



Siebel Database Upgrade Guide

Siebel Innovation Pack 2016, Rev. A
June 2016

ORACLE®

Copyright © 2005, 2016 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Contents

Chapter 1: What's New in This Release

Chapter 2: Overview of Siebel Database Environments

Supported Upgrade Paths for Siebel CRM Version 16.0 17

Terms Used in This Guide 21

Naming Conventions Used in This Guide 21

About File Paths and Commands in Siebel Database Upgrade Topics 22

About Supported Siebel Upgrade Paths 24

About Siebel Upgrade Environments 24

Chapter 3: Siebel Database Upgrade Planning

About Siebel Upgrade Planning Resources 29

Guidelines for Planning Your Siebel Database Upgrade 32

About Upgrading Your RDBMS in the Siebel Environment 35

About Database Sort Order in the Siebel Environment 36

About the Siebel Database Configuration Utilities 37

About the Siebel Upgrade Wizard and Driver Files 43

About Siebel Additive Schema Changes Mode 46

About the Siebel Database Upgrade Log Files 48

About the Siebel Case Insensitivity Wizard 50

About the Siebel Repository Merge 51

About Single-Step Repository Upgrade 56

About Inheriting Upgrade Behavior in a Siebel Upgrade 57

About the Siebel Postmerge Utilities 59

About Tuning Siebel Production Upgrade Files 63

Chapter 4: Application Planning for a Siebel Upgrade

Upgrade Planning for Multilingual Siebel Deployments 69

Upgrade Planning for Language String Migration	71
Upgrade Planning for Siebel Unicode Support	71
Upgrade Planning for Siebel RC4 or AES Encryption	73
Upgrade Planning for Siebel Web Template Files and Style Sheets	73
Upgrade Planning for Siebel Access Control	75
Upgrade Planning for Migrating Siebel Address Data	75
Upgrade Planning for Siebel Workflow Designer	76
Upgrade Planning for Mobile Devices in the Siebel Environment	79
Upgrade Planning for Resonate Central Dispatch in the Siebel Environment	80
Upgrade Planning for Siebel String Translation	80
Upgrade Planning for Siebel Personalization	81
Upgrade Planning for Siebel Pricer and Order Management	81

Chapter 5: Preparing for Siebel Database Upgrade

Verifying Siebel Database Connectivity	83
Preparing Siebel Tables and Views for Upgrade	83
Preparing Siebel Custom Indexes for Upgrade	84
Exporting Siebel Interface Table Data	85
Archiving Unneeded Siebel Repositories	85
Preserving Siebel Dock Objects and Visibility Rules	85
Securing AIX Memory Allocation Segment Space for the Siebel Database	86
Preparing for a Multilingual Upgrade	86

Chapter 6: Preparing an IBM DB2 Database for a Siebel Upgrade

Verifying the IBM DB2 Client for a Siebel Upgrade	89
Verifying IBM DB2 Sort Order for a Siebel Upgrade	89
Setting IBM DB2 Parameters for a Siebel Upgrade	90
Verifying IBM DB2 Permissions for a Siebel Upgrade	92
Verifying IBM DB2 Instance Owner Permissions for a Siebel Upgrade	92
Creating IBM DB2 Temporary Tablespaces and Bufferpools for a Siebel Upgrade	92
Analyzing IBM DB2 Custom Tablespace Requirements for a Siebel Upgrade	93

Verifying the IBM DB2 Application Development Client for a Siebel Upgrade	95
Identifying IBM DB2 Long Columns for Truncation in a Siebel Upgrade	96

Chapter 7: Preparing an Oracle Database for a Siebel Upgrade

Verifying Oracle Database Sort Order for a Siebel Upgrade	99
Setting Oracle Database Configuration Parameters for a Siebel Upgrade	99
Verifying Oracle Database Parameters for Multiple Processors in a Siebel Upgrade	101
Verifying the Oracle Database ODBC Definition for a Siebel Upgrade	101

Chapter 8: Preparing a Microsoft SQL Server Database for a Siebel Upgrade

Verifying Microsoft SQL Server Sort Order for a Siebel Upgrade	103
Setting Microsoft SQL Server Temporary Space Size for a Siebel Upgrade	104
Setting Microsoft SQL Server Configuration Parameters for a Siebel Upgrade	104
Rebuilding Microsoft SQL Server Clustered Indexes for a Siebel Upgrade	105

Chapter 9: Preparing Siebel Application Data for Upgrade

Preparing Siebel Workflow Processes for Upgrade	107
Preparing Siebel Customized Seed Data for Upgrade	108
Migrating Siebel Household Data	109
Preparing Siebel Mobile User Data for Upgrade	111
Preparing Siebel Address Data for Upgrade	112
Migrating Siebel Address Data from Custom Extension Columns	113
Preparing Siebel Territory Management Rules for Upgrade	120
Preparing Siebel Customizable Product Data for Upgrade	121

Chapter 10: Upgrading the Siebel Database

Example of a Siebel Development Environment Upgrade Flow	123
Renaming the Siebel Repository	129
Copying Ancestor Repositories	130
Changing the Siebel Database Configuration Utilities Language	132
Preparing to Run the Siebel Database Configuration Wizard	132

Running the Siebel Database Configuration Wizard on Windows	136
Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on Windows	138
Backing Up the S_LST_OF_VAL Table on Windows	138
Deactivating Records for Unshipped or Unwanted Languages on Windows	139
Importing Newly Created Records from the Data File on Windows	140
Running the Siebel Database Configuration Wizard on UNIX	141
Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on UNIX	143
Backing Up the S_LST_OF_VAL Table on UNIX	143
Deactivating Records for Unshipped or Unwanted Languages on UNIX	144
Importing Newly Created Records from the Data File on UNIX	144
Starting the Siebel Upgrade Wizard	145
Moving the Siebel Repository Files	149
Running the Siebel Case Insensitivity Wizard	149
Applying Siebel Additive Schema Changes	154
Regenerating SQL Files for a Siebel Upgrade	155
Identifying and Dropping Obsolete Indexes for a Siebel Upgrade	156
Preparing for a Nondevelopment Environment Siebel Upgrade	157
Chapter 11: Reviewing the Siebel Upgrade Log Files	
Summarizing Siebel Log Files Using the Logparse Utility	159
Reviewing Siebel Upgrade Log Files for Errors	161
Manually Archiving Siebel Upgrade Log Files	165
Chapter 12: Performing the Siebel Repository Merge	
Preparing for the Repository Merge	167
Configuring Siebel Repository Objects to Inherit Upgrade Behavior	168
Performing a Siebel Repository Merge	169
Reviewing the Siebel Repository Merge Log Files	171
Generating Siebel Enterprise Integration Manager Temporary Columns	174
Reviewing Siebel Repository Object Property Conflicts	174
Regenerating the Siebel Repository Definition Files	178
Deleting Unneeded Siebel Repository Files	180

Migrating Siebel Repository Objects to the Standard User Interface	180
Running the Siebel Postmerge Utilities	185

Chapter 13: Performing the Siebel Incremental Repository Merge

Process of Meeting the Requirements for an Incremental Upgrade to Siebel Innovation Pack 2016, Using the Incremental Repository Merge	187
---	-----

Before You Begin	188
Preparing Siebel Tools for Custom ODBC Data Source Names on Oracle Database	188
Preparing Siebel Tools for Custom ODBC Data Source Names for All Databases	189
Editing the Siebel Tools Configuration File	190

Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x (SIA Repository) and Version 15.x	190
---	-----

Preparing to Run the Siebel Database Configuration Wizard	191
Requirements for Running the Siebel Database Configuration Wizard on Windows	195
Running the Siebel Database Configuration Wizard on Windows	195
Running the Siebel Database Configuration Wizard on UNIX	196
Preparing to Start the Siebel Upgrade Wizard	197
Starting the Siebel Upgrade Wizard	198
Stopping the Siebel Upgrade Wizard on Windows	199
Stopping the Siebel Upgrade Wizard on UNIX	200
Preparing to Restart the Merge	200
Restarting the Merge	202
Executing the RUNSTATS Command on Oracle Database During the Pause Following New Repository Creation	202
Executing the RUNSTATS Command on IBM DB2 During the Pause Following New Repository Creation	203
Executing the RUNSTATS Command on Microsoft SQL Server During the Pause Following New Repository Creation	204

Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x (SIA Repository)	204
--	-----

Upgrading the Siebel Database Schema	205
Executing Incremental Repository Merge on UNIX	213
About Siebel Repository Merge Errors	213
Reviewing the Siebel Repository Merge Log Files	215
Reviewing the Hierarchical Merge Report	216
Reviewing Siebel Repository Object Property Conflicts	218
Reviewing Log Files for All Upgrades	220
Reviewing Conflicts at the Attribute Level	220
Marking Conflict Resolution as Complete Using Siebel Tools	221

Upgrading a Custom Database Schema	221
------------------------------------	-----

Verifying the Repository After Upgrade 224

About Migrating the XML Manifest Data for Siebel Open UI from Previous Releases to the New Manifest Data in the Database 225

Default Configuration for Generic Applet Types and Configuration of Non-Applet User Interface Constituents 228

Generic Applet Types from the XML Manifest File 230

Differences Between Customer Manifest Data and Oracle Manifest Data 234

File Selection Rules 235

Checking the Post-upgphys Verification Process 235

About Migrating Siebel Web Template Data 236

Process of Regenerating the Siebel Repository Definition Files 237

Regenerating the schema.ddl File 237

Regenerating the custrep.dat File 238

Editing the Siebel Tools Configuration File After the Development Environment Merge 239

Performing a Production Test or Production Environment Migration from Siebel CRM Version 8.1.1.x (SIA Repository) 240

Chapter 14: Siebel Postmerge Development Tasks

Reviewing Objects Deleted from the Siebel Repository 241

Reviewing Obsolete Objects in the Siebel Repository 242

Upgrading to the Siebel Symbolic String Model 243

Dropping IBM DB2 8-KB Tablespaces and Buffers After a Siebel Merge 243

Updating Siebel Enterprise Application Integration (EAI) 244

Chapter 15: Postupgrade Tasks for the Siebel Database

Reapplying Schema Customizations in the Siebel Database 245

Migrating Address Data After a Direct SEA to SIA Upgrade 246

Validating Dock Objects and Rule Definitions in the Siebel Database 247

Verifying an Upgraded Oracle Database After a Siebel Upgrade 249

Setting Oracle Database Parameters After a Siebel Upgrade 250

Chapter 16: Reviewing the Siebel User Interface

Troubleshooting Postmerge Siebel User Interface Problems 251

Reviewing Siebel Grid-Based Applets 252

Reviewing Siebel User Interface Navigation 254

Reviewing Siebel Multi-Value Group Shuttle Applets	255
Revising Siebel UI Rich Text Controls	258
Reviewing New Siebel UI Aggregate Categories	259
Revising Siebel Visibility Filters to Display Correctly	259
Assigning a Category and Type to Siebel Chart Views	260
Assigning a Category and Type to Siebel Explorer Views	261
Setting Up Navigation to Inaccessible Siebel Detail Views	262
Eliminating Obsolete Siebel UI Fields	263
Reviewing Siebel UI Objects Affected by Incorporate Custom Layout	263
Reviewing Required Fields in the Siebel User Interface	264
Assigning an Item Identifier to Siebel Web Template Items	265

Chapter 17: Postupgrade Tasks for Siebel Business Applications

Activating License Keys	267
Generating Siebel Reporting Relationships	268
Setting Up Siebel Global Time Zone Support	269
Displaying Regions in Siebel Marketing	270
Configuring Siebel Marketing Purchase Orders for Display	270
Upgrading Siebel Attribute Pricing	271
Verifying Aggregate Discounts in Siebel Pricer	272
Upgrading Inbound Siebel Workflows	273

Chapter 18: Tuning the Siebel Upgrade Files

Starting and Stopping Siebel Upgrade Tuner	275
Managing Parallel Threads Using Siebel Upgrade Tuner	277
Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner	280
Transferring UNIX Files for Use by Siebel Upgrade Tuner	283
Rolling Back Siebel Upgrade Tuner Changes	286

Appendix A: Implementing Siebel High-Availability Upgrade Using Oracle GoldenGate to Upgrade to the Current Release

Overview of Oracle GoldenGate Implementation	290
About Siebel High Availability Upgrade Files	291
Extracting Oracle GoldenGate Files	291
Example of Implementing a Siebel High-Availability Upgrade	298
Limitations When Performing a Siebel Upgrade with Oracle GoldenGate	300

Appendix B: Overview of Performing a Siebel Database Upgrade

Roadmap for Siebel Database Environment Upgrade	301
Process of Upgrading a Siebel Development Environment	308
Process of Upgrading a Siebel Production Test Environment	315
Process of Tuning Siebel Upgrade Performance	320
Process of Upgrading a Siebel Production Environment	324

Appendix C: Siebel Incorporate Custom Layout Upgrade Option

About the Siebel Incorporate Custom Layout Upgrade Option	333
---	-----

Appendix D: Siebel Case Insensitivity Wizard

Overview of What the Case Insensitivity Wizard Does	345
How CIAI Columns and Indexes Are Implemented in the Database	346

Index

1

What's New in This Release

What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2016, Rev. A

Table 1 lists the changes described in this version of the documentation to support this release of the software.

NOTE: Siebel Innovation Pack 2016 is a continuation of the Siebel 8.1/8.2 release. This guide is for upgrading to Siebel Innovation Pack 2016 (version 16.0).

Table 1. What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2016, Rev. A

Topic	Description
"Copying Ancestor Repositories" on page 130	Modified topic. Corrected the patch ID for the ancestor repositories for Siebel CRM version 16.0.
"Upgrading the Siebel Database Schema" on page 205	Modified topic. For the step Export Merge Data to CSV Script or Input File, added a reference to 2145521.1 (Article ID) on My Oracle Support, which provides information about the specific releases (including applicable patchset releases) for which Usage Pattern Tracking can be enabled.

What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2016

Table 2 lists the changes described in this version of the documentation to support this release of the software.

NOTE: Siebel Innovation Pack 2016 is a continuation of the Siebel 8.1/8.2 release. This guide is for upgrading to Siebel Innovation Pack 2016 (version 16.0).

Table 2. What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2016

Topic	Description
"Supported Upgrade Paths for Siebel CRM Version 16.0" on page 17 Multiple topics	New or modified topics. The current release for this guide is Siebel Innovation Pack 2016 (version 16.0). You can upgrade to Siebel CRM version 16.0 from applicable prior Siebel CRM releases, including Siebel CRM version 8.1.1.x, version 8.2.2.x, or version 15.x. The upgrade tasks for this release are similar to those for previous releases.
"Supported Upgrade Paths for Siebel CRM Version 15.5"	Deleted topic. This guide documents upgrades to the current version, Siebel CRM version 16.0.

Table 2. What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2016

Topic	Description
“Upgrade Planning for Siebel Web Template Files and Style Sheets” on page 73	Modified topic. Updated the information about the directory structure for the public directory on the Siebel Web Server Extension (SWSE). In Siebel Innovation Pack 2016, the webmaster directory on the Siebel Server has been removed. The virtual directories now map to public on the SWSE instead of language directories like public/ENU, and several directories, including files, are now located directly under public. For more information about these and other changes, see the <i>Siebel Installation Guide</i> for the operating system you are using.
“Upgrade Planning for Mobile Devices in the Siebel Environment” on page 79	Modified topic. As of Siebel Innovation Pack 2016, Siebel Handheld applications are no longer supported. The information presented for the current release applies to mobile devices and the Siebel Mobile applications.
“Upgrade Planning for Siebel Configurator”	Removed topic. The information in this topic no longer applies in Siebel Innovation Pack 2016, due to desupport of the high interactivity client.
“Copying Ancestor Repositories” on page 130	Modified topic. Corrected the part number, patch ID, some version number references, and other content to be consistent with cases for upgrading to Siebel Innovation Pack 2016 (version 16.0). Added the ancestor repository file r155.dat. Also merged two similar topics into one topic applicable to all repository upgrades.
“Upgrading the Siebel Database Schema” on page 205	Modified topic. Added the Export Merge Data to CSV Script or Input File step in Table 30 on page 206 . This step exports repository merge log data for comparison with Usage Pattern Tracking CSV files, to help you focus your testing efforts.
“Upgrading the Siebel Database Schema” on page 205 “Upgrading a Custom Database Schema” on page 221	Modified topics. Added the Migrate SWT from Files into the Siebel Database step out of Table 31 on page 222 and into Table 30 on page 206 in “Upgrading the Siebel Database Schema” on page 205 .
“About Migrating Siebel Web Template Data” on page 236	Modified topic. Updated the information about the directory structure for the webtempl directory. In Siebel Innovation Pack 2016, all Web templates must now be located in webtempl or webtempl/custom.
“Activating License Keys” on page 267	New topic. In Siebel Innovation Pack 2016, the license keys previously provided in seed data are inactive. A new utility is provided for activating or deactivating license keys.

What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2015, Rev. C

Table 3 lists the changes described in this version of the documentation to support this release of the software.

NOTE: Siebel Innovation Pack 2015 is a continuation of the Siebel 8.1/8.2 release.

Table 3. What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2015, Rev. C

Topic	Description
<p>"Supported Upgrade Paths for Siebel CRM Version 15.5 or Later" (removed for Siebel Innovation Pack 2016)</p> <p>"Executing Incremental Repository Merge on UNIX" on page 213</p>	<p>Modified topics. Updated Siebel Repository files (SRF files) are available for download from My Oracle Support. These SRF files correspond to repository updates in the Siebel database for Siebel Innovation Pack 2015, Patchset 5 or later (version 15.5 or later).</p> <p>It is not necessary to compile the New Siebel Repository SRF in order to run incremental repository merge or perform a full database upgrade.</p>
<p>"About Siebel Additive Schema Changes Mode" on page 46</p>	<p>Modified topic. Indexes are not created or modified by the Apply Additive Schema Changes mode.</p>
<p>"Upgrading the Siebel Database Schema" on page 205</p> <p>"Reviewing the Hierarchical Merge Report" on page 216</p>	<p>Modified topics. MergeReport.bat runs automatically for Microsoft Windows. Additional information about generating the hierarchical merge report on UNIX or Linux has been provided.</p>
<p>"Upgrading the Siebel Database Schema" on page 205</p> <p>"Upgrading a Custom Database Schema" on page 221</p>	<p>Modified topics. Added recommendations to back up the Siebel database before performing these upgrade operations.</p>
<p>"Upgrading the Siebel Database Schema" on page 205</p> <p>"Verifying the Repository After Upgrade" on page 224 (new)</p>	<p>Modified topic and new topic. Moved the description of the Verify Repository After Upgrade step out of Table 31 on page 222 and into a new topic, "Verifying the Repository After Upgrade" on page 224.</p>
<p>"Upgrading to the Siebel Symbolic String Model" on page 243</p>	<p>Modified topic. The symbolic string model applies to Siebel CRM version 7.8.2 and later.</p>

What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2015, Rev. B

Table 4 lists the changes described in this version of the documentation to support this release of the software.

NOTE: Siebel Innovation Pack 2015 is a continuation of the Siebel 8.1/8.2 release.

Table 4. What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2015, Rev. B

Topic	Description
"Supported Upgrade Paths for Siebel CRM Version 15.5 or Later" (removed for Siebel Innovation Pack 2016) Multiple topics	Modified topics. The current release for this guide is Siebel Innovation Pack 2015, Patchset 5 or later (version 15.5 or later). You can upgrade to Siebel CRM version 15.5 or later from applicable prior Siebel CRM releases, including Siebel CRM version 8.1.1.x, version 8.2.2.x, or version 15.0. The upgrade tasks for this release are similar to those for previous releases.
"Copying Ancestor Repositories" on page 130	Modified topic. Updated the My Oracle Support patch number and version number references to be consistent with cases for upgrading to Siebel Innovation Pack 2015, Patchset 5 or later (version 15.5 or later). Added ancestor repository file r150.dat.

What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2015, Rev. A

Table 5 lists the changes described in this version of the documentation to support this release of the software.

NOTE: Siebel Innovation Pack 2015 is a continuation of the Siebel 8.1/8.2 release.

Table 5. What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2015, Rev. A

Topic	Description
"Testing Your Siebel Tools Login" "Setting a Primary Non-ENU Language in Siebel Tools"	Deleted topics. These topics no longer apply, because it is not supported to run Siebel Tools for the current release before you have run incremental repository merge.
"Marking Conflict Resolution as Complete Using Siebel Tools" on page 221	This task is now part of "Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x (SIA Repository)" on page 204 .

What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2015

Table 6 lists the changes described in this version of the documentation to support this release of the software.

NOTE: Siebel Innovation Pack 2015 is a continuation of the Siebel 8.1/8.2 release.

