider the correlation of the language packs you want to install and the characters supported by your RDBMS code page. For example, in a Western European code page RDBMS, you can only install Western European language packs such as English, French, Spanish, or German. In a Japanese code page RDBMS, you can only install Japanese or English language packs.

About Siebel Unicode Support

Upgrades: All Siebel upgrades.

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 Siebel Expert Services participation is mandatory. Contact Expert Services to perform a Unicode migration.

For a list of supported languages and code pages, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

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 Siebel Expert Services.

Contact 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, contact Oracle's Siebel Technical Support or Oracle's Siebel 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, contact Oracle's Siebel Technical Support or Oracle's Siebel Expert Services. See also "[Identifying IBM DB2 Long Columns for Truncation in a Siebel Upgrade](#)" on page 138.
- **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.

Contact Oracle's Siebel Technical Support or Oracle's Expert Services for instructions about how to modify upgrade scripts to handle custom tablespaces.
- **Third-party product integration.** Migration to Unicode may affect integration with third-party systems. For more information, contact Oracle's Siebel Technical Support or Oracle's Siebel Expert Services.
- **Handheld and wireless support.** Certain handheld and wireless application platforms do not support Unicode. For more information, contact Oracle's Siebel Technical Support or Oracle's Siebel 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.

Upgrading to Siebel RC2 or AES Encryption

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Development, production test, production.

Databases: All databases.

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*.

About Siebel User Interface Changes

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Development, production test, production.

Databases: All databases.

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 on 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, placement of views and controls were based on repository configuration and logic applied at run time. In Siebel 7.7, 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, you should plan a careful evaluation of the user interface after upgrade. The upgrade logging utility will list any user interface-related problems encountered during upgrade.

For more information on the user interface, see *Fundamentals*.

Upgrade Planning for Siebel Web Template Files and Style Sheet

Upgrades: All Siebel upgrades.

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 webtemp1 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: `\public\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* guide.

About Database Sort Order in the Siebel Environment

Upgrades: All Siebel upgrades.

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 *System Requirements and Supported Platforms* on Siebel SupportWeb 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 *System Requirements and Supported Platforms* on Siebel SupportWeb to verify sort order and code page requirements for your deployment.

About Upgrading Siebel Access Control

Upgrades from: Siebel 7.0.x.

Environments: Development, production test, production.

Databases: All databases.

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.

Technical Note 312 provides guidance and best practices for implementing access control. This Technical Note includes background information about the Access Group access control mechanism implemented in Siebel 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*.

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.

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 non-person 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 3 depicts the Party changes to the data model that occur during upgrades from Siebel 6.x to Siebel 7.x.

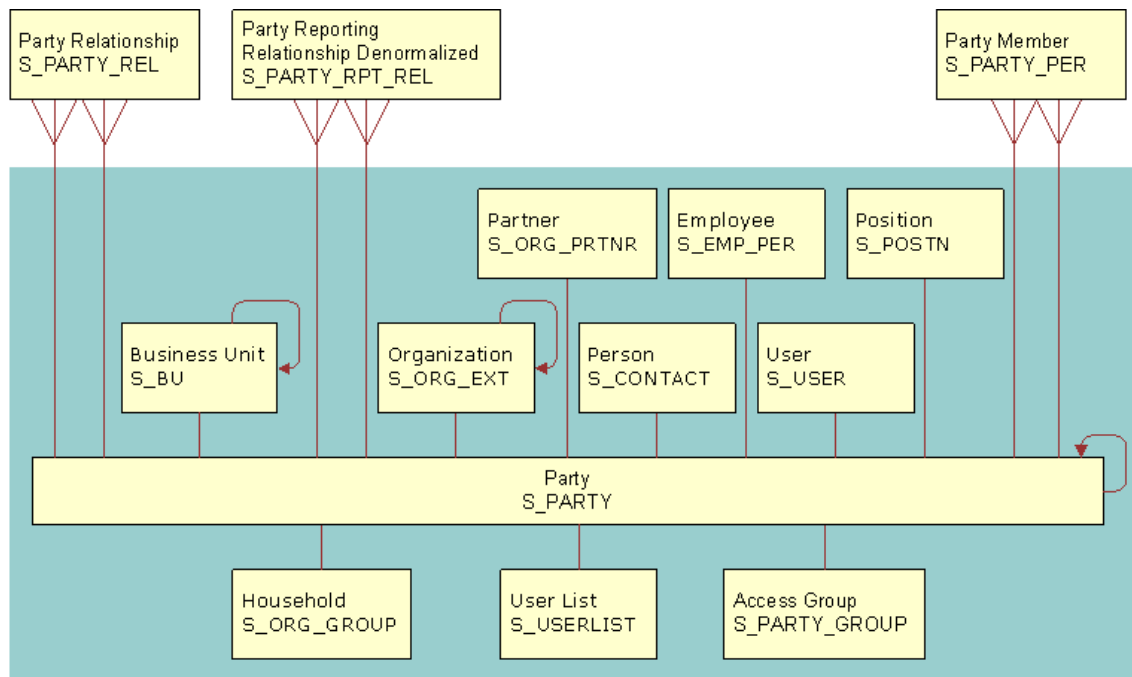


Figure 3. Party Model Changes

How the Party Model Is Implemented During 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 would initiate 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

At Siebel 7.0.x, the Party model changed the relationship between a household and the following entities for Siebel Financial Services:

- | | |
|----------------------------|-------------------|
| ■ Policy/Financial Account | ■ Service Request |
| ■ Activity | ■ Opportunity |
| ■ Claim | ■ 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 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 Oracle's Siebel Technical Support or Oracle's Siebel Expert Services for assistance in revising business components in the current release to support this model.

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 | ■ Household-to-contact |
| ■ Contact-to-household | ■ Household-to-household |
| ■ Contact-to-company | ■ Household-to-company |
| ■ Company-to-contact | ■ Company-to-company |
| ■ Company-to-household | |

About Migrating Siebel HTML Attachments to Base Tables

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Development, production test, production.

Databases: All databases.

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.

6

Application Planning for a Siebel Upgrade

This area contains the following topics:

- [“Determining Your Siebel 7.5.x Release Level” on page 111](#)
- [“Upgrade Planning for Migrating Siebel Address Data” on page 113](#)
- [“Upgrade Planning for Siebel Employee Relationship Management \(ERM\)” on page 114](#)
- [“Upgrade Planning for Siebel Marketing” on page 115](#)
- [“Upgrade Planning for Siebel Workflow Designer” on page 121](#)
- [“Upgrade Planning for Handheld Devices in the Siebel Environment” on page 122](#)
- [“Upgrade Planning for Resonate Central Dispatch in the Siebel Environment” on page 122](#)
- [“Upgrade Planning for Siebel String Translation” on page 123](#)
- [“Upgrade Planning for Siebel Configurator” on page 123](#)
- [“Upgrade Planning for Siebel Personalization” on page 124](#)
- [“Upgrade Planning for Siebel Pricer and Order Management” on page 124](#)
- [“Upgrade Planning: Additional Siebel Application Changes” on page 125](#)

Related Topics

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

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, it prompts 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 the *Maintenance Release Guide for Siebel 7.5.3* on Oracle's Siebel SupportWeb and install the update.

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 Oracle's Siebel SupportWeb and install the update.

- If some of the controls are present, but not all, contact Oracle's Siebel Technical Support.

Upgrade Planning for Migrating Siebel Address Data

Upgrades:

- From Siebel Financial Services 7.0.x to Siebel Industry Applications (SIA) 7.8.x
- From Siebel Business Applications 7.8.x to Siebel SIA 7.8.x
- From Siebel Financial Services 6.2.1 on IBM z/OS platforms to Siebel SIA 7.8.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 upgrep, 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 77](#). 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 6.x, 7.0.x, & 7.5.x.

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 6.x, 7.0.x, & 7.5.x.

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 Siebel Analytics.** The Marketing Server is obsolete. Its functions have been migrated to Siebel Analytics and are implemented on the Siebel Analytics 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 4](#).

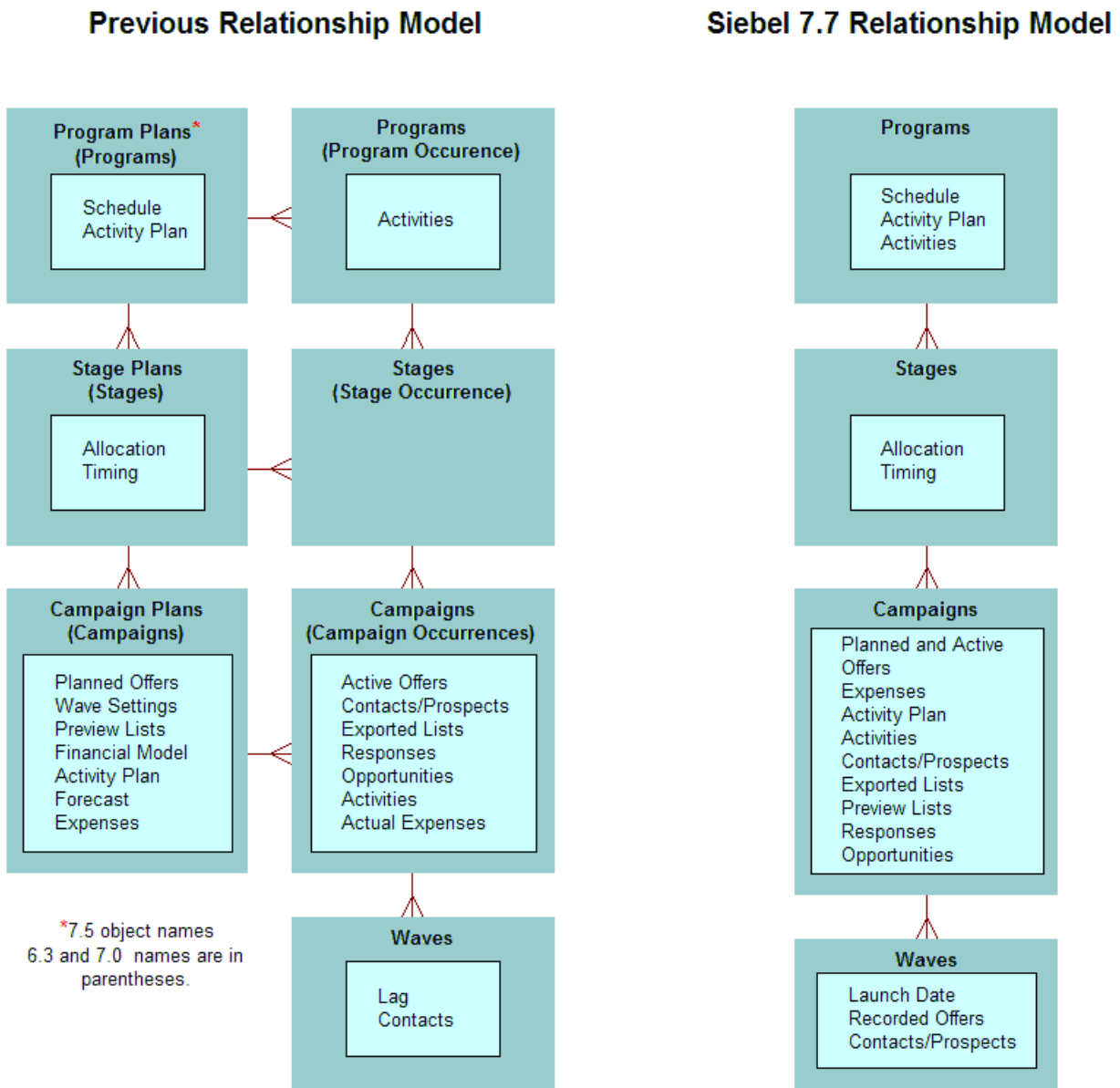


Figure 4. Relationship Model for Siebel Marketing

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 Siebel Analytics

The upgrade obsoletes the Data Dictionary Manager and Marketing Server 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 Siebel Analytics Web 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 20 on page 118 summarizes Siebel Database changes that apply to Siebel Marketing.

Table 20. Summary of Database Changes for Siebel Marketing at 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 20. Summary of Database Changes for Siebel Marketing at Siebel 7.7

Upgrade Step	Description
Updates S_SRC for Stages	Updates CAMP_TYPE_CD = "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 21 on page 120 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 21. Siebel Marketing Data Migration at 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

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

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 Designer Administration 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.

Upgrade Planning for Handheld Devices in the Siebel Environment

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

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 *System Requirements and Supported Platforms* on Siebel SupportWeb.

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 *System Requirements and Supported Platforms* on Siebel SupportWeb.

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 6.x, 7.0.x, & 7.5.x.

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 *System Requirements and Supported Platforms* on Siebel SupportWeb.

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 6.x, 7.0.x, & 7.5.x.

Siebel 7.7 introduced a new method of generating data-migration SQL. When the SQL commands for the upgrade are created, they contain generic values or placeholders. A set of files maps these values to language-specific strings. The upgrade process substitutes these strings in the SQL commands. The translated strings typically are those associated with seed data such as menu items in LOVs.

The strings assigned to values are based on those used in previous Siebel releases. If you have made changes to string mappings in a previous release, contact Oracle's Siebel Technical Support or Oracle's Siebel Expert Services for guidance on managing string mapping during the upgrade.

CAUTION: If you have modified string mappings in a previous release, you must obtain Oracle assistance before upgrading. If you do not do so, data may be lost or corrupted during upgrade.

Upgrade Planning for Siebel Configurator

Upgrades from: Siebel 6.x, 7.0.x, & 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 Technical Note 500 on Oracle's Siebel SupportWeb.