Table 6. What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2015

Topic	Description
<p>"Supported Upgrade Paths for Siebel CRM Version 15.0" (removed for Siebel CRM version 15.5)</p> <p>Multiple topics</p>	<p>New or modified topics. Siebel Innovation Pack 2015 is provided as a single release, Siebel CRM version 15.0. You can upgrade to Siebel CRM version 15.0 from applicable prior Siebel CRM release, including Siebel CRM version 8.1.1.x or version 8.2.2.x releases. Upgrade tasks for this release are similar to those for previous releases.</p> <p>One-step upgrades from Siebel CRM version 7.5.3 or version 7.7.2 (SIA repository) are no longer supported for the current release. These upgrade cases require a two-step upgrade.</p>
<p>"Supported Upgrade Paths for Siebel CRM Version 8.1.1.x"</p> <p>"Supported Upgrade Paths for Siebel CRM Version 8.2.2.x"</p>	<p>Deleted topics. These topics are no longer needed, because this guide documents upgrades to the current version, Siebel CRM version 15.0 (now 16.0).</p>
<p>"Copying Ancestor Repositories" on page 130</p>	<p>Modified topic. Corrected the part number, some version number references, and other content to be consistent with cases for upgrading to Siebel Innovation Pack 2015.</p> <p>Removed ancestor repository files p753.dat and r77.dat. Added ancestor repository file r2014.dat.</p>

Table 6. What's New in Siebel Database Upgrade Guide, Siebel Innovation Pack 2015

Topic	Description
<p>"Upgrading a Custom Database Schema" on page 221</p>	<p>Modified topic. Added new steps to Table 31, which identifies steps that the upgphys process executes. New steps for Siebel Innovation Pack 2015 are as follows:</p> <ul style="list-style-type: none"> ■ Migrate data from Siebel Web template files into the Siebel database. <p>For more information about this step, see "About Migrating Siebel Web Template Data" on page 236.</p> <ul style="list-style-type: none"> ■ Create list import package for Siebel Marketing <p>This step creates a package to support the Siebel Marketing list import functionality for Oracle Database 12c. For more information, see <i>Siebel Marketing Installation and Administration Guide</i>.</p>
<p>"About Migrating Siebel Web Template Data" on page 236</p>	<p>New topic. In Siebel Innovation Pack 2015 and later, the upgphys process executes a script that migrates data from Siebel Web template files into table-based content in the Siebel database.</p> <p>The migrated Siebel Web template data is used by Siebel CRM Composer, which is available for preview purposes in the current release. For more information about Siebel CRM Composer and Siebel Web template migration, see <i>Siebel CRM Composer Developer Preview Guide</i>.</p> <p>NOTE: For Siebel Innovation Pack 2015, your existing deployment and usage of Siebel Web templates are unaffected by this migration.</p>

2

Overview of Siebel Database Environments

This chapter provides an overview of the upgrade process of the three Siebel database environments. It includes the following topics:

- [Supported Upgrade Paths for Siebel CRM Version 16.0 on page 17](#)
- [Terms Used in This Guide on page 21](#)
- [Naming Conventions Used in This Guide on page 21](#)
- [About File Paths and Commands in Siebel Database Upgrade Topics on page 22](#)
- [About Supported Siebel Upgrade Paths on page 24](#)
- [About Siebel Upgrade Environments on page 24](#)

NOTE: For a detailed overview of the upgrade process and related tasks, see [Appendix B, “Overview of Performing a Siebel Database Upgrade.”](#)

Supported Upgrade Paths for Siebel CRM Version 16.0

[Table 7](#) lists supported upgrade paths. The current release for this guide is Siebel Innovation Pack 2016 (Siebel CRM version 16.0). The repository for the current release is SIA in all cases. For more information on terminology, see [“Terms Used in This Guide” on page 21](#).

Updated Siebel Repository files (SRF files) are provided as part of Siebel Innovation Pack 2016. These SRF files correspond to repository updates in the Siebel database for the current release. The updated SRF files are available for all of the supported Siebel languages. Use the updated SRF files after you install Siebel Innovation Pack 2016, for either of the following upgrade tasks summarized in this topic. (It is not necessary to compile the New Siebel Repository SRF.)

- If you run incremental repository merge for an existing Siebel database.
- If you perform a full database upgrade for an existing Siebel database.

You also require the updated SRF files to run Siebel Business Applications after you install a new Siebel database for Siebel CRM version 16.0. For more information, see the *Siebel Installation Guide* for the operating system you are using.

NOTE: After installing the Siebel CRM software for the current release, you must reset any passwords in the siebns.dat file on the Siebel Gateway Name Server that were previously encrypted using RC4 encryption. In the current release, such passwords are encrypted using Advanced Encryption Standard (AES) instead of RC4. You must perform this task for any type of upgrade from versions prior to Siebel Innovation Pack 2014. For more information about reencrypting passwords, see *Siebel Security Guide*.

Table 7 lists the supported upgrade paths for Siebel CRM version 16.0.

Table 7. Supported Upgrade Paths for Siebel CRM Version 16.0

Current Version	Target Upgrade Version	Upgrade Approach	Upgrade Tasks
Siebel CRM version 7.5.3 through Siebel CRM version 7.7.2 (SEA or SIA repository)	Siebel CRM version 16.0	<ul style="list-style-type: none"> ■ New installation of Siebel Innovation Pack 2016 (version 16.0) for upgrade. ■ Full database upgrade. ■ Two-step repository upgrade. 	<p>You must perform a two-step repository upgrade:</p> <ol style="list-style-type: none"> 1 Upgrade Siebel CRM version 7.5.3 through Siebel CRM version 7.7.2 (SEA or SIA repository) to Siebel CRM version 8.1.1 (SEA or SIA repository). 2 Upgrade Siebel CRM version 8.1.1 (SEA or SIA repository) to Siebel CRM version 16.0 or later. <p>NOTE: Use incremental repository merge for upgrades from Siebel CRM version 8.1.1 (SIA repository).</p>
Siebel CRM version 7.8.2 (SEA or SIA repository)	Siebel CRM version 16.0	<ul style="list-style-type: none"> ■ New installation of Siebel Innovation Pack 2016 (version 16.0) for upgrade. ■ Full database upgrade. ■ Single-step repository upgrade. 	<ol style="list-style-type: none"> 1 Perform a full database upgrade. A Development environment upgrade and a Production environment upgrade are required.
Siebel CRM version 8.0 (SEA or SIA repository)	Siebel CRM version 16.0		<ol style="list-style-type: none"> 2 Upgrade directly to Siebel CRM version 16.0. It is a single-step repository upgrade and post-upgrade process.
Siebel CRM version 8.1.1.0 through 8.1.1.7 (SEA repository)	Siebel CRM version 16.0		<p>The New Customer Repository, generated through a three-way repository merge, contains all of the innovation pack content from Siebel CRM version 16.0.</p> <p>For more information about repository merge, see Chapter 12, "Performing the Siebel Repository Merge."</p>

Table 7. Supported Upgrade Paths for Siebel CRM Version 16.0

Current Version	Target Upgrade Version	Upgrade Approach	Upgrade Tasks
Siebel CRM version 8.1.1.0 through version 8.1.1.14 (SIA repository)	Siebel CRM version 16.0	<ul style="list-style-type: none"> ■ Migration installation of Siebel Innovation Pack 2016 (version 16.0). ■ Incremental repository merge. 	<ul style="list-style-type: none"> ■ Use incremental repository merge to bring the repository to version 16.0 or later. <p>NOTE: This step is mandatory to be able to run Siebel Business Applications in the current release.</p> <p>For more information, see Chapter 13, “Performing the Siebel Incremental Repository Merge.”</p>
Siebel CRM version 8.2 (SIA repository)	Siebel CRM version 16.0	<ul style="list-style-type: none"> ■ New installation of Siebel Innovation Pack 2016 (version 16.0) for upgrade. 	<ol style="list-style-type: none"> 1 Perform a full database upgrade. A Development environment upgrade and a Production environment upgrade are required.
Siebel CRM version 8.2.1 (SIA repository)	Siebel CRM version 16.0	<ul style="list-style-type: none"> ■ Full database upgrade. ■ Single-step repository upgrade. 	<ol style="list-style-type: none"> 2 Upgrade directly to Siebel CRM version 16.0. It is a single-step repository upgrade and post-upgrade process. <p>The New Customer Repository generated through a three-way repository merge contains all of the innovation pack content of Siebel CRM version 16.0.</p> <p>For more information about repository merge, see Chapter 12, “Performing the Siebel Repository Merge.”</p>

Table 7. Supported Upgrade Paths for Siebel CRM Version 16.0

Current Version	Target Upgrade Version	Upgrade Approach	Upgrade Tasks
Siebel CRM version 8.2.2.0 through version 8.2.2.14 (SIA repository)	Siebel CRM version 16.0	<ul style="list-style-type: none"> ■ Migration installation of Siebel Innovation Pack 2016 (version 16.0). ■ Incremental repository merge. 	<ul style="list-style-type: none"> ■ Use incremental repository merge to bring the repository to version 16.0. <p>NOTE: This step is mandatory to be able to run Siebel Business Applications in the current release.</p> <p>For more information, see Chapter 13, “Performing the Siebel Incremental Repository Merge.”</p>
Siebel CRM version 15.0 through version 15.4	Siebel CRM version 16.0	<ul style="list-style-type: none"> ■ Migration installation of Siebel Innovation Pack 2016 (version 16.0). ■ Incremental repository merge. 	<ul style="list-style-type: none"> ■ Use incremental repository merge to bring the repository to version 16.0 or later. <p>NOTE: This step is mandatory to be able to run Siebel Business Applications in the current release.</p> <p>For more information, see Chapter 13, “Performing the Siebel Incremental Repository Merge.”</p>
Siebel CRM version 15.5 and later	Siebel CRM version 16.0	<ul style="list-style-type: none"> ■ Migration installation of Siebel Innovation Pack 2016 (version 16.0). ■ Incremental repository merge. 	<ul style="list-style-type: none"> ■ Use incremental repository merge to bring the repository to version 16.0 or later. <p>NOTE: This step is mandatory to be able to run Siebel Business Applications in the current release.</p> <p>For more information, see Chapter 13, “Performing the Siebel Incremental Repository Merge.”</p>

Terms Used in This Guide

This guide uses the following terms:

Database Upgrade. A set of standard procedures to migrate your custom repository and data schema from one release of Siebel CRM to a higher release level (that is, to a later release).

Siebel Innovation Pack and Siebel Patchset releases. Siebel Innovation Pack releases are installed either as new installations or as migrations from earlier releases of Siebel CRM version 8.1.1.x, version 8.2.2.x, or version 15.x. Siebel Patchset releases are installed as patch installations into your Siebel Innovation Pack installation. In some cases, a Siebel Patchset release might add optional new feature content. This guide is for upgrading to Siebel Innovation Pack 2016.

For more information about installing the current Siebel Innovation Pack release and about Siebel release types, see the *Siebel Installation Guide* for the operating system you are using. For more information about installing Siebel Patchset releases, including new features, see *Siebel Patchset Installation Guide for Siebel CRM*, 1614310.1 (Article ID) on My Oracle Support.

SEA and SIA. The Siebel Industry Applications (SIA) data model is physically a superset of the Siebel Enterprise Application (SEA) data model. The SIA data model has more tables, more columns in the same tables, and more indexes than the SEA data model, but it does not exclude any tables, columns or indexes from the SEA version. Current versions of Siebel CRM now support only the SIA data model. Some upgrade paths might involve upgrading from an SEA data model to the SIA data model; additional steps apply.

Incremental Repository Merge. A mechanism which allows you to incrementally upgrade your custom repository data (including schema and seed data) from Siebel CRM version 8.1.1.x (SIA only), Siebel CRM version 8.2.2.x, or Siebel CRM version 15.x to Siebel CRM version 16.0. For more information about the incremental repository merge process, see [Chapter 13, "Performing the Siebel Incremental Repository Merge."](#)

NOTE: The Siebel CRM version 16.0 repository contains the cumulative repository, schema, and seed data for all new content (such as Release Features) developed up to Siebel CRM version 16.0.

Naming Conventions Used in This Guide

This guide follows several naming conventions:

- All supported Siebel CRM releases refer to the upgrade path from which you might upgrade to a certain Siebel Business Applications release.

- For more information about your upgrade applicability and for system requirements and supported platform certifications, see the Certifications tab on My Oracle Support.

NOTE: For Siebel CRM version 8.1.1.9 and later and version 8.2.2.2 and later, the system requirements and supported platform certifications are available from the Certifications tab on My Oracle Support. For more information, see article 1492194.1 (Article ID) on My Oracle Support. For earlier Siebel CRM versions, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

- *Current release* means the latest version of Siebel Business Applications.

- The term *Windows* refers to all Microsoft Windows operating systems listed as supported for this release in the Certifications tab on My Oracle Support.
- The term *UNIX* refers to all forms of the UNIX operating system listed as supported for this release in the Certifications tab on My Oracle Support.
- The term *IBM z/OS* refers to the IBM mainframe operating systems that are collectively referred to as z/OS and listed as supported for this release in the Certifications tab on My Oracle Support.

About File Paths and Commands in Siebel Database Upgrade Topics

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

Windows Paths

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

- *SIEBEL_ROOT* is the absolute path of the Siebel Server installation directory. When you install a Siebel Server, the installation program queries for the path to the installation directory. The installation program then installs the Siebel Server in a subdirectory of this path called *si ebsvr*. For example, if you specified C: \sba81 as the installation directory for Siebel CRM 8.1, then *SIEBEL_ROOT* is C: \sba81\si ebsvr.
- *DBSRVR_ROOT* is the absolute path to the Siebel Database Configuration Utilities on the Siebel Server. When you install the Siebel Database Server, the installation program 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 *dbsvr*. For example, if *SIEBEL_ROOT* is C: \sba81\si ebsvr, then *DBSRVR_ROOT* is C: \sba81\dbsvr. In this guide, many examples use the path C: \ and C: \sba81.
- *SIEBEL_TOOLS_ROOT* is the directory in which Siebel Tools is installed, on a developer computer running Microsoft Windows.

UNIX Paths

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

- **SIEBEL_ROOT** is the absolute path of the Siebel Server installation directory and also an environment variable that defines this path. 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 `si ebsrvr`. For example, if you specified `/usr/si ebel` as the installation directory for Siebel CRM version 8.x, then `$SIEBEL_ROOT` is `/usr/si ebel /sba81/si ebsrvr`.

`SIEBEL_ROOT` and other environment variables required for doing an upgrade are located in `si ebsrvr/si ebenv. sh` and `si ebsrvr/si ebenv. csh`. The Siebel Server installation script sets environment variable definitions in these shell scripts. Do not edit or delete these files.

TIP: Before performing command-line procedures in a shell window, you must source `si ebenv. csh` for a C shell or source `si ebenv. sh` for a Bourne shell. Doing this refreshes the environment variables required to run commands.

- **DBSRVR_ROOT** is a path convention used in the *Siebel Database Upgrade Guide*. It is not an environment variable and is not defined in `si ebenv. csh` or `si ebenv. 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/si ebel /sba8x/si ebsrvr`, then `DBSRVR_ROOT` is `/usr/si ebel /sba8x/dbsrvr`. In this guide, many examples use the path `/usr/si ebel` and `/usr/si ebel /sba8x`.

- Run UNIX scripts in a C or Korn shell. Do not run `.ksh` scripts in a Bourne shell.

Path Navigation

Procedural steps that ask you to navigate to a specified directory must 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 press Enter.
- UNIX: In a shell window, make the specified directory the current directory.

Executing Commands

Procedural steps that ask you to execute a command must 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 on UNIX, then treat all file names, directory names, path names, parameters, flags, and command-line commands as lowercase, unless you are instructed otherwise in the product.

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

About Supported Siebel Upgrade Paths

The supported upgrade paths for Siebel Business Applications are described in [“Supported Upgrade Paths for Siebel CRM Version 16.0” on page 17](#) and in the Certifications tab on My Oracle Support.

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

- Changing database platform type during an upgrade, for example changing from IBM DB2 to Oracle Database.
- 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.
- 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

There are three Siebel database environments: Development, Production Test, and Production. If you have a development environment, then you must upgrade it first. When you upgrade your development environment your previous customizations are merged with the new release. The newly merged repository and schema definitions then become inputs to the production test environment upgrade and production upgrades which you will perform after completing the development environment upgrade.

If you do not have a development environment, or have not customized your repository, then 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.

Mapping Your Environments

You might have more or fewer environments than those described previously. [Table 8](#) gives recommendations for mapping your environments to the ones used in *Siebel Database Upgrade Guide*.

Table 8. Mapping Your Environments to Upgrade Processes

Environment Description	Recommended Upgrade
<p>The environment has the following characteristics:</p> <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 Siebel Mobile Web Clients. 	Development environment upgrade.
<p>The environment has the following characteristics:</p> <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 might 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.
<p>The environment is used for live business transactions by both local and remote users.</p>	Production environment upgrade.

Siebel upgrade environments are described in detail in the following topics:

- [“Development Environment” on page 25](#)
- [“Production Test Environment” on page 26](#)
- [“Production Environment” on page 27](#)

Development Environment

This environment is where developers use Siebel Tools to customize Siebel Business Applications, and where customizations are compiled in a Siebel Repository File (SRF). When your Siebel Business Applications instance goes live, the SRF is where all object definitions are retrieved, then displayed to users in your Siebel CRM user interface.

A development environment contains the following Siebel CRM modules:

- Siebel Server
- Web server and Siebel Web Server Extension (SWSE)

- 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' Siebel Mobile Web Clients.
- Siebel Business Applications and test data required to verify the basic function of newly compiled SRF files.

Upgrading the Development environment involves these tasks:

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

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

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

Production Test Environment

This environment is where an installed and fully upgraded development environment undergoes the quality assurance process. This part of your upgrade process allows you to tune upgrade performance in preparation for the transition to your live, production environment.

The production test environment includes the following Siebel CRM 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 Business 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.

Upgrading the Production Test environment involves the following tasks:

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

Production Environment

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

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

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

- **Siebel Application Deployment Manager (ADM).** This module migrates administrative data such as lists of values (LOVs) from the production test environment to the production environment. For details, see *Siebel Application Deployment Manager Guide*.
- **Siebel Anywhere.** This application builds distribution kits for remote users. For details, see *Siebel Anywhere Administration Guide*.

Upgrading the Production environment involves the following tasks:

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

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

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

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

Postupgrade Tasks

Following the upgrade you perform the following tasks:

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

3

Siebel Database Upgrade Planning

This chapter provides information on planning your Siebel database upgrade. It includes the following topics:

- [About Siebel Upgrade Planning Resources on page 29](#)
- [Guidelines for Planning Your Siebel Database Upgrade on page 32](#)
- [About Upgrading Your RDBMS in the Siebel Environment on page 35](#)
- [About Database Sort Order in the Siebel Environment on page 36](#)
- [About the Siebel Database Configuration Utilities on page 37](#)
- [About the Siebel Upgrade Wizard and Driver Files on page 43](#)
- [About Siebel Additive Schema Changes Mode on page 46](#)
- [About the Siebel Database Upgrade Log Files on page 48](#)
- [About the Siebel Case Insensitivity Wizard on page 50](#)
- [About the Siebel Repository Merge on page 51](#)
- [About Single-Step Repository Upgrade on page 56](#)
- [About Inheriting Upgrade Behavior in a Siebel Upgrade on page 57](#)
- [About the Siebel Postmerge Utilities on page 59](#)
- [About Tuning Siebel Production Upgrade Files on page 63](#)

NOTE: For a detailed overview of the upgrade process and related tasks, see [Appendix B, "Overview of Performing a Siebel Database Upgrade."](#)

About Siebel Upgrade Planning Resources

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

Product Documentation

Oracle's Siebel CRM documentation is collectively called the Siebel Bookshelf. The Siebel Bookshelf is available on Oracle Technology Network (OTN) and Oracle Software Delivery Cloud. It might also be installed locally on your intranet or on a network location.

Siebel Installation and Deployment Documentation

The following publications are meant to be used in concert with installation or deployment of the Siebel application and can be found on the *Siebel Bookshelf*:

- *Siebel Deployment Planning Guide*
- *Siebel Installation Guide for Microsoft Windows*
- *Siebel Installation Guide for UNIX*
- *Siebel System Administration Guide*. Provides details on how to administer, maintain, and expand your Siebel Servers.
- *Siebel Security Guide*. Provides information on the Siebel Party data model and other important security topics.
- *Siebel Application Deployment Manager Guide*
- *Siebel Performance Tuning Guide*
- *Configuring Siebel Business Applications*. Provides information about configuring Siebel Business Applications in Siebel Tools. Additionally, this guide provides information on the Siebel Party data model.
- *Siebel Data Model Reference* on My Oracle Support. 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 and the release you are upgrading to. Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services for information on ordering DMRs.

Oracle Technology Network

This site contains the following:

- ***Siebel Bookshelf***. A searchable collection of Oracle's Siebel CRM documentation.
- ***Siebel System Requirements and Supported Platforms*** on Oracle Technology Network.
NOTE: For releases prior to Siebel CRM version 8.1.1.9 or version 8.2.2.2, *Siebel System Requirements and Supported Platforms* on Oracle Technology Network is the definitive list of system requirements and supported third-party products. For all subsequent releases, the system requirements and supported platform certifications are available from the Certifications tab on My Oracle Support. For information about the Certifications application, see article 1492194.1 (Article ID) on My Oracle Support.

My Oracle Support

This site provides the Certifications tab and provides search access to articles, Release Notes, and various other kinds of documentation relevant to Siebel CRM:

- ***Release Notes***. *Siebel Release Notes* on My Oracle Support might contain late-breaking information that the *Siebel Database Upgrade Guide* does not yet include.

- **The Certifications tab on My Oracle Support.** This application is the definitive list of system requirements and supported third-party products. For more information, see article 1492194.1 (Article ID) on My Oracle Support.

- **Articles.** Articles provide important information on specific upgrade issues. Many articles related to upgrade issues are located on My Oracle Support.

477833.1 (Article ID) on My Oracle Support (previously published as Siebel Alert 1002) and 478177.1 (Article ID) on My Oracle Support (previously published as Siebel Alert 1179) are particularly important to all upgrades. References to these and other articles are integrated throughout *Siebel Database Upgrade Guide*.

For Siebel language support, Unicode support, and legacy code page support, see 1513102.1 (Article ID) on My Oracle Support.

- **Troubleshooting.** Troubleshooting documents contain information about how to troubleshoot common upgrade issues such as error messages encountered during the upgrade process, and other unwanted behavior in Oracle's Siebel Business Applications. Troubleshooting documents can be found for a variety of upgrade issues, including error messages found in upgrade logs. For more information browse Troubleshooting content on My Oracle Support.

- **Siebel Patchset releases.** Siebel Patchset releases and documentation are provided on My Oracle Support.

For the readme documents for Siebel Patchset releases, see *Siebel Patchset Installation Guide for Siebel CRM*, 1614310.1 (Article ID) on My Oracle Support. For more information about Siebel Patchset releases, see 1600520.1 (Article ID) on My Oracle Support.

- ***Siebel Maintenance Release Guide on My Oracle Support.***

For each Siebel CRM version 8.1.1.x and version 8.2.2.x release, through Siebel CRM version 8.1.1.11 and version 8.2.2.4 (Siebel Innovation Pack 2013), *Siebel Maintenance Release Guide* on My Oracle Support was provided. This guide contained important information about updates to Siebel Business Applications. These guides are no longer provided for current releases. The versions of this guide for applicable releases are provided here:

- Version 8.1.1.x *Siebel Maintenance Release Guide* on My Oracle Support, 880452.1 (Article ID)
- Version 8.2.2.x *Siebel Maintenance Release Guide* on My Oracle Support, 1441523.1 (Article ID)

Oracle Software Delivery Cloud

This site provides access to Siebel CRM software, such as the current release, Siebel Innovation Pack 2016 (version 16.0). For more information about types of Siebel CRM releases, see the *Siebel Installation Guide* for the operating system you are using.

Oracle Advanced Customer Services

Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services. These services offer detailed implementation planning and technical consulting services as well as rapid response and resolution for critical technical issues affecting Siebel deployments.

Contact your Oracle sales representative for Oracle Advanced Customer Services for important information in the following areas:

- Migrating from Siebel Industry Solutions or Siebel Financial Services to Siebel Business Applications during upgrade
- Migrating to Unicode code page support during upgrade
- Changing operating system type during upgrade

Guidelines for Planning Your Siebel Database Upgrade

This topic provides an overview of the recommended guidelines for planning upgrade resources, estimating the upgrade timeline, and managing the data migration process.

Use the following steps to help plan your upgrade.

- 1 Determine your upgrade path.** Upgrade requirements and tasks differ based on the specific upgrade path involved. For example, when you install Siebel Innovation Pack 2016 as a migration from an earlier version of Siebel CRM version 8.1.1 or 8.1.1.x, Siebel CRM version 8.2.2 or 8.2.2.x, or Siebel CRM version 15.0 or 15.x, then you run incremental repository merge, which is a relatively new upgrade technique. Supported upgrade paths from earlier Siebel CRM versions involve traditional upgrade techniques. For more information, see ["Supported Upgrade Paths for Siebel CRM Version 16.0" on page 17](#).
- 2 Evaluate the complexity of the upgrade.** Determine the complexity of the upgrade effort based on the Siebel CRM modules implemented, the number of integration points, the number of interfaces, the total number of scripts, and the 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 guidelines. 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 might 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 factors 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 guidelines identified in this guide

The upgrade planning process will produce a roadmap for the entire upgrade project that outlines infrastructure, deployment, and training requirements.

Use the results of this process to develop a project plan that identifies required skills and resources for developing and deploying the upgraded application. This will help you with advance budgeting of resources, time, and training.

Upgrade Planning Guidelines

Here are important guidelines to follow when planning an upgrade:

- Review the Certifications tab on My Oracle Support, *Siebel Release Notes* on My Oracle Support, *Siebel Installation Guide* for the operating system you are using, and other relevant documentation. Also review the content of this guide, *Siebel Database Upgrade Guide*.
- 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 Guidelines

Here are important guidelines to follow when planning the upgrade of your database:

- Analyze the impact of the upgrade on table customizations that you have made. Determine whether preupgrade data migration is required. Determine what postupgrade schema changes are required.
- 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 Online Transaction Processing (OLTP).
- If you are migrating multiple languages from a prior version, then plan extra time (one to two weeks) for the repository merge process. The expected merge time might increase with the number of languages in the repository. You also might need to plan for additional installation-related tasks.
- If your locale specific language values have changed over time, then you must make changes in the `upglocale.language_code` file to trigger correct data migrations. For example, in a previous German language release, the German value *Einzelperson* mapped to the ENU value of Individual. In the new release *Individuell* and not *Einzelperson* maps to the ENU value Individual. Prior to running the upgrade utility, review the `upglocale.language_code` files from your present release to that of the upgrade release, to validate, and verify your language strings and to make any necessary changes. `Upglocale.language_code` files are located at `SI EBEL_ROOT/dbsrvr/language_code` folder in the new release's installation directory.
- For IBM DB2, 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 93.

Production Database Upgrade Guidelines

Both the Prepare for Production step and the upgrade tuning process modify the SQL scripts used to perform the upgrade in the production test environment. In addition, it is common to further modify these scripts to meet local requirements.

The recommended way to perform the production upgrade is to use the SQL scripts that you have generated and modified for the production test upgrade. The steps in the production environment upgrade process are as follows:

- Run the Database Configuration Utilities in the production environment.
- In the utilities, enter the information for the production environment instead of the production test environment. For example, you enter the ODBC connection for the production environment.

This configures the driver file to run against the production database rather than the production test database. The utilities also configure the driver file to use the upgrade SQL files you generated for the production test upgrade.
- Run the Upgrade Wizard. The Upgrade Wizard uses the SQL files in the production test environment to upgrade the database in the production environment.

This approach has several advantages:

- You do not have to generate upgrade SQL files in the production environment and then manually transfer customizations to them from the production test environment.
- You do not lose any changes to the SQL files that were made by Siebel Upgrade Tuner in the production test environment.
- You do not have to run the Database Configuration Utilities in Prepare for Production mode again.
- With some exceptions, you do not have to perform database-related configuration tasks required Articles or *Siebel Release Notes* on My Oracle Support.

If your network configuration prevents creating an ODBC connection to your production database from inside your production test environment, and you cannot complete your production upgrade, then create a service request (SR) on My Oracle Support. Alternatively, you can phone Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers remain the same and are listed on My Oracle Support.

About Upgrading Your RDBMS in the Siebel Environment

Environments: Development, production test, production.

If your currently installed RDBMS version is not supported in the new release, then you must upgrade your RDBMS before performing the Siebel database upgrade. For information on supported RDBMS versions, see the Certifications tab on My Oracle Support.

If your currently installed RDBMS version is supported in the new release, and you plan to upgrade your RDBMS, then 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 the Certifications tab on My Oracle Support.

Oracle Database

The following guidelines and requirements are for planning your upgrade on Oracle Database.

- See Article ID 477966.1 on My Oracle Support (previously published as Siebel Alert 1053) for planning information on upgrading an Oracle RDBMS. If you upgrade the Siebel database and then upgrade the Oracle RDBMS, then you must validate the Siebel database schema after the upgrade. See [“Verifying an Upgraded Oracle Database After a Siebel Upgrade” on page 249](#).
- Development, production test, and production upgrades on Oracle Database 11g, can be run in CBO mode.

- If you plan to upgrade your Oracle Database, then see 781927.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 582. This document provides information on setting up Oracle Database for use with Oracle Cost-based optimizer (CBO) and on doing other performance tuning for Oracle Database. Using Oracle's Cost Based Optimization with a Siebel database entails certain requirements for database parameter configuration and for the Optimizer's statistics generation on tables, columns and indexes.

IBM DB2

The following guidelines and requirements are for planning your upgrade on IBM DB2.

- After upgrading your IBM DB2 RDBMS, you must upgrade the database instance.
- The IBM DB2 database must have 4-KB, 16-KB, and 32-KB tablespaces defined on it. Otherwise, your upgrade will not complete successfully.

About Database Sort Order in the Siebel Environment

Environments: Development, production test, production.

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. For Siebel language support, Unicode support, and legacy code page support, see 1513102.1 (Article ID) on My Oracle Support.

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), then you must consider several functional limitations that particularly affect development environment upgrades. If these limitations are unacceptable, then 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.

For Siebel language support, Unicode support, and legacy code page support, see 1513102.1 (Article ID) on My Oracle Support.

About the Siebel Database Configuration Utilities

Environments: Development, production test, production.

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

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

About the Upgrade the Siebel Database Function

When you run the Upgrade the Database function of the Database Configuration Wizard, the wizard prompts you for the upgrade environment (development or production) and the upgrade phase (upprep or upgphys). The wizard then prompts you for the information required to perform the upgrade.

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

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

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

Table 9. Database Configuration Utilities Modes

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

Figure 1 shows how the Database Configuration Wizard (and Upgrade Wizard) work together with the Siebel Tools repository merge to upgrade your environments.

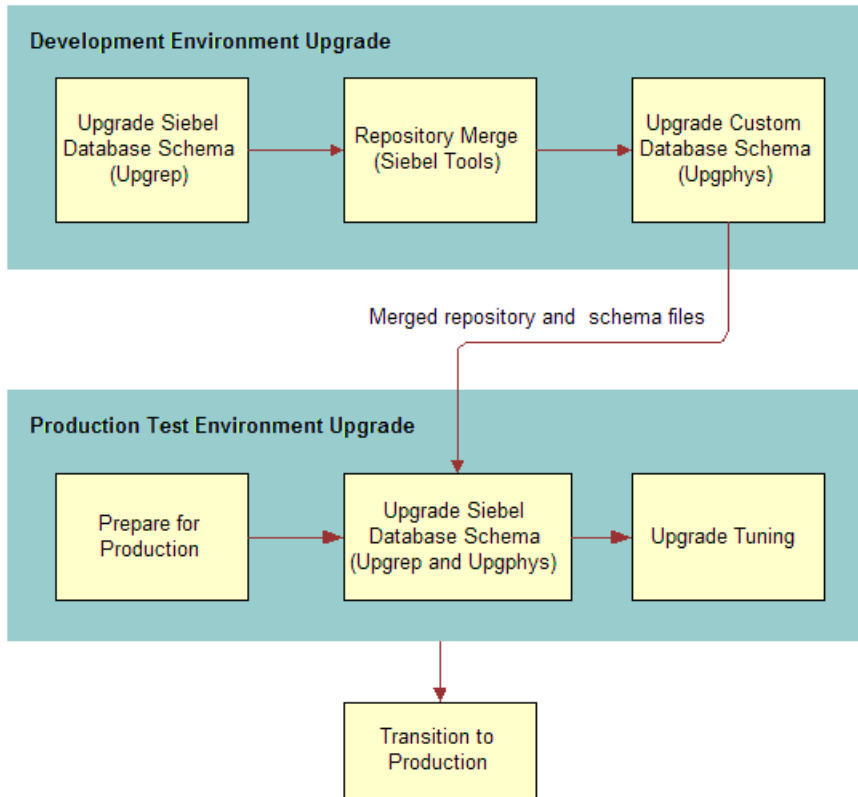


Figure 1. Upgrade Flow

Development Environment Upgrades

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

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

- Creates indexes.

NOTE: Indexes that have been previously deactivated will be recreated during the upprep phase. Check to determine whether these indexes are required following the development upgrade. If they are not, then they might be deactivated again. To deactivate indexes, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

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

Production Test Environment Upgrades

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

- **Prepare for production.**

NOTE: This topic is only applicable for upgrades from Siebel CRM release 7.5.3. For upgrades from later releases, you do not have to perform this step.

This mode does the following in the production test database:

- Examines the development environment database and generates SQL that deduplicates intersection tables and sets up support for database aggregation.
In the production test environment, you must define an ODBC connection to the development environment database before performing this upgrade step.
- Compares the repository schema and the physical database schema. Generates a file that lists indexes present in the physical schema that are not present in the repository schema. You can decide whether to remove these indexes.

- **upgrep.** This mode makes the following changes:
 - Removes interface tables and database triggers
 - Populates columns that must change from NULL to NOT NULL
 - Uses the custrep.dat and schema.ddl files from the development environment upgrade to create new tables and merge existing tables
 - Prepares for index creation. Verifies that there are no unique key violations.
 - Creates indexes
 - Imports seed data
 - Updates primary children foreign key references
 - Performs miscellaneous file actions
 - The upgphys portion runs automatically. It makes several administrative changes to table data, including updating the schema version in S_APP_VER. It does not make schema changes.
- **Upgrade Tuning.** These steps are optional. You can use the Logparse utility to tune the SQL generated for the upgrade. Logparse identifies SQL commands that are not used, or that can be run in parallel. This reduces the time required to perform the production database upgrade.

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

Production Environment Upgrades

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

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

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

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

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

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

How the Upgrade Configuration File and SQL Files Are Created

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

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

In some cases, you must modify the generated SQL files as required by Articles, or *Siebel Release Notes* on My Oracle Support before you run the Upgrade Wizard. To do this, answer No when prompted to run the Upgrade Wizard. Then, edit the SQL files and manually launch the Upgrade Wizard.

How to Locate Master Configuration Files

Master upgrade configuration files are stored in the following location:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

Master upgrade configuration files use the following naming convention:

`master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`

where:

- `UPGRADEOPTION` indicates the upgrade process you are performing
 - `upgrep` indicates Siebel database schema upgrade
 - `upgphys` indicates Custom database schema upgrade
 - `prepare_for_production_upgrade` indicates Prepare for Production upgrade

- *ENVIRONMENT* indicates the environment that you are upgrading
 - dev upgrades Development environment
 - prod upgrades Production environment
- *VERSION* indicates the version from which you are upgrading

For example, if you are upgrading from Siebel CRM version 8.1.1, then the master configuration file generated from the development environment upgrep is as follows:

```
master_upgrep_dev_811.ucf
```

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

```
master_prepare_for_production_upgrade.ucf
```

Related Topics

[“About Siebel Additive Schema Changes Mode” on page 46](#)

[“About the Siebel Upgrade Wizard and Driver Files” on page 43](#)

About the Siebel Upgrade Wizard and Driver Files

Environments: Development, production test, production.

The Upgrade Wizard performs the upgrade. It uses the information collected by the Database 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_811.ucf`
- `driver_upgphys_dev_811.ucf`
- `driver_upgrep_prod_811.ucf`

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

```
[Sql File Entry 0]
Type = DbSql
File Name = rename_existing_repositories.sql
```

Use Table Owner = 1
Use System Admin = 0
IgnoreAllDDLErrors = 0
IgnoreAllDMLErrors = 0
Argument 0 = \$SiebelVersion
Title = Verify Repository Name
Title Message Num = 0
Estimated Disk Space = 0
Backup Db = 0
Parallelizable Item = 0
Prompt User For Status = 0

[File Execute Entry 1]

Type = FileExecute
File Name = \$SiebelRoot\bin\odbcsql
Check Return Code = 1
Return Code Compliance = 0
16 Bit App = 0
Command Line = /s "\$ODBCDataSource" /u \$TableOwner /p \$TablePassword /separator / /
a /c rem /l \$SiebelLogDir/dropif-db.log \$DbsrvrRoot/\$DatabasePlatform/dropif-db.sql
/v
Number of 10 Second Wait Loops = 2000
Return Code = 0
Title = Drop interface tables
Title Message Num = 0
Estimated Disk Space = 0
Backup Db = 0
Parallelizable Item = 0
Prompt User For Status = 0

[File Execute Entry 2]

Type = FileExecute
File Name = \$SiebelRoot\bin\odbcsql
Check Return Code = 1
Return Code Compliance = 0
16 Bit App = 0
Command Line = /s "\$ODBCDataSource" /u \$TableOwner /p \$TablePassword /separator / /
a /c rem /I \$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_Number*
- UNIX: *DBSRVR_ROOT/bin/PLATFORM/UPGRADE/Version_Number*

where:

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

For example, if you are upgrading from Siebel CRM version 8.1, then the driver file for the development environment upgrep will be:

driver_upgrep_dev_81.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 Configuration Utilities” on page 37](#)

[“About the Siebel Repository Merge” on page 51](#)

About Siebel Additive Schema Changes Mode

Environments: Production test, production.

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

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

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

Types of Changes

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

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

Implementation of Additive Schema Changes

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

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

The Upgrade Wizard then performs the following steps:

- Reads the `master_additive_gen.ucf` file to obtain environment information and the name of the driver file.
- Reads the `driver_additive_gen.ucf` file and executes the `ddlmp2` command it contains.
- The `ddlmp2` command performs the following steps:
 - Reads the tables and indexes from the `schema.ddl` file.

- Connects to the specified database and reads the existing schema and indexes.
- Identifies the upgrade schema changes that can be made with the database online.
- Writes these upgrade changes to the `schema.additive.sql` script.
- Writes all actions to `schema.additive.log`.

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

How to Use Additive Schema Changes

Use the following workflow to implement Additive Schema Changes:

- 1** In the production test environment, complete all development and upgrade testing. The `schema.ddl` file must be in its final form.
- 2** As part of running the Tuning Upgrade Performance process, restore the production test environment database to its original, unupgraded state.
- 3** Run the Database Configuration Utilities in Apply Additive Schema Changes mode, and generate the `schema.additive.sql` script.
- 4** **Windows.** You will not be prompted whether you want to run the Upgrade Wizard. Instead, the Upgrade Wizard starts automatically, and creates the `schema.additive.sql` script. The Upgrade Wizard does not run the script against the database.
- 5** **UNIX.** After the Database Configuration Utilities exit, run the Upgrade Wizard to generate the `schema.additive.sql` script. The Upgrade Wizard does not run the script against the database.
- 6** Review the `schema.additive.log` file. Verify that the script was created successfully.
- 7** Review the `schema.additive.sql` script. Verify that it will not make changes that conflict with customizations. Edit the script as required.
- 8** Use an SQL editor to apply the `schema.additive.sql` script to the database.
- 9** Verify that applications function normally. If necessary, revise `schema.additive.sql` and reapply it to the database.
- 10** Complete the remainder of the Tuning Upgrade Performance process. See [“Process of Tuning Siebel Upgrade Performance” on page 320](#).
- 11** When you are ready to upgrade the production database, run the `schema.additive.sql` script against the production database before doing the production upprep.
- 12** There is no need to then run database statistics. The schema changes are not useful for optimizing applications function before the upgrade.

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

Related Topics

[“About the Siebel Database Configuration Utilities” on page 37](#)

About the Siebel Database Upgrade Log Files

Environments: Development, production test, production.

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

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

Upgrade Wizard Log Files

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

Windows: *SI EBEL_ROOT\log\PROCESS*

UNIX: *\$SI EBEL_ROOT/log/PROCESS*

In these paths, *PROCESS* specifies *UPGRADEOPTION_ENVIRONMENT_VERSION* by default.

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

Windows: *SI EBEL_ROOT\log\upgrep_dev_versionnumber*

UNIX: *\$SI EBEL_ROOT/log/upgrep_dev_versionnumber*

You can select a different directory for the log files, by specifying a value for the Log Output Directory option in the Database Configuration Wizard. This option is for specifying a different subdirectory, under the log directory, in which to create the log files.

Each subdirectory in the log directory contains the following files:

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

TIP: Each upgrade process, for example, a development upgprep, has a driver file located in *DBSRVR_ROOT\platform\upgrade\version*. An example of a driver file: *driver_upgprep_dev_81.ucf*. The Upgrade Wizard performs the steps in the driver file. If a step calls a utility, such as *ddlmp* or *dataimp*, then the */l* option in the command syntax specifies the name of the log file. Unless you modified the output location for the log file, it is located at *si_ebsrvr\log\process*.

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

The output and state log file directories are automatically archived on subsequent runs of a process that completes successfully. (The names of subsequent log directories are appended with *_1*, *_2*, and so on.) To preserve disk space, periodically delete or save log directories to another location.
- **summary.** This subdirectory is generated by the Logparse utility. The Logparse utility also generates a report that summarizes the upgrade, *summary.html*. The *summary.html* file contains links to the html pages in the summary subdirectory.
- **summary.html.** This HTML file is created by the Logparse utility. It summarizes the log files in the output directory. Use this file to verify that all upgrade steps completed successfully and to determine whether there were any errors.
- **summary.txt.** This text file is created by the Logparse utility. This file contains the same information as *summary.html*.

NOTE: If a browser is not installed on the computer you are using, then review *summary.txt* instead of *summary.html*.
- **summary.xml.** This file is produced by the Logparse utility during production upgrades and used by the Siebel Upgrade Tuner to parallelize table creation and index creation and to inactivate SQLs that affect no rows. (Development upgrades do not produce a *summary.xml* file.)
- **upgtuner_ftp_get.txt. (UNIX Only).** This file is produced by the Logparse utility during production upgrades. You can use this file to transfer the upgrade scripts from your UNIX computer to a Windows temporary directory to perform upgrade tuning. (Development upgrades do not produce *upgtuner_ftp_get.txt*.)
- **upgtuner_ftp_put.txt. (UNIX Only).** This file is produced by the Logparse utility during production upgrades. After you tune your production upgrade scripts, you can use this file to transfer the upgrade scripts from the temporary directory on your Windows computer back to your UNIX environment. (Development upgrades do not produce *upgtuner_ftp_put.txt*.)

The Logparse Utility

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

Parameters Report

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

Step/Error Summary

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

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

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

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

Performance Report

The performance report provides a list of the longest running SQL scripts and a list of SQL scripts returning no rows. The lists include a link to each script. Upgrade Tuner uses this information as input to help you tune upgrade performance.

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

About the Siebel Case Insensitivity Wizard

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Databases: All databases.

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

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

NOTE: Before performing an upgrade or incremental repository merge, you must drop any CIAI indexes and CIAI columns. Run the Case Insensitivity Wizard again after completing the upgrade or incremental repository merge.