Browsers that will host high interactivity applications must meet certain requirements. These are described in the *System Requirements and Supported Platforms* on Siebel SupportWeb. 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 6.x, 7.0.x, & 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, & 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 22](#).

Table 22. 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 <ul style="list-style-type: none"> ■ Aggregate factor ■ Bundle factor ■ Single factor ■ Matrix factor ■ Script based 	<p>Not automatically upgraded. Must be redesigned and reimplemented as pricing procedures.</p> <p>Bundle factor definitions are upgraded to aggregate discounts and aggregate discount sequences.</p>

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 Technical Note 639 on Oracle's Siebel SupportWeb.

Upgrade Planning: Additional Siebel Application Changes

Upgrades: All Siebel upgrades.

Table 23 displays additional application changes. Review these changes and make needed revisions before beginning the upgrade.

Table 23. Additional Application Changes

Applications or Views Affected	Applies to Upgrades from	Description
Siebel Service	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	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	7.5 and earlier	In Siebel 7.7, the HI Gantt Chart Applet is no longer used. It is replaced by the Calendar GanttChart AX Applet.

Technical Note 511 on Oracle's Siebel SupportWeb 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.

7

Basic Database Preparations for a Siebel Upgrade

This area contains the following topics:

- [“Verifying Siebel Database Connectivity” on page 127](#)
- [“Preparing Siebel Tables and Views for Upgrade” on page 127](#)
- [“Preparing Siebel Custom Indexes for Upgrade” on page 128](#)
- [“Exporting Siebel Interface Table Data” on page 129](#)
- [“Archiving Unneeded Siebel Repositories” on page 129](#)
- [“Preserving Siebel Dock Objects and Visibility Rules” on page 130](#)
- [“Securing AIX Memory Allocation Segment Space for the Siebel Database” on page 130](#)

Related Topics

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

Verifying Siebel Database Connectivity

Upgrades: All Siebel upgrades.

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, contact Oracle’s Siebel Technical Support.

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: All Siebel upgrades.

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.”](#)

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: All Siebel upgrades.

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.”](#)

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, contact Oracle’s Oracle’s Siebel Technical Support.
- **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 drops 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: All Siebel upgrades.

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.”](#)

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: All Siebel upgrades.

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: All Siebel upgrades.

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."](#)

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's Siebel Technical Support or Oracle's Siebel Expert Services.

Securing AIX Memory Allocation Segment Space for the Siebel Database

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

Databases: All databases. (Exception: this topic does not apply to MS 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 LOADPUBLIC@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.

8

Preparing an IBM DB2 Database for a Siebel Upgrade

This area contains the following topics:

- [“Verifying the IBM DB2 Client for a Siebel Upgrade” on page 131](#)
- [“Verifying IBM DB2 Sort Order for a Siebel Upgrade” on page 131](#)
- [“Setting IBM DB2 Parameters for a Siebel Upgrade” on page 132](#)
- [“Verifying IBM DB2 Permissions for a Siebel Upgrade” on page 134](#)
- [“Verifying IBM DB2 Instance Owner Permissions for a Siebel Upgrade” on page 134](#)
- [“Creating IBM DB2 Temporary Tablespaces and Bufferpools for a Siebel Upgrade” on page 135](#)
- [“Analyzing IBM DB2 Custom Tablespace Requirements for a Siebel Upgrade” on page 135](#)
- [“Verifying the DB2 Application Development Client for a Siebel Upgrade” on page 138](#)
- [“Identifying IBM DB2 Long Columns for Truncation in a Siebel Upgrade” on page 138](#)

Related Topics

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

Verifying the IBM DB2 Client for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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 RDBMS.

Verifying IBM DB2 Sort Order for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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 Oracle’s Siebel Technical Services for further analysis.

Related Topic

[“About Database Sort Order in the Siebel Environment” on page 104](#)

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
```

Setting IBM DB2 Parameters for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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 below 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 below.
- 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 24](#) provides guidelines for setting configuration parameters specifically to optimize upgrade performance. Set these parameters for each DB2 instance.

Table 24. 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 25 on page 133 .

DB2 Database Configuration Parameters

The Upgrade Setting column in [Table 25](#) provides guidelines for setting configuration parameters specifically to optimize upgrade performance. Set these parameters for each DB2 instance.

Table 25. 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

Table 25. DB2 Database Configuration Parameters

Parameter	Explanation	Upgrade Setting
CHNGPGS_THRESH	Changed pages threshold	5
LOGRETAIN	Sequential or circular log files	NO Setting this parameter to YES means that log files are archived. You must periodically move or archive the logs to prevent the file system containing the log files from filling up.
LOGFILSIZ	Log file size (4 KB)	Development environments: 8000-16000
SOFTMAX	Triggers bufferpool flushing	50

Verifying IBM DB2 Permissions for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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 `sqllib/function/routine/sqlproc` has write permission for the group:

```
ls -ld sqllib/function/routine/sqlproc
```

- 3 To authorize group write permission, enter the following command:

```
chmod g+w sqllib/function/routine/sqlproc
```

Verifying IBM DB2 Instance Owner Permissions for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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: All Siebel upgrades.

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: All Siebel upgrades.

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 /V 32K_TABLESPACE  
/Q 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.ct1*, 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 ssia /F d:/sea77/dbsrvr/db2udb/ddl.ct1 /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 provided below:

```
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 `ddlmove`.

`ddlmove` 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 `ddlmove`, submit the following arguments:

```
ddlmove /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: All Siebel upgrades.

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 26](#) lists the required DB2 Application Development Client components.

Table 26. 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 6.x, 7.0.x, & 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 below, 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_TBLO` = tableowner of the source database

`SRC_TBLO_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:\sea7xx\dbsrvr` (Windows). Edit the value "CHANGE_ME".

`SIEBEL_ROOT` = The directory where you installed the Siebel Server. For example, `C:\sea7xx\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.

9

Preparing an Oracle Database for a Siebel Upgrade

This area contains the following topics:

- [“Verifying Oracle Database Sort Order for a Siebel Upgrade” on page 141](#)
- [“Verifying Oracle Database Configuration for a Siebel Upgrade” on page 142](#)
- [“Verifying Oracle Database Parameters for Multiple CPUs in a Siebel Upgrade” on page 143](#)
- [“Verifying the Oracle Database ODBC Definition for a Siebel Upgrade” on page 143](#)
- [“Setting Oracle Database Optimizer Mode for a Siebel Upgrade” on page 145](#)

Related Topics

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

Verifying Oracle Database Sort Order for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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.

Related Topic

[“About Database Sort Order in the Siebel Environment” on page 104](#)

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.

Verifying Oracle Database Configuration for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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:

- 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 below.
- After the upgrade, reset the configuration parameters to the values listed in *Siebel Installation Guide* for the operating system you are using.

Before upgrading an Oracle database, complete the following tasks:

- Read *System Requirements and Supported Platforms* on Siebel SupportWeb 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.
- 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 init.ora parameters that the DBA should review are:

- `sort_area_retained` size
- `log_buffers`
- **UNDO_MANAGEMENT (Oracle 9i 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 848 on Oracle's Siebel SupportWeb.
- **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.

Verifying Oracle Database Parameters for Multiple CPUs in a Siebel Upgrade

Upgrades: All Siebel upgrades.

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: All Siebel upgrades.

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. If the development environment database is Oracle 9i, you must edit the ODBC definition after it has been created.

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.

Setting Oracle Database Optimizer Mode for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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."](#)

Oracle 9i

You can run the development upgrade, production test upgrade, and the production upgrade in either Rule Based Optimizer (RBO) mode or in Cost Based Optimizer (CBO) mode.

You need not run all environment upgrades in the same mode. For example, you can run the development environment upgrade in RBO mode and the production test environment upgrade in CBO mode.

Oracle 10g

Run all environment upgrades in CBO mode.

Setting the Optimizer Mode

Set the optimizer mode (OPTIMIZER_MODE) in init.ora as follows:

- For RBO, set the value to RULE.
- For CBO, set the value to CHOOSE.

Do not use the other OPTIMIZER_MODE settings.

Related Topics

Siebel Technical Note 582 on supported optimizer modes for Oracle databases.

Siebel Alert 1011 on supported optimizer modes for upgrading Oracle databases.

10 Preparing an MS SQL Server Database for a Siebel Upgrade

This area contains the following topics:

- [“Verifying MS SQL Server Sort Order for a Siebel Upgrade” on page 147](#)
- [“Setting MS SQL Server Temporary Space Size for a Siebel Upgrade” on page 148](#)
- [“Setting MS SQL Server Configuration Parameters for a Siebel Upgrade” on page 148](#)
- [“Rebuilding MS SQL Server Clustered Indexes for a Siebel Upgrade” on page 150](#)

Related Topics

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

Verifying MS SQL Server Sort Order for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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 MS 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 MS 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.

For more information on database sort order, see [“About Database Sort Order in the Siebel Environment” on page 104.](#)

To verify that your database was created using a binary collation sequence

- 1 In the Query Analyzer window, enter the following command:

```
sp_helpsort
```

This command provides a sort order description.

- 2 Review the sort order description to verify binary sort order; for example:

```
Latin1_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.

Setting MS SQL Server Temporary Space Size for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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 MS 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 MS SQL Server Configuration Parameters for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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 MSSQL 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 below.
- After the upgrade, reset the configuration parameters to the values listed in *Siebel Installation Guide* for the operating system you are using.

[Table 27](#) lists upgrade settings for MS 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 27. Microsoft SQL Configuration Parameters

Parameter	Setting/Comment
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 MS SQL Server Clustered Indexes for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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 MS SQL Server create index command with drop_existing clause to rebuild large tables with high fillfactor (60%-70%).

11 Preparing Siebel Application Data for Upgrade

This area contains the following topics:

- [“Preparing Siebel Workflow Processes for Upgrade” on page 151](#)
- [“Identifying Siebel Seed Data Customizations” on page 152](#)
- [“Preparing Siebel Customized Seed Data for Upgrade” on page 158](#)
- [“Migrating Siebel Household Data” on page 160](#)
- [“Setting Up Campaign Status Values for Siebel Marketing” on page 162](#)
- [“Preserving Siebel Marketing Segment Descriptions” on page 164](#)
- [“Preparing Siebel Mobile User Data for Upgrade” on page 164](#)
- [“Setting the Value of Siebel S_SRC_PAYMENT.TYPE_CD for Upgrade” on page 166](#)
- [“Preparing Siebel Address Data for Upgrade” on page 166](#)
- [“Migrating Siebel Address Data from Custom Extension Columns” on page 167](#)
- [“Preparing Siebel Customizable Product Data for Upgrade” on page 174](#)

Related Topics

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

Preparing Siebel Workflow Processes for Upgrade

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

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 1016 on Oracle's Siebel SupportWeb.

- 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 Designer Administration Guide*.

Upgrades from all Releases

Perform the following task in all environments.

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.

Identifying Siebel Seed Data Customizations

Upgrades from: Siebel 7.8.1 to 7.8.2.

Environments: Development, production test, production.

The upgrade from Siebel 7.8.1 to 7.8.2 deletes existing seed data in S_VOD and related tables and inserts new seed data. This affects customer and order management applications.

This topic describes how to run a report that identifies your customizations to Siebel 7.8.1 seed data in these tables. After the upgrade, use this report to apply these customizations to Siebel 7.8.2 seed data as desired.

Perform the following tasks in the order shown to identify seed data customizations:

- Creating the report view

- Running the report
- Reviewing the report
- Dropping the report view

Prerequisites: You must have Microsoft Excel, and you must be able to enter a stand-alone ^ (circumflex) from the keyboard. To enter a circumflex if it is not on your keyboard, press and hold the left Alt key; then enter 94 on the numeric keypad, and release the Alt key.

Creating the Report View

A script for creating the report view is included in Siebel 7.8.2.