Related Topics

[Appendix D, "Siebel Case Insensitivity Wizard"](#)

Related Books

Configuring Siebel Business Applications

About the Siebel Repository Merge

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Repositories are located in tables in the Siebel database. These tables store the definitions of the objects used to build Siebel Business 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 Business Applications, such as business components, applets, controls, and the relationships between them.
- Definitions of the tables in the Siebel database (metadata). These records define the logical schema of the Siebel database. Later in the upgrade process, you will synchronize the physical schema of the Siebel database with the logical schema as defined by these records.

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

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

The development upprep adds three additional repositories to the Siebel database, as shown in Table 10. These repositories are required for the repository merge.

Table 10. Development Database Repositories

Repository Name	Added During Upgrade?	Siebel Tools Name	Description
Prior Customer Repository	No	Prior Customized	This is your current Siebel Repository, before upgrading. It contains any changes you have made. You renamed it Prior Customer Repository before doing the initial database upgrade.
Prior Siebel Repository	Yes	Prior Standard	The standard Siebel Repository for your installed release (the release 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 in the Prior Siebel Repository, then 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, then 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, then 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, then 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, then 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, then 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, then 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 11 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, and 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 11. 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, or Inactive	Standard, or Inactive
Standard	Customized	Customized	Customized
Standard	Customized	Customized, or Inactive	Customized, or Inactive
Standard	Customized	Revised (conflict)	Revised (conflict)
Not applicable	New	New	New
Not applicable	Not applicable	New	New
Standard	Deleted	Standard	Standard
Standard	Deleted	Not applicable	Not applicable
Standard	Deleted	Not applicable	Not applicable

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, then 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 user interfaces after the merge.

Incorporate Custom Layout

You can choose Incorporate Custom Layout (ICL) when performing a repository merge. ICL is intended to reduce the postmerge user interface reconfiguration workload for customer-modified user interface objects.

An ICL merge is a regular repository 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

[Appendix C, "Siebel Incorporate Custom Layout Upgrade Option"](#)

["About Inheriting Upgrade Behavior in a Siebel Upgrade" on page 57](#)

["About the Siebel Postmerge Utilities" on page 59](#)

["Reviewing Siebel Repository Object Property Conflicts" on page 174](#)

About Single-Step Repository Upgrade

Prior to the release of Siebel CRM version 8.1.1, two Siebel repositories supported Siebel applications: Siebel Business Applications (SEA) repository and Siebel Industry Applications (SIA) repository. Deploying applications from outside either group was not possible through a direct, single-step upgrade process.

Going forward from the release of Siebel CRM version 8.1.1, a single repository is supported (SIA), and an upgrade process is available that enables a direct, single-step upgrade from SEA to SIA repositories. After running database configuration utilities, you can perform direct upgrades from an SEA deployment to a higher version SIA repository through a single-step process, for certain supported upgrade paths. For example, you can directly upgrade to the current release from Siebel CRM version 7.8 SEA or SIA, or from version 8.0 SEA or SIA, using a single-step upgrade method.

Table 12 describes examples of different methods for upgrades to the current release, including upgrade paths that use the single-step repository upgrade. The repository for the current release is SIA in all cases. For more information, see [“Supported Upgrade Paths for Siebel CRM Version 16.0” on page 17](#).

Table 12. Upgrade Methods for Different Upgrade Paths

Current Siebel Version	Target Upgrade Version	Upgrade Method
Siebel CRM version 7.5.3 or version 7.7.2 (SEA or SIA repository)	Siebel CRM version 16.0	Two-step repository upgrade. Upgrade to Siebel CRM version 8.1.1 (SEA or SIA repository), then to Siebel CRM version 16.0.
Siebel CRM version 7.8 or version 8.0 (SEA or SIA repository)	Siebel CRM version 16.0	Single-step repository upgrade.
Siebel CRM version 8.1.1 or later (SEA repository)	Siebel CRM version 16.0	Single-step repository upgrade.
Siebel CRM version 8.1.1 or later (SIA repository)	Siebel CRM version 16.0	Incremental repository merge.
Siebel CRM version 8.2.2 or later (SIA repository)	Siebel CRM version 16.0	Incremental repository merge.
Siebel CRM version 15.0 through version 15.4	Siebel CRM version 16.0	Incremental repository merge.
Siebel CRM version 15.5 or later	Siebel CRM version 16.0	Incremental repository merge.

Data Model Changes for Single-Step Repository Upgrade

Due to data model changes around Address objects between the SEA and SIA repositories, the migration from the SEA repository to the SIA repository will result in some SEA objects being made obsolete. You must reconfigure all Address-related business components that are in use. In general, the same development upgrade process needs to be applied for the SEA to SIA migration on the same version. Migrating from SEA to SIA on the same version requires less effort than upgrading from one version to a higher version. For more information, see [“Migrating Address Data After a Direct SEA to SIA Upgrade” on page 246.](#)

About Inheriting Upgrade Behavior in a Siebel Upgrade

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

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

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

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

For descendants that are for 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, then it is obsolete, and its upgrade behavior is not propagated to descendants.
If an ancestor object is obsolete in the New Siebel Repository, then its descendants are not also obsolete.
If an upgrade ancestor is not present in the New Siebel Repository, then 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, then 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, then 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 51](#)

[Appendix C, “Siebel Incorporate Custom Layout Upgrade Option”](#)

About the Siebel Postmerge Utilities

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

You run the postmerge utilities after the repository merge completes.

The postmerge utilities make revisions to objects in the New Customer Repository. The utilities are useful for identifying problems with user interface customizations.

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

Windows: `SIEBEL_TOOLS_ROOT\repatch\reputility.xml`

The postmerge utilities are as follows:

- **CSSGridRepPatch**. Converts customized applet controls to grid-based layout.

- **CSSUI NavUpgradeReposPrep**. Converts customized screens to the Siebel CRM version 7.7 user interface navigation scheme.
- **CSSMVGUpgradePatch77**. Enables the shuttle feature in customized MVGs.
- **CSSUI UpgradeReports**. Lists unresolved problems in converting customized screens and views to the Siebel CRM version 7.7 user interface 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

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, the item identifier syntax changes to *xyyy*, 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, then the utility restores the control to its original location. If you have modified other properties of the control, then 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.) The utility does not convert these applets to grid-based. New and modified controls display in the correct locations. For upgrades from Siebel CRM version 7.7 or later, the utility makes no changes.

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

CSSUI NavUpgradeReposPrep

At Siebel CRM version 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 Siebel CRM version 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, then the utility creates one. This causes the orphaned views to display.

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

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

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

CSSMVGUpgradePatch77

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

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

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

CSSUI UpgradeReports

This utility makes no changes to the New Customer Repository. Instead, it scans user interface 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 must 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 instance that you used to perform the merge:

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

These Web template files support the user interface 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 7.8.2 and the Label on Top ICL options, then the utility copies Web template files from the `782` and `TopLabel` subdirectories of `reppatch`.

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

CSSWFRepPatch

This utility changes workflow step references from row-id references to name references for child objects of steps. This completes the process of migrating workflows to the repository at Siebel CRM version 7.7. This utility primarily affects upgrades from releases prior to Siebel CRM version 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 user interface objects based on the object's characteristics rather than Upgrade Behavior setting.

Related Topics

[“About the Siebel Repository Merge” on page 51](#)

[“About Inheriting Upgrade Behavior in a Siebel Upgrade” on page 57](#)

[Appendix C, “Siebel Incorporate Custom Layout Upgrade Option”](#)

About Tuning Siebel Production Upgrade Files

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

Platforms: Windows and UNIX only.

The Upgrade Tuner allows you to tune the upgrade files generated by the Siebel Database Configuration Utilities. Tuning the production upprep 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 Business 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 13](#).

Table 13. Upgrade Tuner Support for Parallel Table and Index Threads

Operating System	Oracle Database	IBM DB2	Microsoft SQL Server
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 14](#).

Table 14. Upgrade Tuner Support for Zero-Row SQL Deactivation

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

For information on which versions and releases of the operating systems and RDBMS products that Oracle supports for Siebel products, see the Certifications tab on My Oracle Support.

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

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

When to Use Upgrade Tuner

Use Upgrade Tuner in the production test environment to tune the upgrade files that perform the production upgprep. 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 upgprep 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, the master .ucf file, and the 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. Tables or indexes not assigned to a parallel thread remain in the serial thread.

When you run the Upgrade Tuner, the Number of Threads parameter is automatically set, based on a log analysis, to the value effectively required to run the upgrade. In most cases this value is set at 4. This value, which can be changed in the master upgrep file, drives the maximum number of threads for each single Siebel upgrep task, such as ddlimp executions concurrently running for this process. When setting this value you must consider the system resources of the computer from which the upgrep phase was initiated. These changes must be tested on a copy of the Siebel production database environment, prior to rolling these out for the actual Siebel production upgrade downtime window.

Deactivate 0-Row SQLs

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

Files Required to Run Upgrade Tuner

Upgrade Tuner requires the following files. The location of the files is the same on both Windows and UNIX hosts. [Table 15](#) displays these files.

Table 15. Files Required to Run Upgrade Tuner

File	Location on a Windows Host
summary.xml	<i>SI EBEL_ROOT\log\upgrep_prod_VERSION</i> For example, <i>SI EBEL_ROOT\log\upgrep_prod_version\summary.xml</i>
master_upgrep_prod_VERSION.ucf	<i>SI EBEL_ROOT\bin</i> For example, <i>SI EBEL_ROOT\bin\master_upgrep_prod_version.ucf</i>
schema*.ddl	<i>DBSRVR_ROOT\DBPLATFORM</i> For example, <i>DBSRVR_ROOT\oracl e\schema.ddl , schema_i 1.ddl , schema_t1.ddl</i>

Table 15. Files Required to Run Upgrade Tuner

File	Location on a Windows Host
dri ver_upgprep_prod_ <i>VERSION</i> . ucf	<i>DBSRVR_ROOT\DBPLATFORM\upgrade\VERSION</i> For example, <i>DBSRVR_ROOT\oracl e\upgrade\versi on\dri ver_upgprep_prod_versi on. ucf</i>
*. sql	<i>DBSRVR_ROOT\DBPLATFORM\upgrade\VERSION</i> For example, <i>DBSRVR_ROOT\oracl e\upgrade\versi on\pret. sql</i> , <i>preschm. sql</i>

How Upgrade Tuner Modifies Files

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

- “Driver Configuration File” on page 66
- “Schema.ddl File” on page 67
- “SQL Files” on page 67

Driver Configuration File

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

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

- Making schema changes using the ddlimp 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 *dri ver_upgprep_versi on. ucf* to *dri ver_upgprep_versi on. ucf. ori g*. It then makes changes to *dri ver_upgprep_versi on. 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 copi es *dri ver_upgprep_versi on. ucf* to *dri ver_upgprep_versi on. ucf. ol d*. It then makes changes to *dri ver_upgprep_versi on. ucf*.
- When you save your changes after the third session and all following sessions, Upgrade Tuner saves the driver file to .old again and then updates the driver file.

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

Schema.ddl File

When you run the Upgrade Wizard after the production upgrep it reads the driver file. The driver file contains steps that call the 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 and Exit, Upgrade Tuner creates a schema.ddl thread-file for the new thread.

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

The t1 thread-file contains the table creation information for the tables in Parallel Thread 1. Parallel thread 2 information is contained in the t2 thread-file, and so on.

- When you create new threads in the Parallelize Index Creation page, the thread-files are named i1, i2, and so on. For example, the information for Parallel Thread 1 for index creation is contained in schema_i 1. ddl .
- If you run Upgrade Tuner and change the tables or indexes assigned to a thread, then 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 (ddlimp in table creation mode)
- schema.ddl (ddlimp in table creation mode)
- Index thread files beginning with i1 (ddlimp in index creation mode)
- schema.ddl in index creation mode (ddlimp 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 file `dri ver_upgrep_prod_versi on. ucf`, 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.

4

Application Planning for a Siebel Upgrade

This chapter provides information on Siebel Business Applications and issues to consider when upgrading your Siebel database. This chapter includes the following topics:

- [Upgrade Planning for Multilingual Siebel Deployments on page 69](#)
- [Upgrade Planning for Language String Migration on page 71](#)
- [Upgrade Planning for Siebel Unicode Support on page 71](#)
- [Upgrade Planning for Siebel RC4 or AES Encryption on page 73](#)
- [Upgrade Planning for Siebel Web Template Files and Style Sheets on page 73](#)
- [Upgrade Planning for Siebel Access Control on page 75](#)
- [Upgrade Planning for Migrating Siebel Address Data on page 75](#)
- [Upgrade Planning for Siebel Workflow Designer on page 76](#)
- [Upgrade Planning for Mobile Devices in the Siebel Environment on page 79](#)
- [Upgrade Planning for Resonate Central Dispatch in the Siebel Environment on page 80](#)
- [Upgrade Planning for Siebel String Translation on page 80](#)
- [Upgrade Planning for Siebel Personalization on page 81](#)
- [Upgrade Planning for Siebel Pricer and Order Management on page 81](#)

Upgrade Planning for Multilingual Siebel Deployments

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

As of Siebel CRM 8.0, the upgrade process upgrades both the primary (base) language and all deployed languages.

CAUTION: Each multilingual deployment must contain a primary language as well as those that are designated as secondary, or non-primary. For instance, a deployment might have ENU as the primary language and FRA, and JPN as its non-primary languages. Replacing the primary language, ENU in this example, with one of the non-primary languages during upgrade is not supported, as it will result in severe data corruption, making recovery extremely difficult.

The development and production upprep process does the following for the primary language and all deployed languages:

- (Development upprep). Imports user interface strings into the Prior Standard Repository, New Customer Repository, and New Siebel Repository.

- Imports seed data into the database.
- In the S_LST_OF_VAL table, an SQL script activates all newly imported records and configures them to support multilingual lists of values (MLOVs). LOVs remain enabled for the primary language.

No language-related manual tasks are required to upgrade the deployed languages, except where noted.

Validation

Prior to starting a development or production upgrade, you must have installed the new release's language packs for all deployed languages.

The Database Configuration Utilities validate deployed languages by comparing the status of language packs installed on the Siebel Server with the language IDs of records in the S_LST_OF_VAL table. If there are language IDs in S_LST_OF_VAL but no corresponding deployed language on the Siebel Server, then the validation fails and you cannot continue the upgrade.

Prior to starting a development or production upgrade, you must review the records in S_LST_OF_VAL, using Siebel Tools. Remove or deactivate records for languages that you do not have deployed.

For more information, see ["Running the Siebel Database Configuration Wizard on Windows" on page 136](#) or ["Running the Siebel Database Configuration Wizard on UNIX" on page 141](#), depending on your operating system.

CAUTION: Regardless of whether you have American-English deployed, there will be records in S_LST_OF_VAL where the Language Name value equals English-American. Do not deactivate or delete these records. The validation process ignores them.

Restrictions

The following restrictions apply to multilingual upgrades:

- Installed languages must be deployed. If you have installed a language but not deployed it, then the language will not be upgraded.

A deployed language is one where you have installed the language pack, imported the user interface strings into the Siebel Repository, imported the seed data into the database, activated the seed data records added to S_LST_OF_VAL, and configured the records for MLOVs.
- You cannot select which languages to upgrade. The Upgrade Wizard upgrades all deployed languages.
- You cannot add or remove a language during an upgrade. If you want to add a language that is not currently deployed, then do so after the upgrade. For more information on adding new languages, see *Siebel Global Deployment Guide* and *Siebel Installation Guide* for the operating system you are using.

Upgrade Planning for Language String Migration

Upgrades from: All Supported Siebel releases.

Environments: Development environment only.

Databases: All databases.

Locale-specific language values can change from one release of Siebel Business Applications to another. Some ENU values shipped with your upgrade might no longer map to language strings from your previous release. For instance, in a German environment, the English word *Individual* mapped in previous releases to the German word *Einzelperson*, but in the subsequent release now maps to the German word *Individuell*. To ensure correct data migration, it is advised that you review your previous `upglocale.lang_code` file against the current file of the same name to determine whether any language strings require changes to ensure that they properly map when the upgrade utilities are run.

Upgrade Planning for Siebel Unicode Support

Environments: Development, production test, production.

For Western European languages and Japanese, Siebel Business Applications support both non-Unicode and Unicode code pages. For all other supported languages, Siebel Business 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 Business Applications use Unicode internally. If the RDBMS is not using Unicode, then Siebel Business 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 RC4 or AES.

CAUTION: Migrating to Unicode is more complex than simply importing your existing data into a Unicode database. Failure to execute the migration correctly can result in serious data corruption or unrecoverable data loss. For this reason, Oracle's Advanced Customer Services participation is mandatory. To perform a Unicode migration, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

For Siebel language support, Unicode support, and legacy code page support, see 1513102.1 (Article ID) on My Oracle Support.

For information about Unicode and global deployment for Siebel Business Applications, see *Siebel Global Deployment Guide*.

For information on upgrading to RC4 or AES encryption, see *Siebel Security Guide*.

Planning Considerations for the Unicode Migration

Migrations to Unicode require the assistance of Oracle's Advanced Customer Services.

Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services to migrate your upgraded database from a non-Unicode code page to Unicode. To perform the migration, you can use either database vendor native utilities or Siebel utilities.

If you are planning to migrate your upgraded application to Unicode, then consider the following points:

- **Database size increase. Migration to Unicode increases the size of your database.** For this reason, you must allocate additional space for your database before migrating to Unicode. For more information, create a service request (SR) on My Oracle Support. Also, you can contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- **IBM DB2 data truncation.** Migration to Unicode might cause truncation of certain data in IBM 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 might be truncated and take appropriate measures before migration. For more information, create a service request (SR) on My Oracle Support. Also, you can contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services. See also "[Identifying IBM DB2 Long Columns for Truncation in a Siebel Upgrade](#)" on page 96.
- **IBM DB2 custom tablespace information.** The upgrade does not preserve custom tablespace information for IBM DB2 databases. This presents a problem during your migration to a Unicode code page, because you must know which tables need to be re-created. You must modify the upgrade scripts to handle custom tablespaces.

For instructions about how to modify upgrade scripts to handle custom tablespaces, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- **Third-party product integration.** Migration to Unicode might affect integration with third-party systems. For more information, create a service request (SR) on My Oracle Support. Also, you can contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- **Environment code pages.** You cannot use a non-Unicode code page for your development environment, and then later migrate to Unicode for your production environment.

Supported Types of Unicode

The current release supports two types of Unicode:

- **UTF-8.** UTF-8 uses the same encoding for Western European languages. It occupies one byte for Western European languages and up to three bytes for some Asian languages, such as Japanese.

- **UCS-2.** UCS-2 is supported for IBM DB2 and Microsoft SQL Server databases. UCS-2 does not map one-to-one with Western European languages. It occupies two bytes for all languages.

Upgrade Planning for Siebel RC4 or AES Encryption

Upgrades from: All Supported Siebel releases.

Environments: Development, production test, production.

Databases: All databases.

Platforms: Windows and UNIX only.

As of Siebel Innovation Pack 2014, Siebel CRM no longer supports RC2 encryption. Siebel 7.7 introduced support for the AES encryption method, the government standard for secure applications. Siebel Business Applications continue to support RC4 and AES data encryption.

Data that used the standard encryptor (supported in some prior releases) or RC2 cannot be read by applications in the current release. You must upgrade your encryption method to RC4 or AES.

Use the Encryption Upgrade Utility to convert unencrypted data and data that was encrypted using the standard encryptor or RC2 to the RC4 or AES encryption method. Also run the Encryption Upgrade Utility to upgrade to stronger encryption levels. To upgrade your encryption method, see *Siebel Security Guide*.

Upgrade Planning for Siebel Web Template Files and Style Sheets

Environments: Development, production test, production.

Siebel Web templates and Siebel Web template files (SWT 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 Siebel Web templates to build Web pages, which it then forwards to the Web server.

A Siebel Web template file contains regular HTML or XML tags interspersed with Siebel tags. Siebel tags are prefixed by swe and contain placeholders for user interface objects such as controls and data. HTML formatting tags are defined in cascading style sheets (such as main.css).

When you install Siebel Server or Siebel Tools for the current release, you receive a new set of Siebel Web template files. When you install Siebel Web Server Extension or Siebel Tools, you receive new style sheet files. The upgrade process does not use your existing files. If you have manually customized your Siebel Web template files or style sheet files, then you must evaluate whether to reimplement these customizations in the new Siebel Web template files and style sheet file.

Observe the following planning guidelines for reimplementing customizations:

- Resolve any user interface problems related to object definitions in the Siebel Tools repository first. While doing so, review the areas of the new user interface where you plan to implement Siebel Web template file customizations.

- Evaluate existing customizations to Siebel Web template files, and decide which ones to reimplement. Changes to the user interface in the new release might 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 Siebel Web template files and style sheet files. This allows you to maintain orderly distribution of the files to developers.
- Applets typically have a separate Siebel Web template file for each applet mode. Customize all the mode Siebel Web template files for an applet at the same time. This allows you to verify applet functionality in a single test pass in the user interface.
- Individual Siebel Web template files are typically used by multiple screens, views, or applets. Set up formal test plans that verify customizations are correct across all user interface objects that use each Siebel Web template file. This reduces the amount of time required to verify customizations and prevents unintended changes to the user interface.
- Reimplement style sheet customizations in two passes. On the first pass, implement only those changes required to address user interface usability issues. On the second pass, implement the remaining style sheet customizations after Siebel Web template customizations are complete. This shortens the time required to resolve functional problems in the user interface.

After customizations have been completed, the Siebel Web template files must be copied to the Siebel Server installations in the environment, and the style sheet files must be copied to the Siebel Web Server Extension installations in the environment. The files must also be included in upgrade kits sent to remote sites.

Siebel Web template files have an .swt file extension and are located in the `webtempl` directory in both the Siebel Server and Siebel Tools installations.

The style sheet files are located in the installation directory of Siebel Web Server Extension (SWSE) and Siebel Tools (Windows path syntax). For example:

SWSE: `SWSE_ROOT\publi c\files\main.css`

Siebel Tools: `SIEBEL_TOOLS_ROOT\publi c\files\main.css`

The steps for reimplementing customizations to these files and copying them to new environments are included in the upgrade process topics in [Appendix B, "Overview of Performing a Siebel Database Upgrade."](#)

For information on how Siebel Web templates and style sheet files work, see *Configuring Siebel Business Applications* and *Configuring Siebel Open UI*.

In Siebel Innovation Pack 2015 and later, a script is executed that migrates data from Siebel Web template files into table-based content in the Siebel database. For Incremental Repository Merge, Siebel Web template migration occurs during the upgrep operation. For full upgrades, this migration occurs during the upgphys operation. For more information, see ["About Migrating Siebel Web Template Data" on page 236](#).

Upgrade Planning for Siebel Access Control

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

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 *Siebel Security Guide*.

Upgrade Planning for Migrating Siebel Address Data

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.

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, then 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 whether your custom columns can be mapped to standard columns. If no matching standard columns exist, then create new columns on the target tables. During the database upgrade, you do this after running the Database 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 315](#). 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 Workflow Designer

Platforms: Windows, UNIX, IBM z/OS.

As of Siebel CRM 8.x, the repository merge process preserves customizations you have made to seeded workflows. A seeded workflow is a Workflow Process object and all child objects shipped as part of a Siebel release.

Special premerge and postmerge steps have been added to the regular repository merge to allow merging customer-modified seeded workflows:

- 1 Workflow premerge.** When you start a repository merge, this step runs and prepares customer-modified seeded workflows in the Prior Customer Repository for the repository merge.
- 2 Repository merge.** After the workflow premerge completes, the main repository merge runs. The repository merge identifies modifications you have made to seeded workflows in the Prior Customer Repository. It then copies the modifications to the corresponding seeded workflows in the New Customer Repository.
- 3 Workflow postmerge.** After the main repository merge completes, the workflow postmerge runs and sets workflow versions in the New Customer Repository.

All three parts of the repository merge display in the status bar of the Merge Repositories dialog box when you perform a repository merge.

The workflow merge steps fit into the overall flow of a repository merge as follows:

- Repository merge starts
- Workflow premerge
- Main repository merge
- Workflow post-merge
- ICL merge steps (If this is an ICL merge)
- Merge ends
- Run the postmerge utilities

Workflow Premerge

When you start the repository merge, the workflow premerge step runs and prepares customer-modified seeded workflows for the repository merge.

If the workflow exists in both the Prior Customer Repository and New Siebel Repository, and the version number in the Prior Customer Repository is greater than 0, then the workflow is customer-modified.

For these workflows, the premerge process makes the following changes:

- Deletes version 0 of the customer-modified seeded workflow in the Prior Customer Repository.
- In the Prior Customer Repository, copies the most recent version of the customer-modified seeded workflow and sets the copy's version to 0.

The most recent version of the workflow is the one with the highest version number and Status indicates Completed. This is called the version n workflow.

- Sets the copy's status indicates Completed.

The workflow premerge creates a version 0 copy of version n in the Prior Customer Repository because the merge process requires that workflows it compares between repositories must have the same name, must be version 0, and must have Status indicates Completed.

Repository Merge

The main repository merge compares workflows in the Prior Customer Repository, Prior Siebel Repository, and the New Siebel Repository:

- **Customer-Modified seeded workflows.** The repository merge copies the modifications to version 0 of the workflow to the Prior Customer Repository to the same workflow in the New Customer Repository. The merge also copies versions 1 through n of the workflow to the New Customer Repository.

An exception is when a workflow's object attributes are different in all three compared repositories. This means the object is both customer-modified and has also changed in the new release. This causes a merge conflict.

The repository merge handles workflow attribute conflicts in the same manner as other objects. The merge process typically resolves merge conflicts in favor of the New Siebel Repository. You can review workflow-related attribute conflicts in the Application Upgrade Attribute List screen in Siebel Tools.

If you modify a seeded workflow by deleting any of its child objects, then the merge process does not delete the child objects in the New Customer Repository. After the merge, you must review child objects and delete them as desired.

- **Unmodified seeded workflows.** If the workflow exists in both the Prior Customer Repository and New Siebel Repository, and the highest version level is 0 in the Prior Customer Repository, then the workflow is unmodified.

After the merge, the seeded workflow in the New Customer Repository will be the one that shipped with the new release.

- **Customer-Created workflows.** If the workflow exists in the Prior Customer Repository but not the Prior Siebel Repository, then the workflow is customer-created. The repository merge copies all versions of customer-created workflows to the New Customer Repository.
- **Customer-Deleted Workflows.** If the workflow exists in the Prior Siebel Repository but not the Prior Customer Repository, then the workflow is customer-deleted. The repository merge does not delete the workflow from the New Customer Repository. After the merge, you must review these workflows and delete them as desired.
- **Obsolete seeded workflows.** If the workflow exists in the Prior Siebel Repository but not the New Siebel Repository, then the workflow is obsolete. The repository merge does not copy obsolete seeded workflows from the Prior Customer Repository to the New Customer Repository unless they are customer-modified. After the merge, you must review customer-modified, obsolete workflows and delete them as desired.

Workflow Postmerge

The workflow postmerge step runs after the main repository merge completes and does the following for seeded workflows that are customer-modified:

- Changes the version of the workflow from 0 to $n + 1$ and sets the $n+1$ version's workflow status to In Progress.

For example, if the most recent version of customer-modified Workflow A in the Prior Customer Repository is version 3, then the merged version in the New Customer Repository will be version 4 and will have Status indicates In Progress.

- Copies version 0 of the workflow from the New Siebel Repository to the New Customer Repository. This reinstates the version 0 seeded workflow that shipped with the new release.

Status of Customer-Modified Workflows After the Merge

After the whole repository merge is complete, customer-modified seeded workflows appear as follows in the New Customer Repository:

- **Version 0.** This is the seeded workflow that shipped with the new release.
- **Versions 1 through n .** These versions are copied intact from the Prior Customer Repository.

- **Version $n+1$.** This is the new merged workflow version. It is a combination of the seeded workflow that shipped with the new release and the modifications contained in version n in the Prior Customer Repository.

Logging

Workflow premerge and postmerge. The workflow premerge and postmerge steps write to the same log file:

```
SIEBEL_TOOLS_ROOT\bin\merge0_ver.txt
```

Each time you run the merge process, the name of the merge0_ver.txt file is incremented, for example to merge1_ver.txt.

If the log file contains the error `IDS_ERR_DEV_MRG_PREMERGE_FAILED`, then this means that the premerge step could not delete one or more version 0 seeded workflows. This causes the main merge process to be unable to merge the customer-modified seeded workflow into the New Customer Repository. You must manually reimplement customizations to these workflows. You need not rerun the repository merge.

If the log file contains postmerge errors, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Main repository merge. The main repository merge logs workflow-related messages to the standard merge log file, merge0.txt. If you find workflow-related errors in this log file, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

The main repository merge log file is located in the same directory as the log file for the workflow premerge and postmerge steps.

Upgrade Planning for Mobile Devices in the Siebel Environment

Platforms: Windows, UNIX, IBM z/OS.

Before the upgrade, verify that mobile devices are running an operating system supported by your version of Siebel CRM. Also, verify that third-party software is the correct version. For more information, see the Certifications tab on My Oracle Support.

After the upgrade, you must enter any mobile device-related application configuration changes into the mobile device administration screen.

Mobile Application Upgrade

Mobile applications do not upgrade automatically. Users might need to download the latest version of the application after an upgrade to the current release. It is not necessary to uninstall the application first.

Upgrade Planning for Resonate Central Dispatch in the Siebel Environment

Platforms: Windows, UNIX, IBM z/OS.

Support for Resonate Central Dispatch is discontinued as of Siebel CRM version 7.7. It has been replaced by a load balancing module that is included in the Siebel Web Server Extension. For a description of the load balancing module and information about requirements for third-party HTTP load balancers, see *Siebel Deployment Planning Guide*. For information on configuring load balancing, see *Siebel Installation Guide* for the operating system you are using.

Upgrade Planning for Siebel String Translation

Platforms: Windows, UNIX, IBM z/OS.

Based on the language for the Siebel upgrade process, the Siebel upgrade scripts are created accordingly.

NOTE: To avoid unintended results, you must perform your upgrade in the same language as that of the base language in the prior release.

When the Database Configuration Wizard is started, the sqlgen utility generates the Siebel upgrade scripts based on the information located in the upgfile.xml file. When the Database Configuration Wizard completes, SQL scripts are generated, and replace the placeholders in the upgfile.xml with the language-specific values located in the *DBSRVR_ROOT/lan_code/upglocale/lan_code* file.

In the following example, the locale file is upglocale.fra, for a FRA based installation. The upgfile.xml contains the following statement:

```
update S_DOC_QUOTE
```

set the QUOTE_SUB_TYPE_CD value to `'<XTL_STRING>Private</XTL_STRING>'`;

where the QUOTE_TYPE equals `'<XTL_STRING>Template</XTL_STRING>'`;

The placeholders `<XTL_STRING>Private</XTL_STRING>` and `<XTL_STRING>Template</XTL_STRING>` are values that the sqlgen utility changes during the upgrade. The XTL_STRING values are replaced based on what the sqlgen utility locates as a match for the *Private* and *Template* strings in the upglocale.fra file, located under *dbsrvr/FRA*. For this example, the FRA base language dictates *Template = Modèle* and *Private = Privé*.

CAUTION: If you modified string mappings in your previous Siebel release, then you must contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services in order to modify standard upgrade scripts accordingly. If you do not, then your Siebel application data from the prior release might not be migrated to the new Siebel release's data model.

Upgrade Planning for Siebel Personalization

Access control is based on primary responsibility name. If you have defined conditional expressions for applets in the Administration - Personalization screen, then plan to review these after the upgrade and verify that they use Primary Responsibility name. For more information on access control, see *Siebel Security Guide*. For more information on personalization management, see *Siebel Personalization Administration Guide*.

Upgrade Planning for Siebel Pricer and Order Management

Data associated with Siebel Pricer features are upgraded as shown in [Table 16](#). These upgrade changes apply to upgrades from any Siebel CRM version 7.x release.

Table 16. How Siebel Pricer Features Are Upgraded

Feature	How This Feature Upgraded
Price lists and cost lists	Automatically upgraded as part of price list
Customizable product pricing	Automatically upgraded as part of price list
Service pricing (based on covered product)	Automatically upgraded as part of price list
Volume discount	Automatically upgraded
Attribute pricing	Automatically upgraded as attribute adjustment
Pricing model	Not automatically upgraded. Must be redesigned and reimplemented as pricing procedures.
<ul style="list-style-type: none"> ■ Aggregate factor ■ Bundle factor ■ Single factor ■ Matrix factor ■ Script based 	Bundle factor definitions are upgraded to aggregate discounts and aggregate discount sequences.

Siebel 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 *Siebel Pricing Administration Guide* and *Siebel Order Management Infrastructure Guide*.

For a scenario that describes how to reimplement Siebel CRM version 7.x pricing models, see 473908.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 639.

5

Preparing for Siebel Database Upgrade

This chapter provides initial preparatory information for the Siebel database upgrade. This chapter includes the following topics:

- [Verifying Siebel Database Connectivity on page 83](#)
- [Preparing Siebel Tables and Views for Upgrade on page 83](#)
- [Preparing Siebel Custom Indexes for Upgrade on page 84](#)
- [Exporting Siebel Interface Table Data on page 85](#)
- [Archiving Unneeded Siebel Repositories on page 85](#)
- [Preserving Siebel Dock Objects and Visibility Rules on page 85](#)
- [Securing AIX Memory Allocation Segment Space for the Siebel Database on page 86](#)
- [Preparing for a Multilingual Upgrade on page 86](#)

Verifying Siebel Database Connectivity

Environments: Development, production test, production.

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, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

In the production environment, you do not have to define an ODBC connection to the development environment Siebel database or the production test environment Siebel database.

Preparing Siebel Tables and Views for Upgrade

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

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 for Siebel tables, then 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

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

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, then you must migrate the data to new extension columns before upgrading the Siebel database. For assistance, create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- **Custom indexes that were not defined through Siebel Tools.** Custom indexes created without using Siebel Tools are not included in the schema definition in the Siebel Repository. These indexes are dropped during the database upgrade. To preserve these indexes, add them to the Siebel Repository using Siebel Tools.
- **Custom indexes on interface tables.** Custom indexes on interface tables are not re-created during the upgrade. You must re-create them after the upgrade is complete.
- **Custom indexes on base tables.** The upgrade automatically removes and re-creates custom indexes on base tables.
- **Custom indexes might need to be changed to reflect schema changes.** Reevaluate custom indexes for applicability in the new release. They might 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* on My Oracle Support.

Exporting Siebel Interface Table Data

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

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

Environments: Development, production test, production.

Perform this task before running the Database 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, then 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, then 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 *Using Siebel Tools*.

Preserving Siebel Dock Objects and Visibility Rules

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

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 must reapply the changes after the upgrade is complete.

Changing the definition of dock objects requires the assistance of Oracle Advanced Customer Services. Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Securing AIX Memory Allocation Segment Space for the Siebel Database

Environments: Development, production test, production.

Databases: All databases. (Exception: this topic does not apply to Microsoft SQL Server.)

Platforms: UNIX only.

Before you run an upgrade on AIX, set the following environment variable on the AIX computer that you are using for the upgrade:

```
setenv LDR_CNTRL LOADPUBLI C@MAXDATA=0x60000000
```

This will prevent a shortage of memory allocation segment space that might occur on the computer where both the Siebel Database Server and Siebel Server are installed. After a successful upgrade, reset this parameter to the original value.

Preparing for a Multilingual Upgrade

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

As of Siebel CRM 8.0, the upgrade process upgrades both the primary (base) language and all deployed languages.

The Database Configuration Utilities validate deployed languages by comparing the status of the language packs installed on the Siebel File System with the language IDs of records in the S_LST_OF_VAL table. If there are language IDs in S_LST_OF_VAL but no corresponding deployed language on the Siebel File System, then the validation fails and you cannot continue the upgrade.

Before doing an upgrep, use Siebel Tools to remove or deactivate any records in S_LST_OF_VAL for undeployed languages.

To prepare for a multilingual upgrade

- 1 Make a list of your deployed languages.

A *deployed language* is one where you have installed the language pack, imported the user interface strings into the Siebel Repository, imported the seed data into the database, activated the seed data records added to S_LST_OF_VAL, and configured the records for MLOVs.

- 2 In Siebel Tools, navigate to Screens, System Administration, and then List of Values.
- 3 Locate the Language Name field, and verify that no languages appear that are not currently deployed.

Even if you do not have English installed, then some records will have the Language Name value equal English-American. You can ignore these system records.

- 4 For records where you do not have the corresponding language pack deployed, investigate why these records are present and take one of the following actions:
 - If the record is not needed, delete it.
 - If you are not sure whether the record is needed, remove the check-mark from the Active field. This deactivates the record and prevents it from being included in the validation check performed by the Database Configuration Utilities.

CAUTION: Do not deactivate or delete records where the Language Name equals English-American, even if you do not have English-American deployed. These records are needed by the system. The validation process ignores these records.

6

Preparing an IBM DB2 Database for a Siebel Upgrade

This chapter provides initial preparatory information for a Siebel database upgrade on IBM DB2. This chapter includes the following topics:

- [Verifying the IBM DB2 Client for a Siebel Upgrade on page 89](#)
- [Verifying IBM DB2 Sort Order for a Siebel Upgrade on page 89](#)
- [Setting IBM DB2 Parameters for a Siebel Upgrade on page 90](#)
- [Verifying IBM DB2 Permissions for a Siebel Upgrade on page 92](#)
- [Verifying IBM DB2 Instance Owner Permissions for a Siebel Upgrade on page 92](#)
- [Creating IBM DB2 Temporary Tablespaces and Bufferpools for a Siebel Upgrade on page 92](#)
- [Analyzing IBM DB2 Custom Tablespace Requirements for a Siebel Upgrade on page 93](#)
- [Verifying the IBM DB2 Application Development Client for a Siebel Upgrade on page 95](#)
- [Identifying IBM DB2 Long Columns for Truncation in a Siebel Upgrade on page 96](#)

Verifying the IBM DB2 Client for a Siebel Upgrade

Environments: Development, production test, production.

Databases: IBM DB2 UDB only. This topic applies only to the 64-bit IBM DB2 RDBMS.

The Siebel Server supports only the 32-bit IBM DB2 client. Verify that you have not installed the 64-bit IBM DB2 client on the Siebel Servers. If you have installed the 64-bit IBM DB2 client, then replace it with the 32-bit client.

IBM supports the 32-bit IBM DB2 client working with 64-bit IBM DB2 databases.

Verifying IBM DB2 Sort Order for a Siebel Upgrade

Environments: Development, production test, production.

Databases: IBM DB2 only.

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, then you must re-create your database using the option `COLLATE USING IDENTITY`.

If sort order is correct, but you are still encountering errors, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services to help in further analysis.

To verify that your database was created using Identity sort order

1 Run the following query in the Siebel database:

```
select count (*) from SIEBEL.S_APP_VER where '$' > '/'
```

2 Review the result.

- If sort order is correct, then the result is as follows:

```
1
-----
0
(1) record selected.
```

- If sort order is incorrect, then you must re-create the database, using this option:

```
COLLATE USING IDENTITY
```

Setting IBM DB2 Parameters for a Siebel Upgrade

Environments: Development, production test, production.

Databases: IBM DB2 only.

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 IBM DB2 must be database-managed tablespaces (DMS) rather than system-managed tablespaces (SMS).
- Verify that the tablespaces are not near their capacity. This can be done by connecting to the database and issuing the following command:

```
DB2 list tablespaces show detail
```

The tables which follow provide upgrade-specific settings for the Database Manager and database. Use the following strategy to set parameters:

- Set parameters using the recommendations in *Siebel Installation Guide* for the operating system you are using. Recommendations are located in the chapter on configuring the RDBMS.
- For the upgrade, revise the configuration parameters listed in the tables which follow.
- After the upgrade, reset the configuration parameters to the values listed in *Siebel Installation Guide* for the operating system you are using.

IBM DB2 Database Manager Settings

The Upgrade Setting column in [Table 17](#) provides guidelines for setting configuration parameters specifically to optimize upgrade performance. Set these parameters for each IBM DB2 instance.

Table 17. IBM DB2 Database Manager Configuration Parameters

Parameter	Explanation	Upgrade Setting
SHEAPTHRES	Sort heap threshold (4 KB) If you reset SHEAPTHRES or SORTHEAP, then rebinding the instance is recommended.	Double the value allocated for SORTHEAP. See Table 18 on page 91 .

IBM DB2 Database Configuration Parameters

The Upgrade Setting column in [Table 18](#) provides guidelines for setting configuration parameters specifically to optimize upgrade performance. Set these parameters for each IBM DB2 instance.

Table 18. IBM DB2 Database Configuration Parameters

Parameter	Explanation	Upgrade Setting
SORTHEAP	Sort list heap (4 KB)	20000–40000 Recommended size; this might increase or decrease depending on the amount of memory in the database server computer and the size of the data. A 20000 setting allows SORTHEAP to increase up to 80 MB.
MAXLOCKS	Percentage of lock lists for each application	5
CHNGPGS_THRESH	Changed pages threshold	5
LOGRETAI N	Sequential or circular log files	NO Setting this parameter to YES means that log files are archived. You must periodically move or archive the log files to prevent the file system containing the log files from filling up.
LOGFI LSI Z	Log file size (4 KB)	Development environments: 8000-16000
SOFTMAX	Triggers bufferpool flushing	50

Verifying IBM DB2 Permissions for a Siebel Upgrade

Environments: Development, production test, production.

Databases: IBM DB2 only.

Platforms: UNIX only.

If you are running IBM DB2 on IBM AIX or Oracle Solaris, then perform the following steps before executing the Siebel Database upgrade.

To verify IBM 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

Environments: Development, production test, production.

Databases: IBM DB2 only.

Platforms: UNIX only.

If you are running IBM DB2 on AIX or Oracle Solaris, then 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, then errors will occur during the Siebel Database upgrade.

Creating IBM DB2 Temporary Tablespaces and Bufferpools for a Siebel Upgrade

Environments: Development, production test, production.

Databases: IBM DB2 only.

If your RDBMS is IBM DB2, then 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.

Creating a 16-KB Temporary Tablespace

Use the following procedure to create a 16-KB temporary tablespace.

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.

Creating a 32-KB Temporary Tablespace

Use the following procedure to create a 32-KB temporary tablespace.

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

Environments: Development, production test, production.

Databases: IBM DB2 only.

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, then 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, then 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, then you must resolve them before you proceed with the upgrade.