To create the report view

- 1 Navigate to the following directory:

Windows: *SIEBEL_ROOT\bin*

UNIX: *\$SIEBEL_ROOT/bin*

- 2 Type the following command:

- Windows:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /c REM
/O OutputFileLocation\rpt1_create_view.txt /h /separator /
/L LogFileLocation\rpt1_create_view.log
ScriptLocation\create_view_vod_ver_view.sql
```

- UNIX:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /c REM
/O OutputFileLocation/rpt1_create_view.txt /h /separator /
/L LogFileLocation/rpt1_create_view.log
ScriptLocation/create_view_vod_ver_view.sql
```

- z/OS:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /c REM
/O OutputFileLocation/rpt1_create_view.txt /q SchemaQualifier /h /separator /
/L LogFileLocation/rpt1_create_view.log
ScriptLocation/create_view_vod_ver_view.sql
```

where:

Tableowner = Siebel Database tableowner account name

Password = Tableowner password

ODBCDataSource = Data source name for the Siebel Database

OutputFileLocation = Location of the output file:

- Windows: *SIEBEL_ROOT\log\rpt1_create_view.txt*

❑ UNIX: `$SIEBEL_ROOT/log/rpt1_create_view.txt`

SchemaQualifier = DB2 schema qualifier name

LogFileLocation = Location of the log file:

❑ Windows: `SIEBEL_ROOT\log\rpt1_create_view.log`

❑ UNIX: `$SIEBEL_ROOT/log/rpt1_create_view.log`

ScriptLocation = Location of the script

❑ Windows: `DBSRVR_ROOT\database_platform\create_view_vod_ver_view.sql`

❑ UNIX: `DBSRVR_ROOT/database_platform/create_view_vod_ver_view.sql`

where *database_platform* is the database-type directory name. UNIX example:
`/usr/siebel/sea7xx/dbsrvr/DB2UDB/create_view_vod_ver_view.sql`

3 Review the `rpt1_create_view.log` and `rpt1_create_view.txt` files for errors.

Running the Report

A script for running the report is included in Siebel 7.8.2. The report file is created in UTF-8 format.

To run the report

1 Navigate to the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

2 Type the following command:

■ Windows:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /c REM  
/O OutputFileLocation\rpt1_run.txt /h /separator /  
/L LogFileLocation\rpt1_run.log ScriptLocation\vod_diff_rpt.sql
```

■ UNIX:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /c REM  
/O OutputFileLocation/rpt1_run.txt /h /separator /  
/L LogFileLocation/rpt1_run.log ScriptLocation/vod_diff_rpt.sql
```

■ z/OS:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /c REM  
/O OutputFileLocation/rpt1_run.txt /q SchemaQualifier /h /separator /  
/L LogFileLocation/rpt1_run.log ScriptLocation/vod_diff_rpt.sql
```

where:

Tableowner = Siebel Database tableowner account name

Password = Tableowner password

ODBCDataSource = Data source name for the Siebel Database

OutputFileLocation = Location of the output file:

□ Windows: *SIEBEL_ROOT*\log\rpt1_run.txt

□ UNIX: \$*SIEBEL_ROOT*/log/rpt1_run.txt

SchemaQualifier = DB2 schema qualifier name

LogFileLocation = Location of the log file:

□ Windows: *SIEBEL_ROOT*\log\rpt1_run.log

□ UNIX: \$*SIEBEL_ROOT*/log/rpt1_run.log

ScriptLocation = Location of the script:

□ Windows: *DBSRVR_ROOT*\database_platform\vod_diff_rpt.sql

□ UNIX: *DBSRVR_ROOT*/database_platform/vod_diff_rpt.sql

where *database_platform* is the database-type directory name. UNIX example:
/usr/siebel/sea7xx/dbsrvr/DB2UDB/vod_diff_rpt.sql

3 Review the rpt1_run_.log file for errors.

Reviewing the Report

To review the report, import rpt1_run.txt into Microsoft Excel. The text is ^ (circumflex) delimited.

To review the report

- 1** Start Microsoft Excel and open rpt1_run.txt.
- 2** In the wizard or the dialog boxes that display on how to format the file, make the following selections:
 - Original data type: Delimited.
 - Start import at row: 1.
 - File origin: Accept the displayed code page.

- Delimiters: Remove the check mark from Tab and place a check mark in Other. In the adjacent box, enter ^ (circumflex).
If the circumflex is not on your keyboard, press and hold the left Alt key and enter 94 on the numeric keypad. Then release the Alt key.
- Treat consecutive delimiters as one: Verify this check box does not contain a check mark.
- Text qualifier: Accept the listed default.
- Column data format: General.

The imported text file displays in column format. The first row lists the column heads.

3 Save the file in .xls format.

4 Increase column widths as required to display all the text in each field.

5 Review the report.

The report lists customizations you have made to seed data since installing Siebel 7.8.1. Use [Table 28](#) to review the report.

Table 28. Columns in the Seed Data Customizations Report

Column	Explanation
Type	The seed data object type. Object types map to UI applet names and include: <ul style="list-style-type: none">■ Variable Definition■ Variable Map Mode■ Child Variable Map■ Variable Source■ Variable Srce Param■ Signal Action■ Signal Property■ Action Parameter
Name	Corresponds to record names in the UI
Action	<ul style="list-style-type: none">■ New. You added this object after Siebel 7.8.1 was installed.■ Modified. You modified this object after Siebel 7.8.1 was installed.■ Deleted. You deleted this object after Siebel 7.8.1 was installed.

Table 28. Columns in the Seed Data Customizations Report

Column	Explanation
Roottype	These columns list the family tree of the object. The columns provide both the object type and object name.
Rootname	
childtype	The Roottype and Rootname are the top of the family tree. The lowest level of the tree is greatgrandchildtype and greatgrandchildname.
childname	
grandchildtype	To determine the position in the family tree of an object, do the following: <ol style="list-style-type: none"> 1 Note the object's type in the type column. If this is the same as the roottype, the object is at the top of the tree. 2 If type and roottype are not equal, then check childtype, grandchildtype and so on until you find the type. Use this method to construct the family tree of an object.
grandchildname	
greatgrandchildtype	
greatgrandchildname	
Field_ <i>n</i>	Where <i>n</i> is the field number. These columns list the field names in the object's record.
Field_ <i>n</i> _OldVal	The Siebel 7.8.1 value for this field. If this value blank, it may mean that you have created a new object.
Field_ <i>n</i> _NewVal	The current value of this field. If blank, it may mean that you deleted this object. By looking at the Action field and then comparing the OldVal and NewVal fields you can determine how an object was customized.
Y	Ignore this column, if present.

Dropping the Report View

A script for dropping the report database view is included in Siebel 7.8.2.

To drop the report view

- 1 Navigate to the following directory:

Windows: *SIEBEL_ROOT*\bin

UNIX: *\$SIEBEL_ROOT*/bin

- 2 Type the following command:

- Windows:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /c REM
/O OutputFileLocation\rpt1_drop_view.txt /h /separator /
/L LogFileLocation\rpt1_drop_view.log ScriptLocation\drop_view_vod_ver_view.sql
```

■ UNIX:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /c REM  
/O OutputFileLocation/rpt1_drop_view.txt /h /separator /  
/L LogFileLocation/rpt1_drop_view.log ScriptLocation/drop_view_vod_ver_view.sql
```

■ z/OS:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /c REM  
/O OutputFileLocation/rpt1_drop_view.txt /q SchemaQualifier /h /separator /  
/L LogFileLocation/rpt1_drop_view.log ScriptLocation/drop_view_vod_ver_view.sql
```

where:

Tableowner = Siebel Database tableowner account name

Password = Tableowner password

ODBCDataSource = Data source name for the Siebel Database

OutputFileLocation = Location of the output file:

□ Windows: *SIEBEL_ROOT*\log\rpt1_drop_view.txt

□ UNIX: \$*SIEBEL_ROOT*/log/rpt1_drop_view.txt

SchemaQualifier = DB2 schema qualifier name

LogFileLocation = Location of the log file:

□ Windows: *SIEBEL_ROOT*\log\rpt1_drop_view.log

□ UNIX: \$*SIEBEL_ROOT*/log/rpt1_drop_view.log

ScriptLocation = Location of the script:

□ Windows: *DBSRVR_ROOT*\database_platform\drop_view_vod_ver_view.sql

□ UNIX: *DBSRVR_ROOT*/database_platform/drop_view_vod_ver_view.sql

where *database_platform* is the database-type directory name. UNIX example:
/usr/siebel/sea7xx/dbsrvr/DB2UDB/drop_view_vod_ver_view.sql

- 3 Review the rpt1_drop_view.log and rpt1_drop_view.txt files for errors.

Preparing Siebel Customized Seed Data for Upgrade

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

If you modify seed data through the GUI or through EIM, the upgrade process preserves the modifications.

CAUTION: Modified seed data are not upgraded, even if the upgraded seed data are required for an application to run normally.

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:

- a** 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.
- b** 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
- c** 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 more information on log files, see ["About the Siebel Database Upgrade Logs" on page 191](#).

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.

If you are upgrading from Siebel 7.8.1 to 7.8.2, you can run a report to identify seed data customizations in S_VOD and related tables. For more information, see ["Identifying Siebel Seed Data Customizations" on page 152](#).

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 MSSQL	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')

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.

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. For information on the Party model and how it affects Siebel Financial Services, see [“About the Siebel Party Model” on page 106](#).

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_Mig_populate.txt
/L LogFileLocation/HH_Mig_populate.log ScriptLocation/HH_Mig_populate.sql /v
```

where:

Tableowner = Tableowner

Password = Tableowner password

ODBCDataSource = Data source of the database

OutputFileLocation = Location of the output file:

□ Windows: *SIEBEL_ROOT\log\HH_Mig_populate.txt*

□ UNIX: *\$SIEBEL_ROOT/log/HH_Mig_populate.txt*

LogFileLocation = Location of the log file :

□ Windows: *SIEBEL_ROOT\log\HH_Mig_populate.log*

□ UNIX: *\$SIEBEL_ROOT/log/HH_Mig_populate.log*

ScriptLocation = Location of the script:

□ Windows: *DBSRVR_ROOT\database_platform\HH_Mig_populate.sql*

■ UNIX: `DBSRVR_ROOT/database_platform/HH_Mig_populate.sql`

Windows example:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /a /c REM /separator / /O
C:\sea7xx\siebsrvr\Log\HH_Mig_populate.txt /L
C:\sea7xx\siebsrvr\Log\HH_Mig_populate.log
C:\sea7xx\dbsrvr\DB2UDB\HH_Mig_populate.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_Mig_populate.sql script.
- Before doing the upgrade, contact Oracle's Siebel Technical Support or Oracle's Siebel Expert Services and 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.

Setting Up Campaign Status Values for Siebel Marketing

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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 29 lists the standard Status values for each object.

Table 29. 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 30 lists search specifications for locating records containing Status values. Siebel 7.5 terminology is used for the object names.

Table 30. 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'
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

Preserving Siebel Marketing Segment Descriptions

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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.

Preparing Siebel Mobile User Data for Upgrade

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

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: `SIEBEL_ROOT\docking\MOBILEUSERNAME`.

UNIX: `$SIEBEL_ROOT/docking/MOBILEUSERNAME`.

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.

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) 7.8.x
- From Siebel Business Applications 7.8.x to Siebel SIA 7.8.x
- From Siebel Financial Services 6.2.1 on IBM z/OS platforms to Siebel SIA 7.8.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.

Migrating Siebel Address Data from Custom Extension Columns

Upgrades:

- From Siebel Financial Services 7.0.x to Siebel Industry Applications (SIA) 7.8.x
- From Siebel Business Applications 7.8.x to Siebel SIA 7.8.x
- From Siebel Financial Services 6.2.1 on IBM z/OS platforms to Siebel SIA 7.8.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: Development and production test.

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 Oracle’s Siebel Technical Support or Oracle’s Siebel Expert Services if you need assistance.

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 *nnn*]

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.ctl` 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
```

```
, ZIPCODE
, ADDR_NAME
, 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
)
select
ROW_ID
, CREATED
, CREATED_BY
, LAST_UPD
, LAST_UPD_BY
, MODIFICATION_NUM
, CONFLICT_ID
, DISA_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)}
    else 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.

Preparing Siebel Customizable Product Data for Upgrade

Upgrades: All Siebel upgrades. Exception: does not apply to upgrades from Siebel 6.x.

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.

12 Upgrading the Siebel Database

This area contains the following topics:

- [“Renaming the Siebel Repository” on page 177](#)
- [“Changing the Siebel Database Server Configuration Utilities Language” on page 178](#)
- [“Preparing to Run the Siebel Database Server Configuration Utilities” on page 179](#)
- [“Running the Siebel Database Server Configuration Utilities Under Windows” on page 182](#)
- [“Running the Siebel Database Server Configuration Utilities Under UNIX” on page 183](#)
- [“Starting the Siebel Upgrade Wizard” on page 184](#)
- [“Stopping the Siebel Upgrade Wizard” on page 186](#)
- [“Regenerating SQL Files for a Siebel Upgrade” on page 187](#)
- [“Identifying and Dropping Obsolete Indexes for a Siebel Upgrade” on page 187](#)
- [“Preparing for a No-Development-Environment Siebel Upgrade” on page 188](#)
- [“Installing New Siebel License Keys During an Upgrade” on page 189](#)

Related Topics

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

Renaming the Siebel Repository

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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). Do not append the repository name with additional characters, because this results 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 Prior 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 Server Configuration Utilities Language

Upgrades: All Siebel upgrades.

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."](#)

The Database Server Configuration Utilities launches in the language selected when you ran the Siebel Enterprise System Installer. You can change the language in which the Utility runs, if desired, from the language chosen during installation.

To change the Database Server 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 Server 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.

For more information about the Database Server Configuration Utilities, see ["About the Siebel Database Server Configuration Utilities" on page 32](#).

Preparing to Run the Siebel Database Server Configuration Utilities

Upgrades: All Siebel upgrades.

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 Server Configuration Utilities. Collect this information and verify it before running the utility.

The Database Server Configuration Utilities requests information about the upgrade process you want to perform. It then adds this information to a master upgrade file and calls an SQL generator. The SQL generator uses the information to create or populate SQL files.

The Upgrade Wizard runs after the Database Server Configuration Utilities exits. The wizard executes the SQL files against the Siebel Database.

Under Windows, the Database Server Configuration Utilities is a wizard that displays a series of windows. You enter the requested information in each window. Most of the windows contain a help button. When you click Help, an explanation of what to enter in the wizard displays.

Under UNIX, the Database Server Configuration Utilities displays as a set of screens in a shell command window. You enter the requested information at the prompt.

[Table 31 on page 180](#) lists the information that you must enter in the Database Server Configuration Utilities.

[Table 32 on page 181](#) lists additional information the utility requires when you perform a production environment or production upgrade.

For more information on the Database Server Configuration Utilities, see [“About the Siebel Database Server Configuration Utilities” on page 32](#).

Table 31. Information Required for the Database Server Configuration Utilities

Screen Name	Required Information
Gateway Name Server Address	Windows only. The Siebel Gateway Name Server machine name and 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:\sea7xx\dbsrvr.
RDBMS Platform	Choose the RDBMS type.
Siebel Database Operation	Choose Upgrade Database. The other 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.
Upgrade Options	Choose the upgrade mode, for example Prepare for Production upgrade or upgrep.
Siebel Industry Application	This displays if you are upgrading from a Siebel Industry Solution or Siebel Industry Application. Choose the application you are upgrading from. 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.
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	Account name and password for the Siebel Administrator of the Siebel Database you are upgrading.
Database Table Owner	Account name and password for the Siebel Database table owner.

Table 31. Information Required for the Database Server Configuration Utilities

Screen Name	Required Information
Index Table Space Name	Oracle and IBM DB2 only. Index tablespace name and tablespace name (4-KB table's space name for DB2).
16-KB Table Space Name	DB2 only. The 16-KB and 32-KB tablespace names.
Database Server OS	Choose the RDBMS server operating system type.
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 19, "Tuning the Siebel Upgrade Files."</p>
Log Output Directory	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.

Additional Information Required for Production Upgrades

When you perform a production test environment or production environment upgrade, the additional information shown in [Table 32](#) 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 32. Additional Information Required for Production Upgrades

Screen Name	Required Information
ODBC Data Source Name for Development Database	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.
Database User Name for Development Database	Account name and password of the Siebel Administrator of the Siebel Database in the development environment.

Table 32. Additional Information Required for Production Upgrades

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.

Running the Siebel Database Server Configuration Utilities Under Windows

Upgrades: All Siebel upgrades.

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 Server Configuration Utilities to upgrade the Siebel Database. The Utility 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.

For more information on the Database Server Configuration Utilities, see [“About the Siebel Database Server Configuration Utilities” on page 32.](#)

Prerequisite: Collect the information that the Database Server Configuration Utilities requires. See [“Preparing to Run the Siebel Database Server Configuration Utilities” on page 179.](#)

To run the Database Server Configuration Utilities under Windows

- 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 Select Start > Programs > Siebel Enterprise Servers 7.x > Configure DB Server.

The first window of the Database Server Configuration Utilities appears.

- 3 Enter the information requested in each window and click Next.

To return to previous windows, click Previous.

- 4 When the “runnow” window displays, do the following:

- **Uppreg mode:** Answer No. Do not start the Upgrade Wizard.
- **Upphys mode:** Answer Yes to start the Upgrade Wizard.

- **Prepare for Production mode:** Answer Yes to start the Upgrade Wizard.

Just before displaying the runnow window, the Database Server Configuration Utilities calls the SQL generator to create or populate SQL scripts.

To manually start the Upgrade Wizard, see [“Starting the Siebel Upgrade Wizard” on page 184](#).

Running the Siebel Database Server Configuration Utilities Under UNIX

Upgrades: All Siebel upgrades.

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 Server Configuration Utilities to upgrade the Siebel Database. The Utility 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.

For more information on the Database Server Configuration Utilities, see [“About the Siebel Database Server Configuration Utilities” on page 32](#).

Prerequisite: Collect the information that the Database Server Configuration Utilities requires. See [“Preparing to Run the Siebel Database Server Configuration Utilities” on page 179](#).

To run the Database Server Configuration Utilities 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.

The path must end in siebsrvr. For example, /usr/siebel/sea7xx/siebsrvr

- 3 Source the environment variables:

Korn shell (do not use a Bourne 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/sea7xx/siebsrvr.

- **LANGUAGE.** This is the language in which the Database Server Configuration Utilities runs. The value of this variable is a language identifier string. For example, `enu` is the identifier string for English.

If either the `SIEBEL_ROOT` or `LANGUAGE` value is not set or is incorrect, you must correct them before proceeding.

- 5 Start the Database Server Configuration Utilities script in either a C shell or Korn shell:

```
./bin/dbsrvr_config.ksh
```

- 6 Enter the information requested in each screen. Click ENTER to proceed to the next screen.

After you have entered all the requested information, the utility displays a screen that lists the values you entered. If you need to make changes, exit and rerun the utility.

- 7 When prompted to start the Upgrade Wizard, do the following:

- **Uppreg mode:** Answer No. Do not start the Upgrade Wizard.
- **Uppphys mode:** Enter Y to start the Upgrade Wizard.
- **Prepare for Production mode:** Answer Yes to start the Upgrade Wizard.

Just before displaying the prompt, the Database Server Configuration Utilities calls the SQL generator to create or populate SQL scripts.

To manually start the Upgrade Wizard, see [“Starting the Siebel Upgrade Wizard” on page 184](#).

Starting the Siebel Upgrade Wizard

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

Platforms: All platforms.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

The Upgrade Wizard calls drivers that execute SQL scripts against the Siebel Database. These scripts were created or populated when you ran the Database Server Configuration Utilities.

When the Upgrade Wizard encounters errors that prevent continuing the upgrade, it writes the errors to the log files and stops the upgrade. After you correct the problems causing the errors, you can restart the Upgrade Wizard. In most cases, the wizard resumes from where it stopped.

For more information on the Upgrade Wizard, see [“About the Siebel Upgrade Wizard and Driver Files” on page 36](#).

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 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`

To manually start the Upgrade Wizard

- 1 Navigate to the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Enter the following command:

Windows: `siebugp /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`

UNIX: `srvrupgwiz /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`

where:

`_UPGRADEOPTION_ENVIRONMENT_VERSION` is the portion of the upgrade configuration file name that lists upgrade mode, upgrade environment, and the Siebel release from which you are upgrading. The file is located in `SIEBEL_ROOT\bin` (UNIX: `$SIEBEL_ROOT/bin`).

For example, to start a development upgrade from Siebel 7.0.4 under Windows, enter the following command:

`siebugp /m master_upgrep_dev_704.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.

Related Topics

[“Stopping the Siebel Upgrade Wizard” on page 186](#)

Stopping the Siebel Upgrade Wizard

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

Platforms: All platforms.

The Upgrade Wizard sends SQL commands to the RDBMS. These commands make schema changes and migrate data. If you stop the Upgrade Wizard and restart it, the Upgrade Wizard resumes the upgrade after the last upgrade step it successfully completed.

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 hung. 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 hung, contact Oracle’s Siebel Technical Support.

Stopping the Upgrade Wizard can have varying effects on the RDBMS. Before restarting the Upgrade Wizard, review the RDBMS logs. Run SQL commands as needed to resolve errors found in the RDBMS logs.

Stopping the Upgrade Wizard Under Windows

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

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 already-issued SQL commands.

Related Topics

[“Starting the Siebel Upgrade Wizard” on page 184](#)

Regenerating SQL Files for a Siebel Upgrade

Upgrades: All Siebel upgrades.

Environments: Development and production test.

If you enter incorrect information in the Database Server Configuration Utilities, the SQL files it generates will be incorrect and must not be used. You must run the utility 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 generates SQL files, it creates 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, contact Oracle’s Siebel Technical Support for assistance.

- 2 Navigate to `DBSRVR_ROOT\common` (UNIX: `DBSRVR_ROOT/common`) and locate the file `sqlgen.usg`.
This is the lock file that prevents the utility from regenerating the SQL files.

- 3 Delete the `sqlgen.usg` file.

- 4 Rerun the Database Server Configuration Utilities.

Identifying and Dropping Obsolete Indexes for a Siebel Upgrade

Upgrades: All Siebel upgrades.

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, it does 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 statement that drops the index. The utility places this SQL statement in a file.

This file is not executed when you run the Upgrade Wizard. You must manually review the file. If it contains indexes you want to drop, you must copy the corresponding SQL statements to another SQL file. This file is executed by the Upgrade Wizard after you run the Database Server Configuration Utilities in upgrep mode.

When the Upgrade Wizard runs, 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 Upgrade Wizard after performing the database upgrep.

Preparing for a No-Development-Environment Siebel Upgrade

Upgrades: All Siebel upgrades.

Environments: Production test, production.

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.ct1` 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: All Siebel upgrades.

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.

13 Reviewing the Siebel Upgrade Log Files

This area contains the following topics:

- [“About the Siebel Database Upgrade Logs” on page 191](#)
- [“Summarizing Siebel Log Files Using Logparse” on page 192](#)
- [“Reviewing Siebel Upgrade Log Files for Errors” on page 195](#)
- [“Manually Archiving Siebel Upgrade Log Files” on page 197](#)

Related Topics

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

About the Siebel Database Upgrade Logs

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

The Upgrade Wizard produces all log files for a given process in one directory. This includes summary.txt and summary.html files:

Windows: *SIEBEL_ROOT\log\PROCESS*

UNIX: *\$SIEBEL_ROOT/log/PROCESS*

where *PROCESS* defaults to *UPGRADEOPTION_ENVIRONMENT_VERSION*.

For example, all log files for a Siebel Database schema upgrade of a development environment from Siebel 7.0.4 appear in the following locations:

Windows: *SIEBEL_ROOT\log\upgprep_dev_704*

UNIX: *\$SIEBEL_ROOT/log/upgprep_dev_704*

All log files for the prepare for production upgrade process appear in the following directory:

Windows: *SIEBEL_ROOT\log\prepare_for_production_upgrade*

UNIX: *\$SIEBEL_ROOT/log/prepare_for_production_upgrade*

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

Each subdirectory in the log file directory contains the following files and subdirectories:

- **Output.** Directory containing process log files.

- **State.** Directory containing the state.log file.

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.** Directory that appears after execution of the log parse utility, containing source files for summary.html.
- **Summary.html.** Browser-based file created by the log parse utility, summarizing the log files in the output directory.
- **Summary.txt.** Text file created by the log parse utility, summarizing the log files in the output directory.

NOTE: For UNIX, 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 log parse utility during production upgrades and used by the Siebel Upgrade Tuner to parallelize table creation and index creation and 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 log parse 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 log parse 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.)

Summarizing Siebel Log Files Using Logparse

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Use the log parse utility to create summary.txt and summary.html, a browser-based summary of all files in the log output directory. Use the summary file to review database process log files, verify parameters, review errors, and identify costly SQL and DDL steps.

To execute the log parse 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
```

Table 33. Command Line Flags for Logparse

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:\sea7xx\siebsrvr, then <i>install_dir</i> is C:\sea7xx. Enclose the path in quotes.	Required
/g	LANGUAGE_CODE	The language code for the log files, for example enu.	Required
/r	PROCESS	Name of the schema process for which you want a summary generated, for example upgrep_dev_601.	Default value=ALL (creates summary files for all processes in the output directory)
/l	LOGPARSE_FILENAME	Name of the log parse log file.	Default=logparse.log
/n	MAX_NUMBER_SQL	Maximum number of longest-running SQLs to display in the summary.	Default value=10
/t	THRESHOLD_TIME	Threshold time for longest running SQLs, 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

Windows example (language is English):