To analyze tablespace requirements for IBM DB2

- 1 Navigate to the following directory:

Windows: `\\$IEBEL_ROOT\bin`

UNIX: `/$IEBEL_ROOT/bin`

2 Type the following command line:

```
tblsize /U TABLEOWNER /P PASSWORD /C ODBC_DATASOURCE /F DDL_FILE  
/B DEFAULT_TABLESPACE /X DEFAULT_INDEXSPACE /K 16K_TABLESPACE /N 32K_TABLESPACE  
/Q REPORT_FILENAME /L LOG_FILENAME
```

where:

- *TABLEOWNER* is the tableowner
- *PASSWORD* is the tableowner password
- *ODBC_DATASOURCE* is the data source of the database
- *DDL_FILE* is the absolute path to the DDL file (this file is called `ddl.ctl`, and it is located in the `dbsrvr/DB2` directory)
- *DEFAULT_TABLESPACE* is the name of the 4-KB page standard Siebel tablespace
- *DEFAULT_INDEXSPACE* is the name of the standard Siebel index space
- *16K_TABLESPACE* is the name of the 16-KB page standard Siebel tablespace
- *32K_TABLESPACE* is the name of the 32-KB page standard Siebel tablespace
- *REPORT_FILENAME* is the name of the report generated by the utility
- *LOG_FILENAME* is the name of the log file (default: `custtbl.log`)

Example:

```
tblsize /U siebel /P siebel /C ssa /F d:\sea77\dbsrvr\DB2\ddl.ctl /B siebel_4k  
/X siebel_idx /K siebel_16k /V siebel_32k /Q d:\sba82\dbsrvr\DB2\report.txt /L  
$SIEBEL_ROOT/Log/tblsize.log
```

3 Review the report generated by the utility to determine whether the estimated table pagesize postupgrade is larger than the size of the actual custom table pagesize.

An example of the report generated by this utility is displayed in the following example:

```
Table Name = S_EVT_ACT  
Custom Tablespace Id = 5  
Custom Tablespace Name = CUST_TBS_EVT_ACT  
Custom Tablespace Pagesize = 4096  
Estimated Table Pagesize (postupgrade) = 5067  
Status = Does not fit in its custom tablespace
```

CAUTION: 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.

- 4 Move the tables from their old tablespaces to the new ones by running `ddl move`.

`ddl move` is a utility for moving tables from one tablespace to another tablespace. This utility is located in the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 5 To run `ddl move`, submit the following arguments:

```
ddl move /U TABLEOWNER /P TABLE_PASSWORD /C ODBC_DATASOURCE /E STOP_ON_DDL_ERROR
/G GRANTEE /B TABLESPACE /X INDEX_TABLESPACE /M TABLE_NAME /L LOG_FILENAME
/Z UCS2_DATABASE
```

where:

- `TABLEOWNER` is the tableowner of the database (required)
- `TABLE_PASSWORD` is the password of the tableowner of the database (required)
- `ODBC_DATASOURCE` is the data source of the database (default environment variable: `SIEBEL_DATA_SOURCE`)
- `STOP_ON_DDL_ERROR` is the stop on DDL Error parameter (default: Y)
- `GRANTEE` is the grantee for tables (`SSE_ROLE`)
- `TABLESPACE` is the name of the tablespace that you are moving the table to
- `INDEX_TABLESPACE` is the name of the index space that you are moving the table to
- `TABLE_NAME` is the Table Name Like Support value (default: N)
- `LOG_FILENAME` is the name of the log file (default: `ddlmove.log`)
- `UCS2_DATABASE` specifies whether the database uses Unicode (default: N)

Verifying the IBM DB2 Application Development Client for a Siebel Upgrade

Environments: Development, production test, production.

Databases: IBM DB2 only.

The IBM DB2 Application Development Client must be installed on the RDBMS server. Table 19 lists the required IBM DB2 Application Development Client components.

Table 19. IBM DB2 Application Development Client Components

Operating System	IBM DB2 Application Development Client Components
Microsoft Windows	DB2 Application Development Client
IBM AIX	Application Development Tools (ADT) ADT Sample Programs
HP-UX	Application Development Tools for HP-UX
Oracle Solaris	Application Development Tools (ADT) ADT Sample Programs

For information on installing the Application Development Client, see IBM documentation.

Identifying IBM DB2 Long Columns for Truncation in a Siebel Upgrade

Environments: Development, production test, production.

Databases: IBM DB2 only.

In Siebel CRM version 7.7, the maximum length for IBM DB2 long columns with a type of varchar was reduced to 16,350 from 16,383. Upgrading from version 7.5.3 truncates long varchar columns that exceed 16,350. To prevent a data truncation error that might cause transaction processing (txnproc) or transaction routing (txnroute) to fail, perform the steps in this task to identify these columns and reduce the data in these columns.

CAUTION: If you do not truncate the data in long varchar columns that exceed the maximum length specified in the following task, then a data truncated error occurs, and transaction processing and transaction routing might 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 is the username of the source database
 - SRC_PSWD is the password for the source database
 - SRC_TBL0 is the tableowner of the source database
 - SRC_TBL0_PSWD is the tableowner password for the source database
 - SRC_ODBC is the ODBC data source name of the source database (edit the value "CHANGE_ME")

- SRC_REPOSITORY_NAME is the repository name of the source database
- DBSRVR_ROOT is the directory where you installed the Siebel Database Server files on the Siebel Server, for example, C:\sba81\dbsrvr (Windows). Edit the value "CHANGE_ME".
- SIEBEL_ROOT is the directory where you installed the Siebel Server. For example, C:\sba81\siebsrvr (Windows). Edit the value "CHANGE_ME".
- VALID_RESULTS_DIR is the 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 IBM DB2 command line processor.

7

Preparing an Oracle Database for a Siebel Upgrade

This chapter provides initial preparatory information for a Siebel database upgrade on the Oracle Database. This chapter includes the following topics:

- [Verifying Oracle Database Sort Order for a Siebel Upgrade on page 99](#)
- [Setting Oracle Database Configuration Parameters for a Siebel Upgrade on page 99](#)
- [Verifying Oracle Database Parameters for Multiple Processors in a Siebel Upgrade on page 101](#)
- [Verifying the Oracle Database ODBC Definition for a Siebel Upgrade on page 101](#)

Verifying Oracle Database Sort Order for a Siebel Upgrade

Environments: Development, production test, production.

Databases: Oracle only.

The NLS_SORT parameter determines the sort order of query returns. NLS_SORT must be set to BINARY for the development environment upgrade, including the repository merge. It must also be set to BINARY when using Siebel Tools to compile an SRF file.

This setting is strongly recommended for production test environment upgrades and production upgrades.

To verify that your database is using binary sort order

- 1 Run SQLPlus to connect to the Oracle Database.
- 2 Issue the following query:

```
SQL> SELECT * FROM NLS_DATABASE_PARAMETERS;
```
- 3 In the returned parameters, locate NLS_SORT and verify that its value is BINARY.
 - If NLS_SORT has a value of BINARY, then the sort order is binary and no action is required.
 - If NLS_SORT is anything other than BINARY, then reset the value to BINARY.

Setting Oracle Database Configuration Parameters for a Siebel Upgrade

Environments: Development, production test, production.

Databases: Oracle only.

This topic provides upgrade-specific settings for the Oracle RDBMS. Use the following strategy to set parameters:

- For information about supported Oracle Database versions and any required patches, see the Certifications tab on My Oracle Support. If required, upgrade your Oracle Database and client software to the supported version, using Oracle's tools and documented procedures.
- Set parameters using the recommendations in 781927.1 (Article ID) on My Oracle Support. (This article was previously published at Siebel Technical Note 582.)
- Set additional parameters using the recommendations in *Siebel Installation Guide* for the operating system you are using. Recommendations are located in the chapter on configuring the RDBMS.
- Configure the database and set parameters as indicated in the following topics.
- After the upgrade, reset the configuration parameters to the values listed in *Siebel Installation Guide* for the operating system you are using and in 781927.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 582.

General Configuration

Before upgrading an Oracle Database, complete the following tasks:

- Verify that your Oracle clients and Oracle Database server are at the same release level.
- **UNIX only.** If you have installed the Oracle 64-bit client on the Siebel Server, then verify that \$ORACLE_HOME/lib32 instead of \$ORACLE_HOME/lib is included in LIBPATH (AIX), SHLB_PATH (HP-UX), LD_LIBRARY_PATH (Oracle 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 Business 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.

See *Siebel Performance Tuning Guide* for other parameter settings.

- **UNDO_MANAGEMENT (Oracle Database 10g and later).** Set the UNDO_MANAGEMENT parameter to 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 477025.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Alert 848.

- **Rollback Segments.** Verify that you have only one large rollback segment on line that is appropriately sized so that the largest of transactions can be accommodated. Take all other rollback segments off line.

The upgrade might affect some of the largest tables in your implementation of Siebel CRM version 7.x, causing them to grow by as much as 40%.

Customer experience has shown that repository merges involving multiple languages can require a rollback segment as large as 1 GB.

Query Optimizer Settings

For the upgrade, use the query optimizer settings listed in 781927.1 (Article ID) My Oracle Support. This document was previously published as Siebel Technical Note 582 and Siebel Alert 1011.

Verifying Oracle Database Parameters for Multiple Processors in a Siebel Upgrade

Environments: Development, production test, production.

Databases: Oracle only.

If you have multiple processors (CPUs), then verify that the following parameters are set correctly. For information on settings, refer to Oracle Database documentation:

- `parallel_max_servers`. To enable use of multiple processors, this must be set to a number greater than 1.
- `parallel_min_servers`

Verifying the Oracle Database ODBC Definition for a Siebel Upgrade

Environments: Development, production test, production.

Databases: Oracle only.

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.

Verifying or Editing an ODBC Definition on Windows

Use the following procedure to verify or edit an ODBC definition on Windows.

To verify or edit an ODBC definition on 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`

In this entry, *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, then right-click in the data display pane, and choose New, and then 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.

Verifying or Editing an ODBC Definition on UNIX

Use the following procedure to verify or edit an ODBC definition on UNIX.

To verify or edit an ODBC definition on 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:

Set the `ColumnsAsChar` value to 1.

Set the `ColumnSizeAsCharacter` to 1.

- 3 Save the file.
- 4 Stop and restart any processes that are using this `.odbc.ini` file.

8

Preparing a Microsoft SQL Server Database for a Siebel Upgrade

This chapter provides initial preparatory information for a Siebel database upgrade on the Microsoft SQL Server database. This chapter includes the following topics:

- [Verifying Microsoft SQL Server Sort Order for a Siebel Upgrade on page 103](#)
- [Setting Microsoft SQL Server Temporary Space Size for a Siebel Upgrade on page 104](#)
- [Setting Microsoft SQL Server Configuration Parameters for a Siebel Upgrade on page 104](#)
- [Rebuilding Microsoft SQL Server Clustered Indexes for a Siebel Upgrade on page 105](#)

Verifying Microsoft SQL Server Sort Order for a Siebel Upgrade

Environments: Development, production test, production.

Databases: MS SQL Server only.

Platforms: Windows only.

Verify that the sort order of the master database and the database instance are the same. This prevents repository merge failure due to sort order mismatch.

Binary sort order is required for the development environment upgrade and strongly recommended for the production environment upgrades.

When you install Microsoft SQL Server, the collation method of the database instance is set by default to dictionary sort order. Every database you create thereafter inherits this setting.

When you create a database, you can accept the inherited sort order or specify the sort order. It is recommended that you set the sort order to binary at the Microsoft SQL Server instance level so that this sort order is inherited by newly created databases.

The sort order of the master database cannot be changed without rebuilding the instance. Consult your Microsoft documentation for instructions on setting database collation.

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

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

```
sp_hel psort
```

This command provides a sort order description.

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

Lati n1_General _BI N

If you find that your Microsoft SQL Server database was not created using a binary collation sequence, then you must rebuild your database and reload your data. Review the Microsoft documentation for detailed instructions.

Setting Microsoft SQL Server Temporary Space Size for a Siebel Upgrade

Environments: Development, production test, production.

Databases: MS SQL Server only.

Platforms: Windows only.

Set the size of the database that Microsoft SQL Server uses for temporary space needed to execute queries.

To setup TEMPDB space

- 1 Make TEMPDB as big as the biggest table in the Siebel database, or half the size of the Siebel database.
- 2 Make sure that the files used by TEMPDB are configured to allow auto-growth.

This allows Microsoft 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 shri nkdatabase against TEMPDB.

Setting Microsoft SQL Server Configuration Parameters for a Siebel Upgrade

Environments: Development, production test, production.

Databases: MS SQL Server only.

Platforms: Windows only.

This topic provides upgrade-specific settings for Microsoft SQL Server. Use the following strategy to set parameters:

- Set parameters using the recommendations in *Siebel Installation Guide* for the operating system you are using. Recommendations are located in the chapter on configuring the RDBMS.

- For the upgrade, revise the configuration parameters listed in [Table 20](#).
- After the upgrade, reset the configuration parameters to the values listed in *Siebel Installation Guide* for the operating system you are using.

[Table 20](#) lists upgrade settings for Microsoft SQL Server database parameters. For parameters not listed in this table, it is recommended that you accept the default settings.

Most of the parameter settings in the table are the default settings.

Table 20. Microsoft SQL Server Configuration Parameters

Parameter	Setting
Max. degree of parallelism	1
Cost threshold for parallelism	5
Fill factor (%)	90
Index create memory (KB)	0

For the Siebel database, set the following options to ON (enabled) for the upgrade:

- **truncate log on chkpt.** Set this option to ON (enabled) for upgrade only. Also, for upgrade only, execute the `alter` command against the Siebel database, specifying `set recovery simple`.
- **torn page detection.**
- **auto create statistics.**
- **auto update statistics.**
- **Database size.** Increase your database file size by resetting the Autogrowth parameter to between 25% and 50%. Failure to do this could diminish upgrade performance and possibly impact the success of your upgrade.
- For a full list of recommended settings for your postupgrade production environment, see the chapter on configuring the RDBMS in *Siebel Installation Guide* for the operating system you are using.

Rebuilding Microsoft SQL Server Clustered Indexes for a Siebel Upgrade

Environments: Development, production test, production.

Databases: MS SQL Server only.

Platforms: Windows only.

If you have large tables that you use extensively (such as `S_EVT_ACT`, `S_CONTACT`, `S_OPTY`, `S_OPTY_POSTN`, `S_ORG_EXT`), then use the Microsoft SQL Server `create index` command with `drop_existing` clause to rebuild large tables with high fill factor (60%-70%).

9

Preparing Siebel Application Data for Upgrade

This chapter provides guidelines for preparing your Siebel Business Applications data for the Siebel database upgrade. This chapter includes the following topics:

- [Preparing Siebel Workflow Processes for Upgrade on page 107](#)
- [Preparing Siebel Customized Seed Data for Upgrade on page 108](#)
- [Migrating Siebel Household Data on page 109](#)
- [Preparing Siebel Mobile User Data for Upgrade on page 111](#)
- [Preparing Siebel Address Data for Upgrade on page 112](#)
- [Migrating Siebel Address Data from Custom Extension Columns on page 113](#)
- [Preparing Siebel Territory Management Rules for Upgrade on page 120](#)
- [Preparing Siebel Customizable Product Data for Upgrade on page 121](#)

Preparing Siebel Workflow Processes for Upgrade

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This task prepares workflows for upgrade.

CAUTION: In order to be preserved after you perform a full upgrade or run incremental repository merge, any paused workflow processes and task-based UIs must be completed first, before you perform these upgrade operations.

To prepare 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 the status of QUEUED are removed from S_SRM_REQUEST.

The upgrade process does not clear these.

Related Topics

[“Upgrade Planning for Siebel Workflow Designer” on page 76](#)

Preparing Siebel Customized Seed Data for Upgrade

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

Modified seed data are not upgraded, even if the upgraded seed data are required for an application to run normally, with the following exceptions:

- If you have modified seed data through the user interface or through Siebel Enterprise Integration Manager, then the repository merge preserves the modifications using the same logic as for other repository objects.
- As of Siebel CRM 8.0, customizations to seeded workflows in the Prior Customer Repository are merged into workflows in the New Customer Repository by the repository merge process.

Before starting the upgrade, evaluate whether you must 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.

NOTE: The paragraphs that follow provide a generalized description of how seed data for U.S. English (ENU) is imported during the upgrade process, including whether existing data is replaced. This description does not apply to all types of seed data, including seed data for non-ENU languages.

The ROW_ID value for ENU 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 column stores the date when the record was last updated.

If you modify an ENU seed data record, then 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, then the ROW_ID value does not begin with 0, and the LAST_UPD value is the record creation date.

When you perform the upprep part of the upgrade, the dataimp utility upgrades ENU seed data records as follows:

- 1 For records where the ROW_ID value begins with 0, it erases ENU seed data records where the value for LAST_UPD is 1980-01-01, unless prevented by scripting.
- 2 Dataimp replaces these records with those contained in seed data files included in the release. In the new records, the value for LAST_UPD is 1980-01-01.
- 3 Dataimp does not erase and replace seed data 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 ENU 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.

About Deactivated LOVs

If you deactivated any standard Siebel LOVs in your prior release, then you must, before executing the UPDATE statements on the S_LST_OF_VAL table, make sure to reactivate the corresponding standard Siebel LOVs through the prior release's user interface. Standard Siebel LOVs that were manually deactivated through the Siebel user interface (by changing the Active value to N) will have a ROW_ID value beginning with 0, but the LAST_UPD value will no longer be 1980-01-01. For more information, see ["Running the Siebel Database Configuration Wizard on Windows" on page 136](#) or ["Running the Siebel Database Configuration Wizard on UNIX" on page 141](#).

Discarding Seed Data Modifications

Use the following procedure to discard seed data modifications.

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 Database	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 DB2 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 SQL Server	UPDATE <i>tablename</i> SET LAST_UPD = CONVERT(DATETIME, ' 1980-01-01') WHERE ROW_ID LIKE ' 0%' AND LAST_UPD > CONVERT(DATETIME, ' 1980-01-01')

Related Topics

["About the Siebel Database Upgrade Log Files" on page 48](#)

["About the Siebel Repository Merge" on page 51](#)

["Upgrade Planning for Siebel Workflow Designer" on page 76](#)

Migrating Siebel Household Data

Upgrades: Applies to Siebel Financial Services upgrades from 7.x that have retained the Siebel 6.x form of household associations.

Environments: Production test, production.

Platforms: Windows, UNIX, IBM z/OS.

Siebel CRM version 7.0.x introduced the Party model. This changed the way relationships between households and entities, such as activity and claim, are handled.

You have two options for migrating household data:

- Migrating household relationships to the Party model (recommended)
- Retaining the Siebel 6.x form of household relationships

Migrating Household Relationships to the Party Model

To check household data integrity and support migration of household data to the Party model, you must run the household verification script (HH_MIG_populate.sql).

The script verifies that at least the same number of entities will belong to a household after the upgrade as belong to it before the upgrade.

The household verification script makes the following assumptions:

- A household has at least one contact.
- The primary contact of a Policy/Financial Account is one of the contacts associated with this Policy/Financial Account.
- The primary contact of a Claim is one of the contacts associated with this Claim.
- The primary contact of an Opportunity is one of the contacts associated with this Opportunity.
- The primary contact of a Company is one of the contacts associated with this Company.

The script populates a temporary table with data, TEMP_HH_OBJ_MIG and generates a report based on an output file. Output is in the form of row IDs. The script verifies that every household associated with an entity includes a contact associated with that entity.

If there is no output, then this means data integrity is good, and no action is required. If you receive output, then you must examine the relationship between contacts and households.