```
logparse /s C:\sea7xx /g enu /r upgrep_dev_704 /l logparse.log /n 10 /t 00:00:10
/e 5
```

The summary.html file automatically opens for your review if you executed the log parse utility for a single database process (that is, if you used the command line flag */r PROCESS*, where *PROCESS=UPGRADEOPTION_ENVIRONMENT_VERSION*).

If you executed the log parse utility for all processes in the output directory (that is, if you used the command line flag `/r ALL`), you must manually open the summary file that you want to review.

Log files appear in your primary language, but `summary.html` appears only in English (enu).

To review the `summary.html` or `summary.txt` file

- 1 If the summary file is not already open, open `summary.html` from the following directory:

Windows: `SIEBEL_ROOT\log\PROCESS`

UNIX: `$SIEBEL_ROOT/log/PROCESS`,

Where:

`PROCESS` is the database process you want to review.

UNIX: If a browser is not installed on the machine you are using, open `summary.txt` from `$SIEBEL_ROOT/log/PROCESS`.

- 2 Review the parameters to make sure they are correct.

- If you are reviewing `summary.html`, select the Parameters link to review your upgrade parameters.

When you are finished, click Back to return to the summary screen.

- If you are reviewing `summary.txt`, scroll down to review your upgrade parameters.

Passwords are encrypted.

- 3 Review the step details for errors.

- If you are reviewing `summary.html`, select the Steps/Errors link, then scroll down the page to review the step details for errors.

- ☐ If no errors occurred for a step, then the step detail displays `Errors:None`.

- ☐ If more than five errors occurred for a step, the number of errors found appears as a link. Click the link to review all errors for that step.

- If you are reviewing `summary.txt`, scroll down to review the step details for errors.

If more than five errors occurred for a step, you must manually review the log files.

`Summary.html` and `summary.txt` only identify errors encountered by the ODBC driver for the database. `Summary.html` does not identify errors from Siebel utilities. To determine if errors were encountered by Siebel utilities, review the log files for these utilities.

- 4 For every error found, determine whether it is an acceptable error by following the procedure [“Reviewing Siebel Upgrade Log Files for Errors” on page 195](#).

If you are reviewing `summary.html`, click Back to return to the summary screen.

- 5 Identify SQL steps that have the greatest performance impact by selecting the Performance Information for SQL link in `summary.html` or by scrolling down to the following sections in `summary.txt`:

- Longest-running queries (based on the criteria you defined using the `/t` command line option)

- Queries that returned zero rows (based on the criteria you defined using the /n command line option)
- If a group of steps are executed in parallel, only the duration time of the longest step is added to the total process time. This means that the sum of durations from parallel-step logs will be less than the duration shown in summary.html (summary.txt).

If you are reviewing summary.html, click Back to return to the summary screen.

6 Review the performance impact caused by table and index creation.

- If you are reviewing summary.html, select the Performance Information for DDL link to review performance information.

This page may take a while to load due to its large size.

- If you are reviewing summary.txt, scroll down to review performance information.

If no information is available, then the process did not execute DDL.

7 Resolve any errors that occurred and, if necessary, restart the upgrade.

Reviewing Siebel Upgrade Log Files for Errors

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Do not proceed with the upgrade until unacceptable errors have been corrected. If you cannot correct the error, contact Oracle’s Siebel Technical Support or Professional Services to report the error in detail.

The log files may include errors that are expected and benign. Before you continue with the upgrade, you must compare any error messages found in the log files to a list of acceptable error messages.

If you encountered an error during the upgrade, review log files named upgwiz.log (Windows) or srvrupgwiz1.log (UNIX). The log file is located in the following directory:

Windows: `SIEBEL_ROOT\log\PROCESS\output`

UNIX: `$SIEBEL_ROOT/log/PROCESS/output`

The name of the log file increments for subsequent log files that are created when the upgrade wizard encounters a problem and the user runs the upgrade wizard again.

Review the end of the log file for details about the latest failure. If the step that failed was not a native SQL step (which would be listed in the log file), then it occurred as part of an external utility for which you need to review a corresponding log file, identified by the /L parameter.

Some Log Files Can Be Ignored

If the upgrade completed successfully, there are several log files that you can safely ignore:

- Windows: `sw_cfg_XXX.log` and `siebel.log`
- UNIX: `srvrupgwiz_*.log` and `siebel_*.log`. For example, `srvrupgwiz1_01.log`, and `srvrupgwiz1_02.log`
- Any other log file that existed before the start of the upgrade

CAUTION: UNIX Only: The log file `srvrupgwiz1_01.log` is a different file than `srvrupgwiz1.log`. Do not ignore log files named `srvrupgwiz1.log`, `srvrupgwiz2.log` and so on.

Related Topics

[“Summarizing Siebel Log Files Using Logparse” on page 192](#)

Troubleshooting Steps 19: How To Troubleshoot Messages Generated During the Repository Merge in Siebel 7, located on Oracle’s Siebel SupportWeb

Troubleshooting Steps 21: How to troubleshoot messages generated while running the Repository Upgrade in Siebel v7

Troubleshooting Steps 28: Errors.rtf -- Siebel version 7.5.3

Reviewing the Log Files

To manually review the log files for unacceptable errors

- 1 Review the `state1.log` file to see at what step the upgrade failed. This step can be traced back to the driver file. The `state1.log` file is located in the following directory:

Windows: `SIEBEL_ROOT\log\PROCESS\state`

UNIX: `$SIEBEL_ROOT/log/PROCESS/state`

- 2 Print the `errors` file. This file is located in the installation subdirectory for your database platform:

Windows: `DBSRVR_ROOT\DATABASE_PLATFORM\errors.rtf` or `errors.htm`

UNIX: `DBSRVR_ROOT/DATABASE_PLATFORM/errors.txt`

- 3 Sort the files in the following directory by date.

Windows: `SIEBEL_ROOT\log\PROCESS\output`

UNIX: `$SIEBEL_ROOT/log/PROCESS/output`

- 4 Open each log file, starting with the earliest, and search for errors.

Log files are identified by the `.log` extension. Errors are either tagged with the word “error” or enclosed in square brackets “[...]”.

You must start with the earliest log file to shorten your research time if you find unacceptable errors in an early log file.

- 5 For each error found, compare the error description against the list of acceptable errors documented in the `errors` file.

The log files generated by the repository upgrade wizard (for example `srvrupgwiz1.log`) appear in the `errors` file as `upgwiz1.log`, `upgwiz2.log`, incrementing for additional log files.

- If you find the error in the `errors` file, it is acceptable and no action is required. Continue to review the errors found in the log file.
- If an error appears multiple times in a log file, but only one occurrence of that error appears in the `errors` file, all errors of that type are acceptable and no action is required. Continue to review the errors found in the log file.
- If a log file is not listed in the `errors` file, there are no acceptable error messages for that log file. You must correct the condition that caused the error before you rerun the Upgrade Wizard.
- If you find an error that is not listed in the `errors` file, it is unacceptable. You must correct the condition that caused the error before you rerun the Upgrade Wizard.

On Oracle's Siebel SupportWeb, navigate to Troubleshooting Steps > Product Areas > Upgrade, and choose Troubleshooting Steps 21. 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's Siebel Technical Support. *Do not proceed with the upgrade.*

- 6 Repeat the previous step for each log file.

Although other errors are rarely encountered, this review is critical. Certain errors, such as a failure to create indexes, may result in performance problems or anomalous behavior in Siebel Business Applications.

Manually Archiving Siebel Upgrade Log Files

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

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 (9) 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 `siebel_log_archive` environment variable to 20—for example, to retain twenty (20) log files.

14 Performing the Siebel Repository Merge

This area contains the following topics:

- [“Configuring Siebel Repository Objects to Inherit Upgrade Behavior” on page 199](#)
- [“Configuring Siebel Tools for the Repository Merge” on page 200](#)
- [“Performing a Siebel Repository Merge” on page 201](#)
- [“Determining if a Siebel Repository Merge was Successful” on page 205](#)
- [“Generating Siebel EIM Temporary Columns” on page 207](#)
- [“Reviewing Siebel Repository Object Property Conflicts” on page 208](#)
- [“Regenerating the Siebel Repository Definition Files” on page 209](#)
- [“Moving the Siebel Repository Files” on page 211](#)
- [“Setting Label Alignment for Siebel Text Fields” on page 212](#)
- [“Copying UI Files to a New Siebel Environment” on page 214](#)
- [“Deleting Unneeded Siebel Repository Files” on page 215](#)
- [“Migrating Siebel Repository Objects to the Standard UI” on page 215](#)
- [“Running the Siebel Postmerge Utilities” on page 220](#)

Related Topics

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

Configuring Siebel Repository Objects to Inherit Upgrade Behavior

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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.

For more information, see [“About Inheriting Upgrade Behavior in a Siebel Upgrade” on page 49.](#)

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.

Configuring Siebel Tools for the Repository Merge

Upgrades: All Siebel upgrades.

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.”](#)

The topic applies to the Siebel Tools installation you will use to perform the repository merge.

If you are upgrading on Oracle 9i and want to run the development upgrade in Rule Based Optimization (RBO) mode, you must change the value of the SqlStyle parameter in the Siebel Tools configuration file. After the upgrade, you must reset the parameter to the default.

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.

Performing a Siebel Repository Merge

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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 `tools.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
EnableToolsConstrain	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:

`/editseeddata`

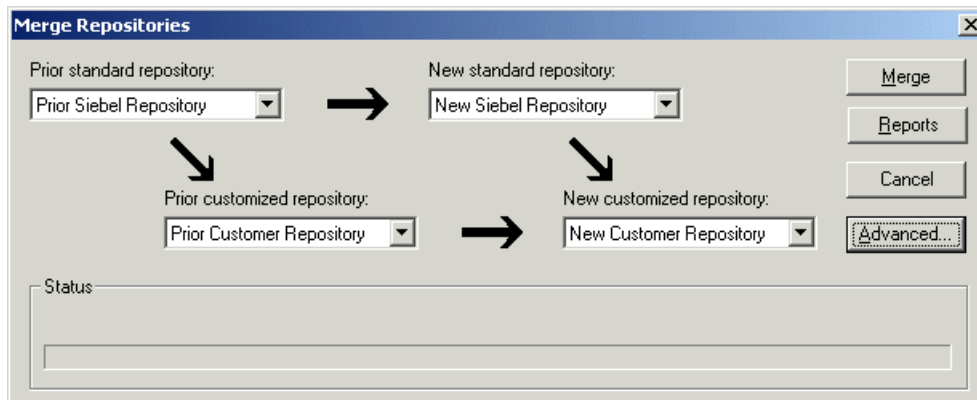
- 9 Navigate to View > Options > Language Settings.
- 10 Verify that the language mode setting is set as desired.

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 described below.

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.

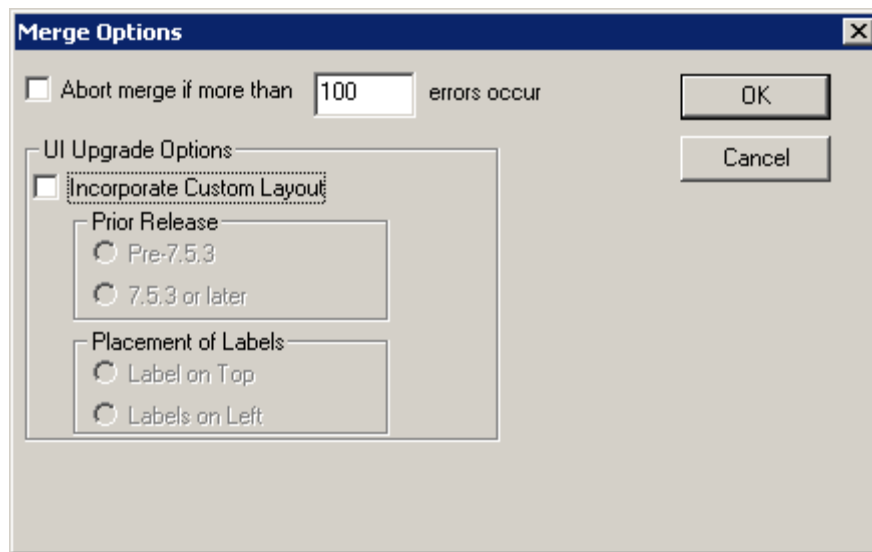
The Merge Options dialog box appears.

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.

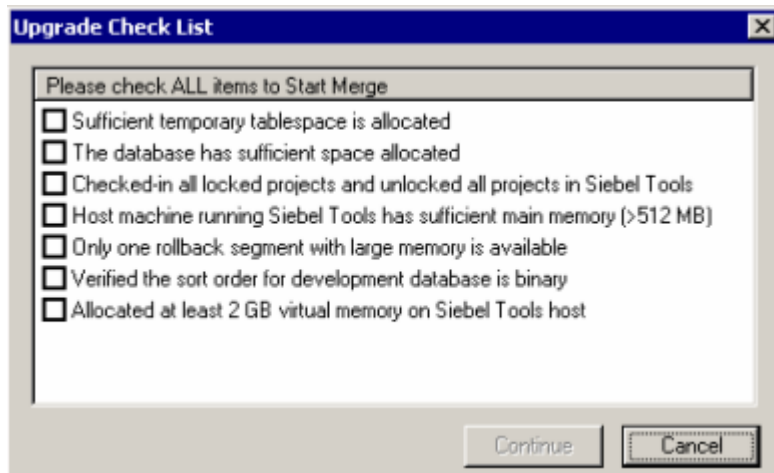


To continue, click OK.

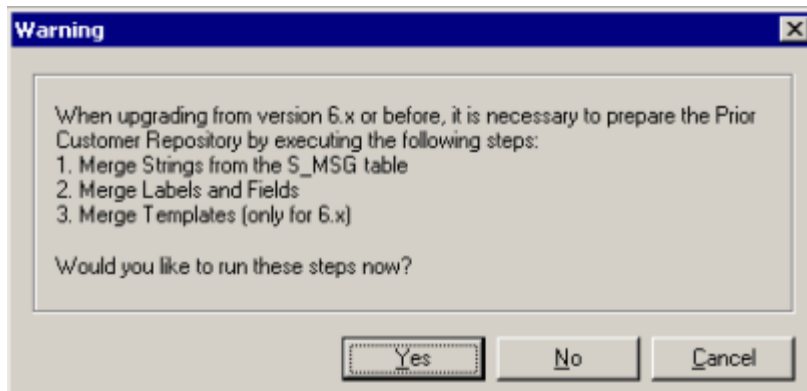
The Upgrade Check List dialog box appears.

- 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.

Determining if a Siebel Repository Merge was Successful

Upgrades: All Siebel upgrades.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform a Siebel Database Upgrade."](#)

The repository 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 Applet - Product marked as 'Upgrade Anc'
```

- **Unacceptable errors.** All other types of merge errors are unacceptable errors and mean that the merge was not successful.

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.

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 ! (exclamation point).
- 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 successful.
 - If the log contains unacceptable errors, the merge has failed.
- 10 If the merge contains unacceptable errors, go to Oracle's Siebel SupportWeb and search for "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, use the directions in "Troubleshooting Steps 19" to contact Oracle's Siebel Technical Support.

Related Topics

Troubleshooting Steps 19: How To Troubleshoot Messages Generated During the Repository Merge in Siebel 7, located on Oracle's Siebel SupportWeb

Generating Siebel EIM Temporary Columns

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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 upgEimCo1.

This indicates that the EIM temporary columns were created successfully.

Reviewing Siebel Repository Object Property Conflicts

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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.

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 objects'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 ["Determining if a Siebel Repository Merge was Successful" on page 205](#).

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

Regenerating the Siebel Repository Definition Files

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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:

```
ddlDict /u DatabaseOwner /p Password /c "ODBCDataSource" /d TableOwner /f  
DBSRVR_ROOT\DatabasePlatform\schema.ddl /e y /a y /l SiebelLogDir\sch_dict.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 logs for errors. If the log indicates there are errors, contact Oracle's Siebel Technical Support.

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:

```
repimexp /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 logs for errors. If the log indicates there are errors, contact Oracle's Siebel Technical Support.

Moving the Siebel Repository Files

Upgrades: All Siebel upgrades.

Environments: Production test, production.

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 209](#).

To move the repository files

- 1 In the development environment, navigate to the following directory:

Windows: *DBSRVR_ROOT\Platform*

UNIX: *DBSRVR_ROOT/Platform*

Where

Platform 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\Platform*

UNIX: *DBSRVR_ROOT/Platform*

- 4 Make a copy of these files and store them in a safe location.

Setting Label Alignment for Siebel Text Fields

Upgrades from: Siebel 7.x using ICL.

Environments: Development environment only.

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:
`\public\<lang>\files\main.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:
`\public\lang\files\printmain.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.

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.

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 was copied to the webtemp1 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 webtemp1 directory. For more information, see ["About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option" on page 55.](#)

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 237.](#)

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 (*install_dir*) in the development environment, locate the style sheet files:

install_dir\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.

Deleting Unneeded Siebel Repository Files

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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

Migrating Siebel Repository Objects to the Standard UI

Upgrades: All Siebel upgrades. Perform this task if you selected Incorporate Custom Layout (ICL) on your previous upgrade.

Environments: Development environment only.

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 inactivated 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 upgreq 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 list below 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 inactivated standard UI objects in the New Customer Repository and replaced them with ICL objects from the Prior Customer Repository. Migration to the standard UI inactivates and deletes ICL objects and activates their counterpart standard UI objects.
- **Standard UI objects.** These are the UI objects that were inactivated 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.

For a full description of how ICL works during an upgrade, see ["About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option" on page 55](#).

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 218 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 logs the changes to the repository in the following log file:

SIEBEL_ROOT\log\iclmigration.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: Your Siebel Tools installation must be Siebel 7.8 or later.

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:

- Your Siebel Tools installation must be Siebel 7.8 or later.
- 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 upgrep 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.

Running the Siebel Postmerge Utilities

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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\reputility.log

Where

SIEBEL_TOOLS_INSTALL_DIR *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.

15 Reviewing the Siebel User Interface

This area contains the following topics:

- [“Troubleshooting Postmerge Siebel UI Problems” on page 223](#)
- [“Verifying Siebel Business Address Applet Configuration” on page 225](#)
- [“Reviewing Siebel Grid-Based Applets” on page 225](#)
- [“Reviewing Siebel UI Navigation” on page 226](#)
- [“Reviewing Siebel Multi-Value Group \(MVG\) Shuttle Applets” on page 227](#)
- [“Revising Siebel UI Rich Text Controls” on page 229](#)
- [“Reviewing New Siebel UI Aggregate Categories” on page 231](#)
- [“Revising Siebel Visibility Filters to Display Correctly” on page 231](#)
- [“Assigning a Category and Type to Siebel Chart Views” on page 232](#)
- [“Assigning a Category and Type to Siebel Explorer Views” on page 233](#)
- [“Setting Up Navigation to Inaccessible Siebel Detail Views” on page 234](#)
- [“Eliminating Obsolete Siebel UI Fields” on page 235](#)
- [“Reviewing Siebel UI Objects Affected by Incorporate Custom Layout” on page 235](#)
- [“Reviewing Required Fields in the Siebel UI” on page 236](#)
- [“Assigning an Item Identifier to Siebel Web Template Items” on page 237](#)

Related Topics

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

Troubleshooting Postmerge Siebel UI Problems

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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 124](#).
- 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 Topic

Troubleshooting Steps 19: How to troubleshoot messages generated during the Repository Merge in Siebel 7, located on Oracle's Siebel SupportWeb

Verifying Siebel Business Address Applet Configuration

Upgrades from: Siebel 7.0.x. 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 6.x, 7.0.x, & 7.5.x.

Environments: Development environment only.

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. For more information on Incorporate Custom Layouts (ICL), see ["About the Siebel Incorporate Custom Layout \(ICL\) Upgrade Option"](#) on page 55.

The reputility.log, 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.
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.

Reviewing Siebel UI Navigation

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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.

Reviewing Siebel Multi-Value Group (MVG) Shuttle Applets

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Development environment only.

Reputility log section: “Multivalue 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>

Revising Siebel UI Rich Text Controls

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Development environment only.

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

Reviewing New Siebel UI Aggregate Categories

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Development environment only.

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.

Revising Siebel Visibility Filters to Display Correctly

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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.

Assigning a Category and Type to Siebel Chart Views

Upgrades from: Siebel 6.x, 7.0.x, & 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.

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 reputility 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.

Assigning a Category and Type to Siebel Explorer Views

Upgrades from: Siebel 6.x, 7.0.x, & 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.

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.

Setting Up Navigation to Inaccessible Siebel Detail Views

Upgrades: All Siebel upgrades. Exception: this topic does not apply to 7.x upgrades that used the Incorporate Custom Layout (ICL) feature.

Environments: Development environment only.

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\reputility.log`

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.

Eliminating Obsolete Siebel UI Fields

Upgrades: All Siebel upgrades. This topic applies only to upgrades using the Incorporate Custom Layout (ICL) feature.

Environments: Development environment only.

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.

Reviewing Siebel UI Objects Affected by Incorporate Custom Layout

Upgrades from: Siebel 7.0.x, 7.5.x, 7.7.x, 7.8.x. This topic applies only to upgrades using the Incorporate Custom Layout (ICL) feature.

Environments: Development environment only.

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.

Reviewing Required Fields in the Siebel UI

Upgrades from: Siebel 7.0.x, 7.5.x, 7.7.x, 7.8.x. This topic applies only to upgrades using the Incorporate Custom Layout (ICL) feature.

Environments: Development environment only.

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: All Siebel upgrades.

Environments: Development environment only.

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 Technical Note 400 on Oracle's Siebel SupportWeb and resolve duplicate item identifiers.
- 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.

16 Siebel Postmerge Development Tasks

This area contains the following topics:

- [“Reviewing Objects Deleted from the Siebel Repository” on page 239](#)
- [“Reviewing Obsolete Objects in the Siebel Repository” on page 240](#)
- [“Migrating Custom Siebel Workflows” on page 241](#)
- [“Upgrading to the Siebel Symbolic String Model” on page 242](#)
- [“Dropping IBM DB2 8-KB Tablespaces and Buffers After a Siebel Merge” on page 242](#)
- [“Updating Siebel Enterprise Application Integration \(EAI\)” on page 243](#)

Related Topics

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

Reviewing Objects Deleted from the Siebel Repository

Upgrades: All Siebel upgrades.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

Deleted objects are those that were in the repository when it was installed but that you have since deleted. The merge retrieves these objects from the Prior Siebel Repository and adds them to the New Customer Repository. Typically, adding these objects does not cause a problem with your upgraded configuration.

Prerequisite: The repository merge must have been successful. See [“Determining if a Siebel Repository Merge was Successful” on page 205.](#)

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.

Reviewing Obsolete Objects in the Siebel Repository

Upgrades: All Siebel upgrades.

Environments: Development environment only.

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 ["Determining if a Siebel Repository Merge was Successful"](#) on page 205.

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.

Migrating Custom Siebel Workflows

Upgrades from: Siebel 7.7.x & 7.8.x.

Environments: Development environment only.

Custom workflows are not migrated to the New Customer Repository during the upgrep. You must use Siebel Tools to migrate these workflows to the New Customer Repository.

The migration process has the following characteristics:

- All workflows that are in the Prior Customer Repository but not the New Customer Repository are copied to the New Customer Repository. Version 0 workflows are not copied.
- If you have modified a version 0 workflow, change the name of this workflow in the Prior Customer Repository before starting the migration process. This causes the migration process to copy the workflow to the New Customer Repository.
- You can repeat the migration process as needed. You can also exit the migration process before completion and run it again later. You can perform or repeat the migration process at any time after the upgrep and before the upgphys.

To migrate custom workflows

- 1 Start Siebel Tools.

The Siebel Tools version must be 7.8.x or later.

- 2 Select Tools > Upgrade > Migrate Custom Workflow Processes.

The Select Repositories dialog box displays.

- 3 Make the following selections in the dialog box, then click OK:

Field Name	Make This Selection
Prior customized repository	<ul style="list-style-type: none"> ■ Prior Customer Repository. ■ Or select the repository that you are using as the Prior Customer Repository.
New customized repository	<ul style="list-style-type: none"> ■ New Customer Repository. ■ Or select the repository you are using as the New Customer Repository.

The Migrate Workflow Process dialog displays. This dialog displays the steps in the migration. An OK in the first column means the workflow was migrated successfully.

- 4 To stop the migration before it completes, click Cancel.

To restart the migration later, begin at Step 1 above.

5 Review the log file for errors.

The log file is located at `Tools_install_dir\log\upgwf.log`.

To locate logged errors, search the log for the word "error."

Troubleshooting

If the custom workflow migration process encounters an error and cannot migrate a workflow, it prints the word "error" instead of "OK" in the first column of the Migrate Workflow Processes dialog and in the log file.

In most cases, errors are caused by a problem with the workflow record in the Prior Customer Repository. Examine the record, resolve the error, and rerun the workflow migration.

Upgrading to the Siebel Symbolic String Model

Upgrades from: Siebel 6.x, 7.0.x, & 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*.

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

Upgrades: All Siebel upgrades.

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: All Siebel upgrades.

Environments: Development environment only.

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, inactivate 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.

17

Postupgrade Tasks for the Siebel Database and File System

This area contains the following topics:

- [“Reapplying Schema Customizations in the Siebel Database” on page 245](#)
- [“Checking for Inactivated EIM Table Columns in the Siebel Database” on page 246](#)
- [“Updating Siebel File System Attachments” on page 246](#)
- [“Validating Dock Objects and Rule Definitions in the Siebel Database” on page 248](#)
- [“Verifying an Upgraded Oracle RDBMS After a Siebel Upgrade” on page 250](#)
- [“Setting Oracle Database Parameters After a Siebel Upgrade” on page 251](#)

Related Topics

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

Reapplying Schema Customizations in the Siebel Database

Upgrades: All Siebel upgrades.

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, several 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.

The upgrade process generates a report that you can review for information about tables that are either obsolete. This report, `xtndobstb1.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 `xtndobstb1.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 6.x, 7.0.x, & 7.5.x.

Environments: Development, production test, production.

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 inactivated (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 application function.

Updating Siebel File System Attachments

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."](#)

During the upgrade from Siebel 7.0.x, files from `S_WEB_CNTNT` were migrated to `S_CB_ASSET_VER`. In addition, at Siebel 7.5.x, the naming convention for file system attachments changed. Schema table names are included in the attachment file names.

You must run a utility to rename file system attachments that correspond to inactive tables for Siebel 7.x so that they are accessible by Siebel Business Applications.

Upgrading the Siebel File System

Perform the following procedure to upgrade the Siebel File System.

To upgrade the file system attachments

- 1 Navigate to the following directory:

Windows: *SIEBEL_ROOT\bin*

UNIX: *\$SIEBEL_ROOT/bin*

- 2 For each obsolete table, enter the following command:

Windows: `chng_file_sys.bat SOURCE_TABLE TARGET_TABLE "FILE_SYSTEM"`

UNIX: `chng_file_sys.ksh -s SOURCE_TABLE -t TARGET_TABLE -f "FILE_SYSTEM"`

where:

- *SOURCE_TABLE* = name of the original, inactive, table
- *TARGET_TABLE* = name of the new table to which the original data was migrated
- *"FILE_SYSTEM"* = name of the directory where the Siebel File System attachments reside (entered inside quotation marks)

Windows example:

```
chng_file_sys.bat S_EMPLOYEE_ATT S_CONTACT_ATT  
"C:\siebfile\att"
```

UNIX example:

```
chng_file_sys.ksh -s S_EMPLOYEE_ATT -t S_CONTACT_ATT  
-f "/usr/siebel/siebfile/att"
```

Review the renamed files carefully to verify that they can be accessed by Siebel Business Applications.

Make sure that attachment files for obsolete tables are renamed or copied to alternate locations. For example, since *S_EMPLOYEE_ATT* is migrated to *S_CONTACT_ATT*, you need to rename a file such as *S_EMPLOYEE_12-1ABC.SAF* to *S_CONTACT_12-1ABC.SAF*.

Migrating Files

Run the following procedure if you are upgrading from Siebel 7.0.x.

The table *S_LIT* was migrated to *S_CB_ASSET_VER*; therefore, you need to migrate files associated with migrated records. Perform the following steps to copy the files named *S_LIT*.SAF* and rename them to *S_CB_ASSET_VER*.SAF*, so that the files correspond to the new table name.

To migrate *S_LIT* files to *S_CB_ASSET_VER*

- 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 sadmin sadminpw SIEBEL "C:\siebfile" "C:\sea7xx\siebsrvr"  
"C:\sea7xx\dbsrvr"
```

UNIX example:

```
file_upg_mm.ksh SEBL sadmin sadminpw SIEBEL /usr/siebel/siebfile $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.

Validating Dock Objects and Rule Definitions in the Siebel Database

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

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's Siebel Technical Support or Oracle's Siebel Expert 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*.

Related Topic

[“Preserving Siebel Dock Objects and Visibility Rules” on page 130](#)

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
/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** Contact Oracle's Siebel Technical Support to determine if the errors must be resolved.

The following types of entries indicate that no errors were found:

- ❑ Errors: 0
- ❑ Syntax OK

Verifying an Upgraded Oracle RDBMS After a Siebel Upgrade

Upgrades: All Siebel upgrades.

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 Alert 1053 on Oracle's Siebel SupportWeb.