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* is the tableowner

- *Password* is the tableowner password
- *ODBCDataSource* is the data source of the database
- *OutputFileLocation* is the location of the output file:
 - Windows: *SIEBEL_ROOT\log\HH_Mi g_populate.txt*
 - UNIX: *\$SIEBEL_ROOT/log/HH_Mi g_populate.txt*
- *LogFileLocation* is the location of the log file:
 - Windows: *SIEBEL_ROOT\log\HH_Mi g_populate.log*
 - UNIX: *\$SIEBEL_ROOT/log/HH_Mi g_populate.log*
- *ScriptLocation* is the location of the script:
 - Windows: *DBSRVR_ROOT\database_platform\HH_Mi g_populate.sql*
 - UNIX: *DBSRVR_ROOT/database_platform/HH_Mi g_populate.sql*

Windows example:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /a /c REM /separator / /O  
C: \sea7xx\si ebsrvr\Log\HH_Mi g_populate.txt /L  
C: \sea7xx\si ebsrvr\Log\HH_Mi g_populate.log  
C: \sea7xx\dbsrvr\DB2\HH_Mi g_populate.sql /v y
```

- 2 If you receive output, then 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.

Preparing Siebel Mobile User Data for Upgrade

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

This topic applies primarily to developers running the Siebel 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 Siebel Mobile Web Client users.

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 Siebel Mobile Web client users for the database upgrade

- 1 Perform a partial synchronization for mobile users, sending all transactions to the server database.
- 2 Verify that Siebel Mobile Web Clients have synchronized and that all changes have been merged into the server database as follows:
 - a Check that no transaction files remain in the synchronization inbox and outbox for any mobile user. The synchronization inbox for each user is on the Siebel Server:
Windows: `SI EBEL_ROOT\docki ng\MOBI LEUSERNAME`
UNIX: `$SI EBEL_ROOT/docki ng/MOBI LEUSERNAME`
Transaction files are in the format *number.dx*; for example, 00000023. dx.
 - b Check the mobile users Remote Status view and resolve any insert conflicts.
 - c Log onto a Siebel Business Application, such as Call Center, as the Siebel administrator. Use the Tasks view from the Administration-Server Management 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, then the S_ESCL_REQ table must not have any rows.
- 3 To prevent synchronization of Siebel Mobile Web Clients with the database server, stop or disable all Siebel Remote components on all Siebel Servers.
- 4 Disconnect all Siebel Web Clients from the Siebel Server by stopping the appropriate Application Object Managers, as described in *Siebel System Administration Guide*.

Preparing Siebel Address Data for Upgrade

Upgrades:

- From Siebel CRM 8.0.x (SEA) to Siebel CRM 8.1.1.x (SIA)
- From Siebel CRM 8.1.x (SEA) to Siebel CRM 8.1.1.x (SIA)
- From Siebel CRM 7.8.2 (SEA) to Siebel CRM 8.1.1 (SIA)

Environments: Production test, production.

Databases: All databases.

At Siebel CRM version 7.7, the way address data is stored 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`

In these paths, `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, then use Siebel Enterprise Integration Manager or the application to delete unwanted records.
- 4 After addressing all the duplicate row IDs, rerun the script and verify there are no more duplicates.

Related Topics

[“Upgrade Planning for Migrating Siebel Address Data” on page 75](#)

[“Migrating Siebel Address Data from Custom Extension Columns” on page 113](#)

Migrating Siebel Address Data from Custom Extension Columns

Upgrades:

- From Siebel CRM 8.0.x (SEA) to Siebel CRM 8.1.1.x (SIA)
- From Siebel CRM 8.1.x (SEA) to Siebel CRM 8.1.1.x (SIA)
- From Siebel CRM 7.8.2 (SEA) to Siebel CRM 8.1.1 (SIA)

Environments: Development and production test.

Platforms: Windows, UNIX, IBM z/OS.

NOTE: This topic does not apply to Siebel Business Applications that you are upgrading to a later release of Siebel Business Applications.

If you have custom extension columns on the Address table `S_ADDR_ORG`, then you must manually edit your upgrade scripts to prevent loss of data during an upgrade. Perform the following task, and contact your Oracle sales representative for Oracle Advanced Customer Services if you need assistance from Oracle’s Application Expert Services.

Requirement. You must have run the Database Configuration Utilities in `upprep` 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.ctl file with a text editor.

The path to the file in the Siebel Database Server installation:

Windows: *DBSRVR_ROOT\database_platform*

UNIX: *DBSRVR_ROOT/database_platform*

In these paths, *database_platform* is the database type, for example DB2.

- 2 **Development environment upgrades only.** Add the custom extension columns from S_ADDR_PER and S_ADDR_ORG that are attributes of the association of the address to contact or account to the S_CON_ADDR definition.

Example of the file:

```
[Object nmn]
```

```
Type = Table
```

```
  Name = S_CON_ADDR
```

```
  Space = USERSPACE1
```

```
    Column 1 = ROW_ID VARCHAR(15) OTNULL
```

```
    Column 2 = CREATED TIMESTAMP NOTNULL DEFAULT %NOW%
```

```
    Column 3 = CREATED_BY VARCHAR(15) NOTNULL
```

```
    Column 4 = LAST_UPD TIMESTAMP NOTNULL DEFAULT %NOW%
```

```
    Column 5 = LAST_UPD_BY VARCHAR(15) NOTNULL
```

```
    Column 6 = DCKING_NUM NUMERIC(22, 7) DEFAULT 0
```

```
    . . . . .
```

```
    Column 46 = RURAL_ROUTE_NUM VARCHAR(10)
```

```
    Column 47 = START_DT TIMESTAMP
```

```
    Column 48 = X_CUSTOM_EXTENSION_COLUMN1 VARCHAR(15)
```

```
    Column 49 = X_CUSTOM_EXTENSION_COLUMN2 VARCHAR(15)
```

- 3 **Development environment upgrades only.** In the ddl.ctl 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_sea.sql`, add the custom extension columns for `S_CON_ADDR` to the migration scripts if any custom extension column was added to the definition of `S_CON_ADDR` in the `ddl.ctf` file in the preceding steps.

Here is the path to the file in the Siebel Database Server installation:

Windows: `DBSRVR_ROOT\database_platform\upgrade\version`

UNIX: `DBSRVR_ROOT/database_platform/upgrade/version`

The following is an example of this 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 i 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_sea.sql`, add the custom extension columns for `S_ADDR_PER` to the migration script if any custom extension column was added to the definition of `S_ADDR_PER` in the `ddl.ctf` file.

Here is the path to the file in the Siebel Database Server installation:

Windows: `DBSRVR_ROOT\database_platform\upgrade\version`

UNIX: `DBSRVR_ROOT/database_platform/upgrade/version`

Example of the file:

```
insert into S_ADDR_PER
```

```
(ROW_ID
```

```
, CREATED
```

```
, CREATED_BY
```

```
, LAST_UPD
```

```
, LAST_UPD_BY
```

```
, MODIFICATION_NUM
```

```
, CONFLICT_ID
```

```
, DISA_CLEANSER_FLG
```

```
, ADDR
```

```
, ADDR_NUM
```

```
, ADDR_TYPE_CD
```

```
, CITY
```

```
, COMMENTS
```

```
, COUNTRY
```

```
, COUNTY
```

```
, INTEGRATION_ID
```

```
, INTEGRATION2_ID
```

```
, INTEGRATION3_ID
```

```
, PROVINCE
```

```
, STATE
```

```
, ZI PCODE
, ADDR_NAME
, NAME_LOCK_FLG
, DESCR PTOR
, EMAI L_ADDR
, FAX_PH_NUM
, PH_NUM
, ADDR_LI NE_2
, ADDR_LI NE_3
, DCKI NG_NUM
, LATI TUDE
, LONGI TUDE
, ACTI VE_FLG
, ADDR_MAI L_CD
, ADDR_SUB_CD
, END_DT
, RURAL_ROUTE_NUM
, START_DT
, X_CUSTOM_EXTENSI ON_COLUMN1
, X_CUSTOM_EXTENSI ON_COLUMN2
)
sel ect
ROW_ID
, CREATED
, CREATED_BY
, LAST_UPD
, LAST_UPD_BY
, MODI FI CATI ON_NUM
, CONFLI CT_ID
, DI SA_CLEANSE_FLG
```

```

, ADDR
, ADDR_NUM
, ADDR_TYPE_CD
, CITY
, COMMENTS
, COUNTRY
, COUNTY
, INTEGRATION_ID
, INTEGRATION2_ID
, INTEGRATION3_ID
, PROVINCE
, STATE
, ZIPCODE
, case when ADDR_NAME = (select t.DUP_ADDR_NAME
    from    TMPTBL_ADDR t
    where t.DUP_ADDR_NAME = S_ADDR_ORG.ADDR_NAME and t.MIN_ROW_ID <>
    S_ADDR_ORG.ROW_ID
        )
    then {fn concat({fn concat({fn rtrim({fn substring(ADDR_NAME, 1, 99-
    {fn length(ROW_ID)}})}), ' : ')}, ROW_ID)}
    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.

Related Topics

[“Upgrade Planning for Migrating Siebel Address Data” on page 75](#)

[“Preparing Siebel Address Data for Upgrade” on page 112](#)

Preparing Siebel Territory Management Rules for Upgrade

Upgrades from: Siebel 7.8.x.

Environments: Production test, production.

Territory rules management was introduced in Siebel CRM 7.8. As of Siebel CRM 8.0, territory rules cannot have overlapping effective date ranges. For example, the following two rules have overlapping effective date ranges:

Rule 1

Territory: A100

Account: XYZ Corp

Effective Start Date: 1/1/2006

Eff. End Date: 6/30/2006

Rule 2

Territory: A100

Account: XYZ Corp

Effective Start Date: 4/1/2006

Eff. End Date:

You must identify rules with overlapping effective date ranges and modify the rules to eliminate any overlap.

To prepare territory rules for upgrade

- 1 Run the following script against the database:

Windows: `DBSRVR_ROOT\common\TM_DupRuleCheck.sql`

UNIX: `DBSRVR_ROOT/common/TM_DupRuleCheck.sql`

The script returns a list of all territory rules that have overlapping effective date ranges.

- 2 In Territory Management, revise rule effective date ranges to remove the overlaps.

Preparing Siebel Customizable Product Data for Upgrade

Environments: Production test, production.

Customizable Products in Work Spaces

The upgrade does not migrate unreleased customizable products in work spaces. If you want to migrate unreleased customizable products, then 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.

10 Upgrading the Siebel Database

This chapter provides guidelines for performing a Siebel database upgrade. This chapter includes the following topics:

- [Example of a Siebel Development Environment Upgrade Flow on page 123](#)
- [Renaming the Siebel Repository on page 129](#)
- [Copying Ancestor Repositories on page 130](#)
- [Changing the Siebel Database Configuration Utilities Language on page 132](#)
- [Preparing to Run the Siebel Database Configuration Wizard on page 132](#)
- [Running the Siebel Database Configuration Wizard on Windows on page 136](#)
- [Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on Windows on page 138](#)
- [Running the Siebel Database Configuration Wizard on UNIX on page 141](#)
- [Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on UNIX on page 143](#)
- [Starting the Siebel Upgrade Wizard on page 145](#)
- [Moving the Siebel Repository Files on page 149](#)
- [Running the Siebel Case Insensitivity Wizard on page 149](#)
- [Applying Siebel Additive Schema Changes on page 154](#)
- [Regenerating SQL Files for a Siebel Upgrade on page 155](#)
- [Identifying and Dropping Obsolete Indexes for a Siebel Upgrade on page 156](#)
- [Preparing for a Nondevelopment Environment Siebel Upgrade on page 157](#)

Example of a Siebel Development Environment Upgrade Flow

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) 8.0 to Siebel SIA 8.1
- Environment: Development
- Upgrade mode: upgrep
- Database: Oracle Database
- Multilingual: No

Table 21 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 21. Example of Steps in a Development Environment Upgrep

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

Table 21. Example of Steps in a Development Environment Upgprep

Step	Script or Input File	Comment
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	preschm.sql	Performs DML operations. Moves data between tables. Changes data in existing fields based on specified conditions.
Prepare for index creation for SEA	preschm_sea.sql	Same as preschm.sql.
Prepare for index creation for SIA	preschm_sia.sql	Same as preschm.sql.
Create indexes	<ul style="list-style-type: none"> ■ ddlimp utility ■ ddl.ctl as input 	The input file specifies the structure of indexes to be created.
Delete old license key	delappkey.sql	Deletes the Siebel license key from S_APP_KEY.
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 that 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.

Table 21. Example of Steps in a Development Environment Upgrep

Step	Script or Input File	Comment
Upgrade data after seed data import SIA	upg_data_afterseed_sia.sql	None.
Set system preference for codepage for DB	set_codepage.sql	Sets the database codepage in the 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	Updates the S_UPG_COMP table with the product release level. Also updates location of the SRF file. The S_UPG_COMP table stores version information for application executables.
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 8.1.x, then this command imports the standard Siebel 8.1.x repository.
Remove EIM columns and indexes	rmv_anc_eim_proc_col_ind.sql	<p>Removes Siebel Enterprise Integration Manager processing columns and indexes from the Prior Customer Repository and the common ancestor repository (Standard Siebel Repository). This prevents the repository merge from preserving Siebel Enterprise Integration Manager columns incorrectly.</p> <p>The merge will preserve only those Siebel Enterprise Integration Manager columns shipped with the new release.</p>

Table 21. Example of Steps in a Development Environment Upgrade

Step	Script or Input File	Comment
Update Siebel database version	<ul style="list-style-type: none"> ■ update_ver.sql ■ seeduver.sql 	The update_ver.sql script creates a temporary table, S_APP_VER_TEMP, which contains new version information for the database schema. The seeduver.sql script updates S_APP_VER with this information.
Import new Siebel Repository	<ul style="list-style-type: none"> ■ repimexp utility ■ Imports New Siebel Repository 	Imports the New Siebel Repository into S_REPOSITORY. Revises schema version information in S_APP_VER.
Import New Customer Repository	<ul style="list-style-type: none"> ■ repimexp utility ■ Imports New Customer Repository 	Imports the New Customer Repository into S_REPOSITORY. Revises schema version information in S_APP_VER.
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	None.
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	None.

Table 21. Example of Steps in a Development Environment Upgrep

Step	Script or Input File	Comment
Install SQL packages	seeduver.sql	Verifies that versions are set correctly in S_APP_VER.
	ifstrg.sql	Sets storage parameters for Siebel Enterprise Integration Manager tables.
	ifindxstrg.sql	Sets storage parameters for Siebel Enterprise Integration Manager 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.
Create database-level functions and procedures	db_func_proc.sql	Creates or replaces the currency exchange rate function: exrate.

Table 21. Example of Steps in a Development Environment Upgred

Step	Script or Input File	Comment
Set primary children in data tables	gen_primary1.sql	Sets primary child for S_DOC_QUOTE.
	gen_primary2.sql	None.
	gen_primary3.sql	Sets primary child for S_LOY_PROMO.
	gen_primary4.sql	None.

Related Topics

[“About the Siebel Database Configuration Utilities” on page 37](#)

[“About the Siebel Upgrade Wizard and Driver Files” on page 43](#)

Renaming the Siebel Repository

Environments: Development environment only. (Also Production environment on UNIX)

Platforms: Windows, UNIX, IBM z/OS.

NOTE: 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*.

To rename the repository

- 1 Start Siebel Tools and connect to the Siebel database.
Use the version of Siebel Tools for the Siebel CRM 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, then import these archive files back into the repository.
If you do not check these objects back into the repository, then 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 From the View menu, choose 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 the name and change it to Pri or Customer Repository.
For more information about renaming repositories, see *Configuring Siebel Business Applications*.
- 10 Step off the list to commit the record to the database.
If the validation check fails, then verify that you have renamed the repository correctly.

Copying Ancestor Repositories

This task is a step in [“Process of Meeting the Requirements for an Incremental Upgrade to Siebel Innovation Pack 2016, Using the Incremental Repository Merge” on page 187](#). You also perform this task for other repository upgrades.

Use this task to copy the ancestor repositories before beginning your upgrade. Ancestor repository files are located in an Ancestor Repositories media directory that you access from Oracle Software Delivery Cloud or My Oracle Support. Where these files are made available depends on the type of release.

The ancestor repository relevant to your upgrade is the Siebel repository that was shipped with the Siebel CRM release that you are upgrading from. It contains none of your customizations.

NOTE: This guide describes upgrading to Siebel Innovation Pack 2016 (Siebel CRM version 16.0). This release and the associated ancestor repository media are available on Oracle Software Delivery Cloud or My Oracle Support. Use the following procedure to obtain Siebel CRM Version 16.0 Ancestor Repository media.

- For upgrades of Siebel CRM version 15.5 or later to version 16.0 using Incremental Repository Merge:
 - If your repository version is version 15.5, then the ancestor repository version is 15.5 (Siebel Innovation Pack 2015, Patchset 5 or later).
- For upgrades of Siebel CRM version 15.0 to version 16.0 using Incremental Repository Merge:
 - If your repository version is version 15.0, then the ancestor repository version is 15.0 (Siebel Innovation Pack 2015).
- For upgrades of Siebel CRM version 8.1.1.x to version 16.0 using Incremental Repository Merge:
 - If your repository version is version 8.1.1.14, then the ancestor repository version is 8.1.1.14 (Siebel Innovation Pack 2014).
 - If your repository version is version 8.1.1.11, then the ancestor repository version is 8.1.1.11 (Siebel Innovation Pack 2013).
 - If your repository version is version 8.1.1.10, then the ancestor repository version is version 8.1.1.10.

- If your repository version is anything prior to version 8.1.1.10, then the ancestor repository is version 8.1.1.
- For upgrades of Siebel CRM version 8.2.2.x to version 16.0 using Incremental Repository Merge:
 - If your repository version is version 8.2.2.14, then the ancestor repository version is 8.2.2.14 (Siebel Innovation Pack 2014).
 - If your repository version is version 8.2.2.4, then the ancestor repository version is 8.2.2.4 (Siebel Innovation Pack 2013).
 - If your repository version is version 8.2.2.3, then the ancestor repository version is version 8.2.2.3.
 - If your repository version is anything prior to version 8.2.2.3, then the ancestor repository is version 8.2.2.

To copy the ancestor repositories

- 1 To download ancestor repositories from Oracle Software Delivery Cloud, do the following:
 - a For Siebel CRM Version 16.0, access the Siebel CRM Version 16.0 Ancestor Repository media, part number V137750-01.
 - b Proceed to [Step 3](#).
- 2 To download ancestor repositories from My Oracle Support, do the following:
 - a From My Oracle Support, click the Patches & Updates tab.
 - b Enter patch ID 23092685 in the Search box to access the ancestor repository media for Siebel CRM version 16.0.
- 3 After downloading, extract the included file to its own unique folder.

NOTE: Keep these files in a separate directory from other downloaded JAR files.

- 4 From the newly created directory where the ancestor repository media files are located, run the following script to copy the ancestor repositories to your Siebel root directory:

- Windows: copydat.bat (copydat.bat *SIEBEL_ROOT optional_log_file*)

For example, if your Siebel Database Server installation location is c:\siebel\8.1.1.x\ses\dbsrvr, then you might run the following:

```
copydat.bat c:\siebel\8.1.1.x\ses d:\temp\out.log
```

- UNIX: `copydat.sh (sh copydat.sh $SIEBEL_ROOT optional_log_file)`

For example, if your Siebel Database Server installation location is `/export/home/22313/ses/dbsrvr`, then you might run the following:

```
sh copydat.sh /export/home/22313/ses /tmp/out.log
```

The script copies the ancestor repository files `r782.dat`, `r80.dat`, `r811.dat`, `r81110.dat`, `r82.dat`, `r821.dat`, `r822.dat`, `r8223`, `r2013.dat`, `r2014.dat`, `r150.dat`, and `r155.dat` to `$SIEBEL_ROOT/dbsrvr/common/SIA` and copies `hor782.dat`, `hor80.dat`, and `hor811.dat` to `$SIEBEL_ROOT/dbsrvr/common/HOR`. After the script has been successfully run, the Checksum verification is run on the copied files against the checksum values shipped in the manifest file.

NOTE: If a log file location is not specified, then a log file is generated in the directory that you run the script from or, if that directory is not writable, in your temp directory.

Changing the Siebel Database Configuration Utilities Language

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

The Database Configuration Utilities launch in the language selected when you ran the Siebel Enterprise Server Installer. You can change the language in which the utilities run, if desired, from the language chosen during installation.

To change the Database Configuration Utilities language, see the *Siebel Installation Guide* for the operating system you are using.

If you want an additional language to appear in the language list in the Database Configuration Utilities, then you first must install the appropriate language pack on the database server and on the Siebel Server. For information about installing additional language packs, see the *Siebel Installation Guide* for the operating system you are using.

Related Topics

[“About the Siebel Database Configuration Utilities” on page 37](#)

Preparing to Run the Siebel Database Configuration Wizard

Environments: Development, production test, production.

Use this topic to identify the information you must enter when running the Database Configuration Utilities. Collect this information, and verify that it is correct before running the utilities.

NOTE: If you are running Siebel CRM version 8.1.1 (SEA), and want to migrate to Siebel CRM version 8.1.1 (SIA), then you do so with the Database Configuration Wizard.

The Database Configuration Utilities are a group of wizards that request information about the upgrade process you want to perform. The utilities add this information to a master upgrade file and call an SQL generator. The SQL generator uses the information to create or populate SQL files:

- [Table 22 on page 133](#) lists the information that the utilities request for performing an upgrade.
- [Table 23 on page 136](#) lists the additional information specific to Additive Schema Changes mode.

After the Database Configuration Wizard exits you run the Upgrade Wizard. The Upgrade Wizard executes the SQL files against the Siebel database.

Table 22. Information Required for the Database Configuration Utilities

Field Name or Menu	Required Information
Gateway Name Server Address	The Siebel Gateway Name Server computer name.
Enterprise Server Name	The Enterprise Server name.
Siebel Server Directory	The absolute path of the directory where the Siebel Server is installed. For UNIX, do not enter the string \$SIEBEL_ROOT.
Siebel Database Server Directory	The absolute path of the directory where the Siebel Database Server is installed. For example: C:\sba81\dbsrvr
RDBMS Platform	Choose the RDBMS type: IBM DB2, Microsoft SQL Server, or Oracle Database. NOTE: For IBM DB2 for z/OS, see <i>Siebel Database Upgrade Guide for DB2 for z/OS</i> instead of this guide.
Siebel Database Operation menu	For upgrep, upgphys and Prepare for Production modes, choose Upgrade Database. For Apply Additive Schema Changes mode, choose Apply Additive Schema Changes. The remaining menu choices are for database installation and administration.
Environment Type	Choose Development for development environment upgrades. Choose Production for production test environment and production environment upgrades.
Upgrade Options	Choose one of the following: <ul style="list-style-type: none"> ■ Development Environment: <ul style="list-style-type: none"> ■ Upgrade Siebel Database Schema (upgrep) ■ Upgrade Custom Database Schema (upgphys) ■ Production Environment: <ul style="list-style-type: none"> ■ Upgrade Siebel Database Schema (upgrep and upgphys)

Table 22. Information Required for the Database Configuration Utilities

Field Name or Menu	Required Information
Siebel Industry Application	As appropriate for the environment you are upgrading from, choose SIA (for Siebel Industry Applications) or SEA (for Siebel Business Applications, also known as Siebel Cross-Industry Applications).
Current Siebel Version	Choose the option for the application version that you are upgrading your Siebel database from. For more information about valid options for performing a full database upgrade, see “Supported Upgrade Paths for Siebel CRM Version 16.0” on page 17. For information about specifying options for performing an Incremental Repository Merge instead, see “Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x (SIA Repository) and Version 15.x” on page 190.
Database Encoding	Indicate whether your database uses a Unicode code page.
ODBC Data Source Name	Verify the ODBC name for connecting to the Siebel database that you are upgrading. If the ODBC name is not correct, then enter the correct name.
Database User Name and Database Password	Account name and password for the Siebel administrator of the Siebel database that you are upgrading. NOTE: For more information about supported characters for Siebel passwords, see <i>Siebel Security Guide</i> . See also relevant articles on My Oracle Support, such as 1931241.1 (Article ID).
Database Table Owner and Database Table Owner Password	Account name and password for the Siebel database table owner. NOTE: For more information about supported characters for Siebel passwords, see <i>Siebel Security Guide</i> . See also relevant articles on My Oracle Support, such as 1931241.1 (Article ID).
Index Table Space Name and Table Space Name	Oracle Database and IBM DB2 only. Index tablespace name and tablespace name (4-KB tablespace name for IBM DB2).
16-KB Table Space Name, 32K Table Space Name	IBM DB2 only. The 16-KB and 32-KB tablespace names.
Database Server OS	Choose the RDBMS server operating system type.