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:

```
ddlmp /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 /l siebelLogDir/ddlctl_verify_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 logs for errors. If the log indicates there are errors, contact Oracle's Siebel Technical Support.

Setting Oracle Database Parameters After a Siebel Upgrade

Upgrades: All Siebel upgrades.

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 Technical Note 582 on Oracle's Siebel SupportWeb.
- **(Oracle 9i only) optimizer_max_permutations.** Set this parameter to 100. This parameter is obsolete on Oracle 10g and later.
- 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.

18 Postupgrade Tasks for Siebel Applications

This area contains the following topics:

- [“Generating Siebel Reporting Relationships” on page 254](#)
- [“Setting Up Siebel Global Time Zone Support” on page 255](#)

Siebel ERM

- [“Upgrading Siebel ERM Approval Business Process Workflows” on page 256](#)
- [“Upgrading Siebel ERM Customized Microsite and Group News Pages” on page 257](#)
- [“Migrating Course Duration Information for Siebel Training” on page 258](#)
- [“Upgrading the Launch Field in Siebel Training LOV” on page 259](#)
- [“Upgrading the Test Status in Siebel Training” on page 259](#)
- [“Verifying Class and Session Times in Siebel Training” on page 260](#)

Siebel Marketing

- [“Upgrading Responsibilities in Siebel Marketing” on page 261](#)
- [“Reviewing Siebel Marketing Campaign Data” on page 262](#)
- [“Reviewing Renamed Fields in Siebel Marketing” on page 262](#)
- [“Displaying Regions in Siebel Marketing” on page 263](#)
- [“Revising Program Flowchart Icons in Siebel Marketing” on page 263](#)
- [“Setting Default Campaign Execution Options in Siebel Marketing” on page 264](#)
- [“Upgrading Activity Plans for Programs and Campaigns in Siebel Marketing” on page 264](#)
- [“Upgrading the Newsletter Offer Type in Siebel Marketing” on page 265](#)
- [“Configuring Universal Inbox in Siebel Marketing” on page 265](#)

Quote and Order Management

- [“Upgrading Siebel Purchase Orders” on page 266](#)
- [“Configuring Siebel Asset-Based Ordering” on page 266](#)
- [“Reviewing Siebel Address Data” on page 268](#)
- [“Upgrading Siebel Attribute Pricing” on page 269](#)
- [“Verifying Aggregate Discounts in Siebel Pricer” on page 270](#)

Workflows

- “Upgrading Siebel Seeded Workflows” on page 271
- “Upgrading Inbound Siebel Workflows” on page 271

Siebel Financial Services

- “Migrating Data to the Bankruptcy Status Field in Siebel Financial Services” on page 272

Related Topics

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

Generating Siebel Reporting Relationships

Upgrades: All Siebel upgrades.

Environments: Development, production test, production.

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 to Siebel 7.x 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 to Siebel 7.x, 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).

In Siebel 7.x, 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.

The standard Siebel 7.x configuration includes the Generate Reporting Relationships feature as a hidden button on the Position List Applet NB. You need to go through Siebel Tools configuration to expose this button.

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 expose the Generate Reporting Relationships button

- 1 Open Siebel Tools and navigate to the applets folder.
- 2 Find and select the Position List Applet NB applet record.
- 3 Right-click on the record and select Edit Web Layout.
- 4 Drag the GenReportRel button from the Controls/Columns window into one of the button placeholders in the applet layout (that is, one of the empty “x” placeholders in the blue header area of the applet layout).
- 5 Repeat this step for the three different modes (Base, Edit, and Edit List) in which the applet can be displayed. The easiest way to switch between the different modes is to use the Mode drop-down that appears in the Web Controls toolbar of Siebel Tools. After you have dragged the button and dropped it into all three modes of the applet layout, close the layout editor and save your changes.
- 6 Recompile the applet into your existing `siebel.srf` as used by the Web client.
- 7 Launch the Web client using the SRF compiled in the previous step so that the Generate Reporting Relationships button can be invoked.

It is recommended that you do not make this button available in the standard `siebel.srf` file used by your organization in order to preserve control over who can press this button and when it can be pressed.

To generate reporting relationships

- 1 If you have an active Siebel Remote environment, confer with a Siebel Systems 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: All Siebel upgrades.

Environments: Development, production test, production.

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 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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 verify conformance to data access policies at your site, see [“Configuring Universal Inbox in Siebel Marketing” on page 265](#).

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.

Upgrading Siebel ERM Customized Microsite and Group News Pages

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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 CSSSWEFrameListERMPageRender.
- 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 Technical Notes on Oracle's Siebel SupportWeb.

Migrating Course Duration Information for Siebel Training

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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.

Upgrading the Launch Field in Siebel Training LOV

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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.

Upgrading the Test Status in Siebel Training

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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.

Verifying Class and Session Times in Siebel Training

Upgrades from: Siebel 7.0.x, 7.5.x, & 7.7.x.

Environments: Production test, production.

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.

Upgrading Responsibilities in Siebel Marketing

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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.

Reviewing Siebel Marketing Campaign Data

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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.

Reviewing Renamed Fields in Siebel Marketing

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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 40 on page 263](#).

Table 40. 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

Displaying Regions in Siebel Marketing

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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.

Revising Program Flowchart Icons in Siebel Marketing

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

After the upgrade, 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*.

Setting Default Campaign Execution Options in Siebel Marketing

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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.

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*.

Upgrading Activity Plans for Programs and Campaigns in Siebel Marketing

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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.

Upgrading the Newsletter Offer Type in Siebel Marketing

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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.

Configuring Universal Inbox in Siebel Marketing

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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.

Upgrading Siebel Purchase Orders

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

If upgrading from a release prior to Siebel 7.7, 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 6.x, 7.0.x, & 7.5.x.

Environments: Production test, production.

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 41](#). If you have written scripts or made other changes regarding these buttons or named methods, you must manually add these after the upgrade.

Table 41. 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) 7.8.x
- From Siebel Business Applications 7.8.x to Siebel SIA 7.8.x
- From Siebel Financial Services 6.2.1 on IBM z/OS platforms to Siebel SIA 7.8.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."](#)

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/log/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.

Upgrading Siebel Attribute Pricing

Upgrades from: Siebel 7.0.x, 7.5.x, & 7.7.x.

Environments: Production test, production.

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.

Verifying Aggregate Discounts in Siebel Pricer

Upgrades from: Siebel 7.0.x, 7.5.x, & 7.7.x.

Environments: Production test, production.

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.

Upgrading Siebel Seeded Workflows

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

Environments: Development, production test, production.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform a Siebel Database Upgrade.”](#)

In Siebel 7.7, workflow definitions were relocated to the Tools Repository.

Customizations to seeded workflows were saved and migrated during upgrade, but you must manually reimplement them in order for them to work properly.

To upgrade seeded workflows

- 1 Purge workflow instances.

For information on purging these tables, see the topic on purging workflow process instances in *Siebel Business Process Designer Administration Guide*.

- 2 In the Siebel Repository, revise each seeded workflow so that a new copy is created with a new version number.
- 3 Manually merge in your customizations, and then deploy and activate the workflow.

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"?><?'

Migrating Data to the Bankruptcy Status Field in Siebel Financial Services

Upgrades from: Siebel 7.0.x, 7.5.x, & 7.7.x.

Environments: Production test, production.

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.

19 Tuning the Siebel Upgrade Files

This area contains the following topics:

- [“About Tuning Siebel Production Upgrade Files” on page 273](#)
- [“Starting and Stopping Siebel Upgrade Tuner” on page 278](#)
- [“Managing Parallel Threads Using Siebel Upgrade Tuner” on page 280](#)
- [“Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner” on page 282](#)
- [“Transferring UNIX Files for Use by Siebel Upgrade Tuner” on page 285](#)
- [“Rolling Back Siebel Upgrade Tuner Changes” on page 288](#)

Related Topic

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

About Tuning Siebel Production Upgrade Files

Upgrades: All Siebel upgrades.

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 42](#).

Table 42. Upgrade Tuner Support for Parallel Table and Index Threads

Operating System	Oracle	IBM DB2	MSSQL
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 43](#).

Table 43. Upgrade Tuner Support for Zero-Row SQL Deactivation

Operating System	Oracle	IBM DB2	MSSQL
Windows	Yes	Yes	Yes
UNIX	Yes	Yes	Yes
z/OS	N/A	No	N/A

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

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 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 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 SQL statements that affect no table rows and therefore no data. This capability improves 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.

File	Location on a Windows Host
summary.xml	<i>SIEBEL_ROOT</i> \log\upgrep_prod_ <i>VERSION</i> \ For example, <i>SIEBEL_ROOT</i> \log\upgrep_prod_77\summary.xml
master_upgrep_prod_ <i>VERSION</i> .ucf	<i>SIEBEL_ROOT</i> \bin\ For example, <i>SIEBEL_ROOT</i> \bin\master_upgrep_prod_77.ucf

File	Location on a Windows Host
schema*.ddl	<i>DBSRVR_ROOT\DBPLATFORM\</i> For example, <i>DBSRVR_ROOT\Oracle\schema.ddl</i> , <i>schema_i1.ddl</i> , <i>schema_t1.ddl</i>
driver_upgrep_prod_ VERSION.ucf	<i>DBSRVR_ROOT\DBPLATFORM\upgrade\VERSION\</i> For example, <i>DBSRVR_ROOT\Oracle\upgrade\v7_7\driver_upgrep_prod_77.ucf</i>
*.sql	<i>DBSRVR_ROOT\DBPLATFORM\upgrade\VERSION\</i> For example, <i>DBSRVR_ROOT\Oracle\upgrade\v7_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 Upgrade Wizard uses to upgrade the database.

Driver Configuration File

When you add or remove parallel threads and save your changes, Upgrade Tuner modifies the driver configuration file, for example *driver_upgrep_77.ucf*. The driver configuration file is a text file that contains a series of steps. The steps specify the commands that control the production upgrep. 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.

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_upgrep_77.ucf* to *driver_upgrep_77.ucf.orig*. It then makes changes to *driver_upgrep_77.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_upgrep_77.ucf* to *driver_upgrep_77.ucf.old*. It then makes changes to *driver_upgrep_77.ucf*.
- When you save your changes after the third session and all following, 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 ddlimp 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 ddlimp 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+Exit, Upgrade Tuner creates a schema.ddl thread-file for the new thread.

For example, you have no 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.

Starting and Stopping Siebel Upgrade Tuner

Upgrades: All Siebel upgrades.

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.

Related Topic

[“About Tuning Siebel Production Upgrade Files” on page 273](#)

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 Logparse” on page 192.](#)
- 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 285.](#)

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 inactivated 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.

Managing Parallel Threads Using Siebel Upgrade Tuner

Upgrades: All Siebel upgrades.

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.

Related Topics

["About Tuning Siebel Production Upgrade Files" on page 273](#)

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 Logparse" on page 192](#).
- 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 285](#).
- Start Upgrade Tuner. See ["Starting and Stopping Siebel Upgrade Tuner" on page 278](#).

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 upgrep 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.

Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner

Upgrades: All Siebel upgrades.

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 upgrep.

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.

Related Topics

[“About Tuning Siebel Production Upgrade Files” on page 273](#)

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 Logparse” on page 192](#).
- 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 285](#).
- 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 278](#).

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.

Transferring UNIX Files for Use by Siebel Upgrade Tuner

Upgrades: All Siebel upgrades.

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.

Related Topics

[“About Tuning Siebel Production Upgrade Files” on page 273](#)

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 Logparse” on page 192.](#)

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/upgrep_prod_ <i>VERSION</i> / For example, \$SIEBEL_ROOT/log/upgrep_prod_77/summary.xml
master_upgrep_prod_ <i>VERSION</i> .ucf	\$SIEBEL_ROOT/bin/ For example, \$SIEBEL_ROOT/bin/master_upgrep_prod_77.ucf
schema*.ddl	<i>DBSRVR_ROOT</i> /DBPLATFORM/ For example, <i>DBSRVR_ROOT</i> /Oracle/schema.ddl, schema_t1.ddl, schema_t2.ddl
driver_upgrep_prod_ <i>VERSION</i> .ucf	<i>DBSRVR_ROOT</i> /DBPLATFORM/upgrade/ <i>VERSION</i> / For example, <i>DBSRVR_ROOT</i> /Oracle/upgrade/v7_7/driver_upgrep_prod_77.ucf
*.sql	<i>DBSRVR_ROOT</i> /DBPLATFORM/upgrade/ <i>VERSION</i> / For example, <i>DBSRVR_ROOT</i> /Oracle/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.

Rolling Back Siebel Upgrade Tuner Changes

Upgrades: All Siebel upgrades.

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 driver_upgrep_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 driver_upgrep_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 driver_upgrep_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 driver_upgrep_prod_77.ucf and pret.sql as examples.

Related Topics

["About Tuning Siebel Production Upgrade Files" on page 273](#)

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 driver_upgrep_prod_77 and pret.sql to new names.
- 2 Copy driver_upgrep_prod_77.old to driver_upgrep_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 driver_upgrep_prod_77 and pret.sql.
- 2 Copy driver_upgrep_prod_77.orig to driver_upgrep_prod_77.
- 3 Copy pret.sql.orig 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.

A

Siebel Marketing Upgrade Reference

This area contains the following topics:

- [“Obsolete Siebel Marketing Data” on page 291](#)
- [“Obsolete Business Objects in Siebel Marketing” on page 297](#)
- [“Obsolete or Replaced Views in Siebel Marketing” on page 298](#)

Related Topics

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

Obsolete Siebel Marketing Data

Upgrades from: Siebel 6.x, 7.0.x, & 7.5.x.

[Table 44 on page 291](#) lists marketing data that is obsolete as of Siebel 7.7. In the table, the Comments field explains how the data is handled.

Table 44. 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			

Table 44. 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 44. Obsolete Siebel Marketing Data as of Siebel 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
eNewsletter Offers	Obsolete		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_QRY	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 Siebel Analytics repository.

Table 44. 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 Siebel Analytics repository.

Table 44. 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 Siebel Analytics 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 Siebel Analytics repository.

Table 44. 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_ATTRFAMLVMB	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 44. 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 Siebel Analytics repository.
Campaign Load Mappings (Contact Key Formats)	Obsolete			
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_PRPERFMEAS	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 6.x, 7.0.x, & 7.5.x.

[Table 45 on page 297](#) lists the Siebel Repository business objects that are obsolete as of Siebel 7.7.

Table 45. Obsolete Siebel Marketing Business Objects as of Siebel 7.7

Business Object Name	Tools Project
Customer Hierarchies	Server (DD)
DBM Campaign	eMarketing - Campaign Views
DBM Campaign Btn	eMarketing - Campaign Views

Table 45. Obsolete Siebel Marketing Business Objects as of Siebel 7.7

Business Object Name	Tools Project
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 & 7.5.x.

Table 46 on page 298 lists the Siebel 7.x Siebel Repository views that are obsolete or have been replaced as of Siebel 7.7.

Table 46. 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	
All Measures View	Marketing Administration	All Measures	Obsolete	
All Segments across My Organizations	Segment Screen	All Segments	Replaced	All Marketing Segments View
All Tables View	Marketing Administration	Tables	Obsolete	
Attribute Families View	Marketing Administration	Attribute Families	Obsolete	
Bound Measures View	Marketing Administration	Bound Measures	Obsolete	

Table 46. 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 (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	
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
Campaign Groups View - Admin (SCW)	Campaign Screen (SCW)	Groups	Inactivated	

Table 46. 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 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	
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
Campaign Plan Activities View	Campaign Management Screen	Activities	Replaced	Campaign Activity View Admin

Table 46. 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 Responses View (SCW)	Campaign Screen (SCW)	Responses	Replaced	Campaign Responses View
Campaign Results (DBM) Input Summary View	Campaign Management Screen	Results > Results Summary	Obsolete	
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	
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	
Campaign Waves View	Campaign Management Screen	Waves	Replaced	Program Schedule Detail View
Columns View	Marketing Administration	Fields	Obsolete	
Custom Measure Aggregation View	Marketing Administration	Custom Measure > Aggregation	Obsolete	
Custom Measure Details View	Marketing Administration	Custom Measure > More Info	Obsolete	
Custom Measure Restriction View	Marketing Administration	Custom Measure > Restriction	Obsolete	

Table 46. Siebel 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Siebel 7.x Screen	Siebel 7.x View Caption	Outcome	Replacement View
Customer Hierarchies View	Marketing Administration	Customer Hierarchies	Obsolete	
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	
Decision Wizard Player View	Decisions Administration Screen		Obsolete	
Decisions Detail Input Parameters View	Decisions Administration Screen	Identifiers	Obsolete	
Decisions Detail Output Parameters View	Decisions Administration Screen	Outputs	Obsolete	
Decisions Detail Session Parameters View	Decisions Administration Screen	Real Time Inputs	Obsolete	
Decisions Detail View	Decisions Administration Screen	More Info	Obsolete	
Enewsletter Attachments View	Offer Screen	eNewsletter > Attachments	Obsolete	
Enewsletter Email Profile View	Offer Screen	eNewsletter > Profile	Obsolete	
Enewsletter Offer Detail View	Offer Screen	eNewsletter > More Info	Obsolete	
Enewsletter Offer Related URLs View	Offer Screen	eNewsletter > Related URLs	Obsolete	
Enewsletter Offer Sections List View	Offer Screen	eNewsletter > Sections	Obsolete	

Table 46. 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 View	Offer Screen	eNewsletter Offers	Obsolete	
Enewsletter Rules View	Offer Screen	eNewsletter Rules	Obsolete	
External Contacts Mapping	Marketing Administration	Campaign Load Mapping	Obsolete	
Filters Detail View	Marketing Administration	Filters	Obsolete	
Filters View	Marketing Administration	Filters	Obsolete	
Financial Modeler (DBM) View	Campaign Management Screen	Forecast > Input Summary	Obsolete	
Financial Modeler (DBM) View-Fixed Costs	None	Financial Modeler	Obsolete	
Financial Modeler (DBM) View-Inbound Costs	None	Financial Modeler	Obsolete	
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	
Financial Modeler (DBM) View-Revenue Inputs	None	Financial Modeler	Obsolete	
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	
Joins View	Marketing Administration	Joins	Obsolete	

Table 46. Siebel 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Siebel 7.x Screen	Siebel 7.x View Caption	Outcome	Replacement View
List Formats View	Marketing Administration	Output List Layouts	Obsolete	
List Measures View	Marketing Administration	List Measures	Obsolete	
Marketing Plans Funds	Marketing Plans	Funds	Inactivated	
Measure Based Attributes View	Marketing Administration	Buckets	Obsolete	
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	
My Segments View	Segments	My Segments	Replaced	My Marketing Segments View
My Team's Decisions List View	Decision Administration	My Team's Decisions	Obsolete	
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	
Program Responses View	Program Occurrence Screen	Responses	Obsolete	
Program Snapshots View	Program Stages Screen	Snapshot Elements	Obsolete	
Program Stages View	Program Screen	Stages	Inactivated	

Table 46. Siebel 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Siebel 7.x Screen	Siebel 7.x View Caption	Outcome	Replacement View
Related Events View for eNewsletter Offers	Offer Screen	Related Events	Obsolete	
Related Web Offers View f/ Enewsletter	Offer Screen	Related Web Offers	Obsolete	
Related Web Surveys View f/ Enewsletter	Offer Screen	Related Web Surveys	Obsolete	
Response Detail View (SME) - More Info	Response Screen	More Info	Replaced	Response Detail View (Detail)
Segment Criteria View	Segment Screen	More Info	Obsolete	
Segment Program Stages View	Segment Screen	Program Plans	Obsolete	
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	

B

Tables Modified or Seeded During a Siebel Upgrade

This area includes the following topics:

- [“Important Schema Changes at Siebel 7.8” on page 307](#)

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, & 7.7.x.

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 47](#).

Table 47. 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 48](#).

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 48. Bundle Discount Tables

Obsolete Tables	New Tables
S_PRIMDL	S_BUNDLE_SEQ
S_PRIMDL_FCTR	■ 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 49](#).

Table 49. 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 105
- S_CONTACT and S_ORG-GROUP, about multi-org visibility for 105

aggregate categories (new), reviewing 231

AIX memory allocation segment space, securing 130

applet functionality, reviewing 223

attachments

- file system attachments, updating 246

audience for guide 95

B

binary sort order

- Microsoft SQL Server, verifying database was created using 147

business component definitions, changes to party model 108

C

clustered indexes

- rebuilding 150

collation sequence

- See database sort order, verifying

configuration parameters, about and table 148

copied objects, automatic upgrade of upgrade ancestor, choosing 51

custom database schema

- license keys, adding new 189

Customer Repository, preparing for merge

- See *also* repository merge, performing
- inheriting upgrade behavior, about 49
- New Customer repository, preparing to import 209, 211
- upgrade inheritance, configuration steps 199

D

data migration

- party model, support of 108

database server configuration, verifying

- Oracle database server configuration 142

database sort order, verifying

- about 104
- IBM DB2 UDB, verifying database created using identity sort 131
- Microsoft SQL Server, verifying sort order on 147

database, preparing for upgrade

- AIX memory allocation segment space, securing 130
- binary sort order, verifying database created using 147
- database sort order, verifying 104
- Microsoft SQL Server, verifying sort order on 147
- mobile and dedicated users, preparing for upgrade 164
- New Customer Repository, preparing to import 209, 211
- RDBMS software, upgrading 98

dedicated users, preparing for upgrade 164

E

Encryption Upgrade Utility

- RC2 encryption, about upgrading to 102

F

file system

- attachments, updating 246

G

Generate Reporting Relationships button, exposing 254

global deployment, postupgrade tasks

- global time zone, about setting up environment to support 255
- global time zone, enabling after an upgrade 255
- Unicode, migrating to 100

global time zone

- See global deployment, postupgrade tasks

grid-based applets, reviewing 225

guide

- audience for 95
- naming and typographical conventions 20

I**IBM DB2 UDB, preparing database for upgrade**

- 16-KB temporary tablespace, creating 135
- 32-KB temporary tablespace, creating 135
- database configuration parameters, about and table 133
- DB2 permission, verifying on AIX and Solaris 134
- DB2 UDB Application Development Client, verifying installation of 138
- identity sort order, verifying database created using 131
- instance owner permission, verifying for AIX and Solaris 134
- RDBMS software, upgrading 98
- sort order, verifying on 131

ICL, about 55**identity sort order, verifying database created using 131****Incorporate Custom Layouts, about indexes 55**

- clustered indexes, rebuilding 150

inheriting upgrade behavior, about 49**item identifier, Web template 237****L****license keys**

- custom database schema, adding new keys 189

log files

- archiving manually 197
- SiebSrvr log files, about saving 188

M**Microsoft SQL Server**

- clustered indexes, rebuilding 150
- configuration parameters, about and table 148
- database sort order, verifying 147
- Siebel Software Configuration utility, changing language utility runs 178
- temporary database space, about 148

mobile users, preparing for upgrade 164**multilingual deployment, installing additional languages 99****multilingual seed data**

- Siebel Database Server for DB2, installing 99

multi-value group shuttle applets, reviewing 227**N****naming conventions 20****O****objects**

- descendents or copies, viewing 199

Oracle database

- server configuration, before upgrading tasks 142
- sort order, verifying on 141
- upgrading, steps before upgrading 142

P**party model**

- about 106
- business component definitions 108
- data migration, support of 108
- data model, changes to (diagram) 108
- obsolete tables in 7.x 108

planning

- Configurator 123, 124
- Employee Relationship Management (ERM), about 114
- guidelines 95
- handheld devices 122
- Resonate Central Dispatch 122
- Workflow Designer 121

postupgrade tasks

- See *also* pre-upgrade tasks; RC2 encryption, upgrading to
- asset-based ordering 266
- dock object and rule definitions, verifying are correct 248
- file system attachments, updating 246
- Generate Reporting Relationships button, exposing 254
- inbound workflows, updating 271
- log files, manually archiving 197
- previously unused tables now used in 7.x, examples of 245
- Purchase Orders 266
- reporting relationships, generating (procedure) 254
- reporting relationships, generating, about 254
- seeded workflows, incorporating prior customizations into 271
- seeded workflows, using 271
- symbolic string model, upgrading to 242
- tables not used in 7.x, examples of 245
- Transaction Amount field in Payment lines, updating 266
- Universal Inbox 265

pre-upgrade tasks

- AIX memory allocation segment space, securing 130
 - binary sort order, verifying database created using 147
 - database sort order, verifying 104
 - Microsoft SQL Server, verifying sort order on 147
 - mobile and dedicated users, preparing for upgrade 164
 - New Customer Repository, preparing to import 209, 211
 - RDBMS software, upgrading 98
- production environment upgrade**
- development environment, upgrading without 188
 - global time zone, about operating in 255
 - New Customer Repository, preparing to import 209, 211
 - preparing production upgrade on UNIX 183
 - Transaction Amount field in Payment Lines, updating 266

R**RC2 encryption, upgrading to**

- about 102

RDBMS software, preparing for upgrade

- 98

reporting relationships

- about generating 254
- Generate Reporting Relationships button, exposing 254
- generating (procedure) 254

repositories, upgrading

- about 45
- EIM temporary columns, generating 207
- Incorporate Custom Layouts (ICL) 55
- performing merge 201
- postmerge utilities, about 52
- renaming 177
- upgrade behavior, inheriting 49

repository merge results

- determining success 205
- reviewing conflicts 208
- reviewing deleted objects 239
- reviewing obsolete objects 240

rich text controls, revising**RTCs, revising**

- 229

S**S_CONTACT, multi-org visibility for S_LIT files, upgrading to S_CB_ASSET_VER**

- files 246

S_ORG_GROUP, multi-org visibility for

- 105

S_PARTY

- See party model

S_SRC_PAYMENT table, about columns

- added 166

seed data, installing multilingual seed data

- 99

shuttle applets, reviewing

- 227

Siebel Application Integration (EAI), about using SAP and upgrading

- 26

Siebel Business application, upgrading

- multilingual deployment, installing additional languages 99

Siebel data model

- access control, about 105
- party model, business component definitions 108
- party model, changes to data model (diagram) 108
- party model, data migration in support of 108
- party table, about 106
- party table, obsolete tables in 7.x 108
- S_SRC_PAYMENT Table, columns added to 166

Siebel Database Schema, upgrading

- restarting upgrade 184

Siebel Database Server software

- multilingual deployment, installing additional languages 99

Siebel Financial Services

- household data integrity, verifying 160

Siebel Marketing

- obsolete 7.x business objects 297
- obsolete 7.x data 291
- obsolete 7.x views 298

Siebel Purchase Orders, updating

- Transaction Amount field 266

Siebel Repository, preparing for upgrade

- See also Siebel Repository; repository merge, performing
- development environment repositories, preparing 177
- dock objects and visibility rules, recording 130

Siebel Software Configuration utility

- language, changing 178
- The Siebel Upgrade Wizard, about 36

Siebel Tools

- See also Siebel Repository

Siebel Upgrade Wizard

- about 36

Siebel Workflow

- inbound workflow, updating 271
- seeded workflows, incorporating prior customizations into 271
- seeded workflows, using 271

SiebSrvr log files, about saving 188**sort order**

- binary sort order, verified database created using 147
- Microsoft SQL Server, verifying on 147

T**tables, preparing for upgrade**

- DB2 UDB long columns, identifying for truncation 98, 104, 128, 129, 138, 142, 164, 209, 211

Tools repository, renaming 177**Transaction Amount field in Payment lines, updating** 266**typographical conventions** 20**U****Unicode**

- migrating to, about and procedure 100
- migration, planning considerations 100

upgrade inheritance, configuration steps 199**upgrade, planning for**

- resources 93

upgrading ancestor, choosing an 51**upgrading inheritance**

- descendent or copies of an object, viewing 199
- parent to descendents, propagating changes from 52

V**view tabs, verifying display** 234**visibility filters, reviewing** 231**W****Web template item identifier** 237**workflows**

- inbound, updating 271
- seeded, incorporating prior customizations into 271
- seeded, using 271