Table 22. Information Required for the Database Configuration Utilities

Field Name or Menu	Required Information
Parallel Indexing	<p>Oracle Database only. Select parallel indexing if you want SQL commands for index creation to include the arguments parallel and no logging.</p> <p>Parallel indexing causes an index to be created using parallel processing, which requires an RDBMS server with multiple processors. Verify with your database administrator whether 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 set up parallel indexing, you must set up parallel index-creation threads using Siebel Upgrade Tuner. You create parallel threads as part of tuning the production upgrade files. For more information, see Chapter 18, “Tuning the Siebel Upgrade Files.”</p>
Security Group ID/Grantee	Security group or grantee name for Siebel application users. Must have select, update, insert, and delete privileges on Siebel application tables. Specify SSE_ROLE.
Verify Repository After Upgrade	Indicate whether you want to execute the steps to verify the repository during upgphys. To perform upgphys separately, select the Verify Repository After Upgrade option in the Database Configuration Wizard.
Upgrep log directory	If you select the option Verify Repository After Upgrade in the previous step, then you will have to provide the log directory of the upgrep process. The log directory is of the form <code>\$SIEBEL_ROOT/si ebsrvr/l og/upgrep_dev_UpgradeNumber</code> . For example: <code>C:\ses\si ebsrvr\l og\upgrep_dev_811</code> . The l og directory path is a requirement for generating the seed data conflict report.
Log Output Directory	Specify a different subdirectory, under the l og directory, in which to create the log files. Accept the default or enter the directory name. If the directory does not exist, then it will be created. Do not use special characters, such as spaces or slashes.
Select runupg option	Indicate whether you want to run the operation you configured or run it at another time.

Additional Information Required for Apply Additive Schema Changes Mode

You can optionally run the Database Configuration Utilities in Apply Additive Schema Changes mode. Additive Schema Changes generates an SQL script, `schema.additive.sql`. The script contains production upgrep schema changes that can be performed while the database is online. This reduces the amount of time required for the upgrep when the database is offline.

The additional information shown in [Table 23](#) is required when you run the Database Configuration Utilities in Apply Additive Schema Changes mode.

Table 23. Additional Information Required for Apply Additive Schema Changes Mode

Field Name	Required Information
Schema File	The absolute path to the schema.ddl file. If a path is displayed, then verify that schema.ddl is located in this directory.
Output Directory	The absolute path to the directory where schema.additive.sql will be placed after the Upgrade Wizard generates it. The Upgrade Wizard also places the log file in this directory.

Related Topics

[“About the Siebel Database Configuration Utilities” on page 37](#)

[“About Siebel Additive Schema Changes Mode” on page 46](#)

Running the Siebel Database Configuration Wizard on Windows

Environments: Development, production test, production.

Platforms: Windows only.

For a description of information required to run the Database Configuration Wizard, see [Table 22 on page 133](#).

Run the Database Configuration Wizard to upgrade the Siebel database. The Wizard collects information, populates a master configuration file, and calls the SQL generator to create SQL commands. The Upgrade Wizard then uses the configuration file and SQL commands to upgrade the Siebel database.

Requirements

- Collect the information that the Database Configuration Wizard requires. See [“Preparing to Run the Siebel Database Configuration Wizard” on page 132](#).
- Install the new release’s languages packs for all deployed languages. See [“Upgrade Planning for Multilingual Siebel Deployments” on page 69](#).

- Run the Database Configuration Wizard as described in this topic. All customers must perform this procedure. This procedure allows you to determine whether you have deployed languages that are not shipped with the Siebel product, or have unintended languages within your system that must be removed. For a list of shipped languages, see 1513102.1 (Article ID) on My Oracle Support.

NOTE: If this procedure fails due to the presence of unshipped languages within your system, then you will receive an error message stating that your present installation was found to be incomplete. You are also displayed a list of languages which caused the error.

If your system only contains shipped languages, and you still receive this error message, then you must review the records in the S_LST_OF_VAL database table using Siebel Tools. For more information, see [“Preparing for a Multilingual Upgrade” on page 86](#).

- If applicable, perform the tasks in [“Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on Windows” on page 138](#).

CAUTION: Perform the tasks in this process only if your configuration operation failed due to unshipped languages being deployed within your system.

This process allows you to run the configuration utility, validate all data in shipped languages, and pass over all data in unshipped languages. You can manually verify the data in unshipped languages following the successful completion of the Database Configuration Wizard.

For more information about starting the Database Configuration Wizard, see the *Siebel Installation Guide for Microsoft Windows*.

To run the Database Configuration Wizard on Windows

- 1 Ensure that no server tasks except the Siebel Gateway Name Service are running in the background.

To verify, navigate to Start, Settings, Control Panel, and then Services.

NOTE: The Siebel Gateway Name Server must be running for the Database Configuration Wizard to execute.

- 2 Start the Database Configuration Wizard by selecting Start, Programs, Siebel SES Configuration, and then Configure DB Server.

The first window of the Database Configuration Wizard appears.

- 3 Enter the information requested in each screen and click Next.
- 4 After you have entered all the requested information, the wizard displays a screen that lists the values you entered. If you must make changes, then click Back.
- 5 When the window displays inquiring whether you want to start the Upgrade Wizard, do the following:
 - **Development Uppreg mode:** Answer No. Do not start the Upgrade Wizard until you have performed all requirements.
 - **Development Uppphys mode:** Answer Yes to start the Upgrade Wizard.

- **Combined Production Upgrep and Upgphys mode:** Answer No. Do not start the Upgrade Wizard until you have performed all requirements.
- **Additive Schema Changes:** You will not be prompted. The Upgrade Wizard starts automatically and creates the schema.additive.sql script.
- **Prepare for Production mode:** Answer Yes to start the Upgrade Wizard.

Just before displaying the prompt, the wizard calls the SQL generator to create or populate SQL scripts.

Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on Windows

To delete unshipped languages from the S_LST_OF_VAL table on Windows, perform the following tasks.

CAUTION: Perform the tasks in this process only if your configuration operation failed due to unshipped languages being deployed within your system.

- [“Backing Up the S_LST_OF_VAL Table on Windows” on page 138](#)
- [“Deactivating Records for Unshipped or Unwanted Languages on Windows” on page 139](#)
- [“Importing Newly Created Records from the Data File on Windows” on page 140](#)

Backing Up the S_LST_OF_VAL Table on Windows

This task is a step in [“Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on Windows” on page 138](#). You back up your original S_LST_OF_VAL table before you begin deleting obsolete or unshipped languages. You can later import languages from this data file.

To back up the S_LST_OF_VAL table on Windows

- 1 In Siebel Tools, back up the original S_LST_OF_VAL table.
- 2 From the Windows command prompt, navigate to the siebsrvr\bin directory.
- 3 From this directory, execute a command like the following to run the Siebel dataexp utility (replacing \$ tokens with actual values that represent the environment):

```
dataexp /u $UserName /p $Password /c "$OBCDataSource" /d $SiebelTableOwner /f $DataFileToExport.dat /l $LogFile.log
```

The utility connects to the database, and displays the following prompt:

```
Connecting to the database...
```

```
Connected.
```

```
Exporting Tables
```

```
Enter table name:
```

- 4 Type S_LST_OF_VAL, and hit Enter.

When the action is completed, the utility will display the following message:

```
Exporting Tables
```

```
Enter table name: S_LST_OF_VAL
```

```
Reading tables matching "S_LST_OF_VAL" ...
```

```
Exporting table S_LST_OF_VAL ... exported 30102 rows
```

```
Enter table name:
```

- 5 Close the command window and check the log file for errors, to verify that the data file was created.

Deactivating Records for Unshipped or Unwanted Languages on Windows

This task is a step in [“Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on Windows” on page 138](#). You render unshipped or obsolete languages inactive.

To deactivate records for unshipped or unwanted languages on Windows

- 1 Connect to the database using a database client that allows you to run SQL interactively (either one of the native clients, or Siebel odbcsql), and execute the following query:

```
update S_LST_OF_VAL set ACTIVE_FLG = 'N'  
where LANG_ID not in ('ENU', 'lang_code1', ..., 'lang_codeN');
```

NOTE: Be aware that *lang_code** tokens must be replaced with actual codes of the languages that are going to be upgraded (such as FRA, DEU, and so on).

CAUTION: You must not deactivate ENU records, even if you do not have the ENU language deployed in your environment. Some system records in S_LST_OF_VAL are ENU.

- 2 Choose commit (if your database client is not in auto-commit mode), and execute the following query:

```
delete from S_LST_OF_VAL where ACTIVE_FLG = 'N';
```

- 3 Choose commit and exit the database client.

Importing Newly Created Records from the Data File on Windows

This task is a step in “[Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on Windows](#)” on page 138. Import the data file that you created when you backed up the S_LIST_OF_VAL table. This procedure is required to upgrade additional languages (those that were not available or were skipped during the primary upgrade).

Note the following:

- Depending on which data you have exported, how you set up the input file, and which options you use on the command line, you can import data for multiple languages at once or for one language at a time, or import data based on other criteria.
- In order to import data for only particular languages, for example, you must first use filtering mechanisms when you export the data, so that only those languages are to be included in the data file.
- If you use the /e N command-line option for dataimp, then dataimp only imports the data that was originally exported from the tables specified in the input file. If you instead use /e Y, then the dataimp imports the contents of the entire data.

To import newly created records from the data file on Windows

- 1 Create a text file that will serve as the input file for the Siebel dataimp utility.

The contents of the file must be formatted as follows (replacing *lang_code* tokens with actual values that represent the environment):

```
S_LST_OF_VAL  
  
where LANG_ID in ( 'lang_code1' , . . . , 'lang_codeN' )
```

- 2 Save the file with the extension .INP.
- 3 From the Windows command prompt window, navigate to the si ebsrvr\bin directory.
- 4 From the si ebsrvr\bin directory, execute a command like the following to run the Siebel dataimp utility (replacing \$ tokens with actual values that represent the environment):

```
dataimp /u $UserName /p $Password /c "$ODBCDataSource" /d $SiebelTableOwner /f  
$DataFileName.dat /h Log /e N /i $InputFileName.inp /A 1
```

- 5 Check the log file for errors.
- 6 Connect to the database using a database client that allows you to run SQL interactively, to verify that the imported records are there.

For example, you execute the following query:

```
select count (LANG_ID), ACTIVE_FLG from S_LST_OF_VAL where LANG_ID = 'lang_code'  
group by ACTIVE_FLG;
```

If the result set returns both active and inactive records, then you must investigate and see why some records have been deactivated, and either re-activate or delete these inactive records.

Related Topics

[“About the Siebel Database Configuration Utilities” on page 37](#)

[“Starting the Siebel Upgrade Wizard” on page 145](#)

Running the Siebel Database Configuration Wizard on UNIX

Environments: Development, production test, production.

Platforms: UNIX only.

Run the Database Configuration Wizard to upgrade the Siebel database. The Wizard collects information, populates a master configuration file, and calls the SQL generator to create SQL commands. You then run the Upgrade Wizard to upgrade the Siebel database.

Prior to running the Database Configuration Wizard you must install the new release’s languages packs for each language that you deploy.

Requirements

- Collect the information that the Database Configuration Wizard requires. See [“Preparing to Run the Siebel Database Configuration Wizard” on page 132](#).
- Install the new release’s languages packs for all deployed languages. See [“Upgrade Planning for Multilingual Siebel Deployments” on page 69](#).
- Run the Database Configuration Wizard as described in this topic. All customers must perform this procedure. This procedure allows you to determine whether you have deployed languages that are not shipped with the Siebel product, or have unintended languages within your system that must be removed. For a list of shipped languages, see 1513102.1 (Article ID) on My Oracle Support.

NOTE: If this procedure fails due to the presence of unshipped languages within your system, then you will receive an error message stating that your present installation was found to be incomplete. You are also displayed a list of languages which caused the error.

If your system only contains shipped languages, and you still receive this error message, then you must review the records in the S_LST_OF_VAL database table using Siebel Tools. For more information, see [“Preparing for a Multilingual Upgrade” on page 86](#).

- If applicable, perform the tasks in [“Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on UNIX” on page 143](#).

CAUTION: Perform the tasks in this process only if your configuration operation failed due to unshipped languages being deployed within your system.

This process allows you to run the configuration utility, validate all data in shipped languages, and pass over all data in unshipped languages. You can manually verify the data in unshipped languages following the successful completion of the Database Configuration Wizard.

For more information about starting the Database Configuration Wizard, see the *Siebel Installation Guide for UNIX*.

Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on UNIX

To delete unshipped languages from the S_LST_OF_VAL table on UNIX, perform the following tasks.

CAUTION: Use the tasks in this process only if your configuration operation failed due to unshipped languages being deployed within your system.

- “Backing Up the S_LST_OF_VAL Table on UNIX” on page 143
- “Deactivating Records for Unshipped or Unwanted Languages on UNIX” on page 144
- “Importing Newly Created Records from the Data File on UNIX” on page 144

Backing Up the S_LST_OF_VAL Table on UNIX

This task is a step in “Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on UNIX” on page 143. You back up your original S_LST_OF_VAL table before you begin deleting obsolete or unshipped languages. You can later import languages from this data file.

To back up the S_LST_OF_VAL table on UNIX

- 1 In Siebel Tools, back up the original S_LST_OF_VAL table.
- 2 On the Siebel Server computer, navigate to the `SIEBSRVR_ROOT\bin` directory.
- 3 From this directory, execute a command like the following to run the Siebel dataexp utility (replacing \$ tokens with actual values that represent the environment):

```
dataexp /u $UserName /p $Password /c "$ODBCDataSource" /d $SiebelTableOwner /f $DataFileToExport.dat /l $LogFile.log
```

The utility connects to the database, and displays the following prompt:

```
Connecting to the database...
```

```
Connected.
```

```
Exporting Tables
```

```
Enter table name:
```

- 4 Type S_LST_OF_VAL, and hit Enter.

When the action is completed, the utility will display the following message:

```
Exporting Tables
```

```
Enter table name: S_LST_OF_VAL
```

```
Reading tables matching "S_LST_OF_VAL" ...
```

```
Exporting table S_LST_OF_VAL ... exported 30102 rows
```

Enter table name:

- 5 Check the log file for errors, to verify that the data file was created.

Deactivating Records for Unshipped or Unwanted Languages on UNIX

This task is a step in [“Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on UNIX” on page 143](#). You render unshipped or obsolete languages inactive.

To deactivate records for unshipped or unwanted languages on UNIX

- 1 Connect to the database using a database client that allows you to run SQL interactively (either one of the native clients, or Siebel odbcsql), and execute the following query:

```
update S_LST_OF_VAL set ACTIVE_FLG = 'N'
where LANG_ID not in ('ENU', 'lang_code1', ..., 'lang_codeN');
```

NOTE: Be aware that *lang_code** tokens must be replaced with actual codes of the languages that are going to be upgraded (such as FRA, DEU, and so on).

CAUTION: You must not deactivate ENU records, even if you do not have the ENU language deployed in your environment. Some system records in S_LST_OF_VAL are ENU.

- 2 Choose commit (if your database client is not in auto-commit mode), and execute the following query:

```
delete from S_LST_OF_VAL where ACTIVE_FLG = 'N';
```

- 3 Choose commit and exit the database client.

Importing Newly Created Records from the Data File on UNIX

This task is a step in [“Process of Deleting Unshipped Languages from the S_LST_OF_VAL Table on UNIX” on page 143](#). Import the data file that you created when you backed up the S_LIST_OF_VAL table. This procedure is required to upgrade additional languages (those that were not available or were skipped during the primary upgrade).

Note the following:

- Depending on which data you have exported, how you set up the input file, and which options you use on the command line, you can import data for multiple languages at once or for one language at a time, or import data based on other criteria.
- In order to import data for only particular languages, for example, you must first use filtering mechanisms when you export the data, so that only those languages are to be included in the data file.

- If you use the `/e N` command-line option for `dataimp`, then `dataimp` imports data only from the tables that are specified in the input file. If you instead use `/e Y`, then the contents of the entire data file are imported.

To import newly created records from the data file on UNIX

- 1 Create a text file that will serve as the input file for the Siebel `dataimp` utility.

The contents of the file must be formatted as follows (replacing `lang_code` tokens with actual values that represent the environment):

```
S_LST_OF_VAL
where LANG_ID in (' lang_code1' , ... , ' lang_codeN')
```

- 2 Save the file with the extension `.INP`.
- 3 Navigate to the `si ebsrvr/bin` directory.
- 4 From this directory, execute a command like the following to run the Siebel `dataimp` utility (replacing `$` tokens with actual values that represent the environment):

```
dataimp /u $UserName /p $Password /c "$ODBCDataSource" /d $SiebelTableOwner /f
$DataFileName.dat /h Log /e N /i $InputFileName.inp /A 1
```

- 5 Check the log file for errors.
- 6 Connect to the database using a database client that allows you to run SQL interactively, to verify that the imported records are there.

For example, you execute the following query:

```
select count (LANG_ID), ACTIVE_FLG from S_LST_OF_VAL where LANG_ID = ' lang_code'
group by ACTIVE_FLG;
```

If the result set returns both active and inactive records, then you must investigate and see why some records have been deactivated, and either re-activate or delete these inactive records.

Related Topics

[“About the Siebel Database Configuration Utilities” on page 37](#)

[“Starting the Siebel Upgrade Wizard” on page 145](#)

Starting the Siebel Upgrade Wizard

Environments: Development, production test, production.

The Siebel Upgrade Wizard executes the upgrade of the Siebel database. It takes a master configuration file as input. This file contains environment information and a driver file name. The Upgrade Wizard executes the steps in the driver file to perform the upgrade.

As the Upgrade Wizard performs the steps in the driver file, it lists the steps in a state log. The state log is located in `si ebsrvr/LOG/process/state`, where `process` is the upgrade process, for example `upgrep_prod_811` (upgrade from 8.1.1, `upgrep` process, production test or production environment).

If the Upgrade Wizard encounters an error and exits during an upgrade, then you can restart it after correcting the error. The Upgrade Wizard reads the state log and continues the upgrade from the last successfully completed step.

When you run the Database Configuration Utilities on Windows, they will prompt you if you want to start the Upgrade Wizard. When you run the Database Configuration Utilities on UNIX, you must start the Upgrade Wizard manually.

Requirements for Restarting the Upgrade Wizard

If the Upgrade Wizard stops due to errors, then verify that you have met these requirements before restarting the wizard:

- Carefully review the relevant log files to make sure that your upgrade has completed successfully up to that point.
- Back up your complete set of log files, from the beginning of the process to the point at which it stopped, to another directory.

This backup maintains a complete record of your log files, and prevents your previous log files from being overwritten, which could prevent accurate diagnosis of the reason for the break in the upgrade.

- If you are continuing a previous and incomplete schema upgrade, then do not change the Log Output Directory that you previously selected.
- If problems with your environment prevent the upgrade from restarting, then you must restore the database from the prior base version (the version from which you are upgrading). For example, environment problems might occur when table creation fails due to a database problem (insufficient storage or network problems), which cause subsequent upgrade steps to fail.

If you must restore your database and restart the upgrade, then delete or store the upgrade log files. The files are located in the following directory:

Windows: `SI EBEL_ROOT\log\PROCESS\output`

UNIX: `$SI EBEL_ROOT/log/PROCESS/output`

Also delete the state. log file. It is located in the following directory:

Windows: `SI EBEL_ROOT\log\PROCESS\state`

UNIX: `$SI EBEL_ROOT/log/PROCESS/state`

Starting the Upgrade Wizard

Use this procedure to start the Upgrade Wizard. See the following topics to stop the Upgrade Wizard, [“Stopping the Upgrade Wizard on Windows” on page 147](#) and [“Stopping the Upgrade Wizard on UNIX” on page 148](#).

To start the Upgrade Wizard

- 1 Navigate to the following directory:

Windows: *SIEBEL_ROOT\bin*

UNIX: *\$SIEBEL_ROOT/bin*

- 2 Enter the following command:

Windows: `siebug /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`

UNIX: `svrupgwiz /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`

In these commands, *UPGRADEOPTION_ENVIRONMENT_VERSION* is the portion of the upgrade configuration file name that lists upgrade process, upgrade environment, and the Siebel CRM release (version) from which you are upgrading. The file is located in *SIEBEL_ROOT\bin* (UNIX: *\$SIEBEL_ROOT/bin*).

Numbers like the following examples are used for the Siebel CRM release portion of the file name:

- 78
- 80
- 811 (SEA)

Table 24 lists an example of the file names for an upgrade from Siebel CRM version 8.0.

Table 24. Example of File Names for an Upgrade from Siebel CRM 8.0

Upgrade Mode	File Name
Development environment upgrep	master_upgrep_dev_80.ucf
Development environment upgphys	master_upgphys_dev_80.ucf
Additive Schema Changes	master_additive_gen.ucf
Prepare for Production	master_prepare_for_production_upgrade.ucf
Production environment upgrep and upgphys	master_upgrep_prod_80.ucf

- 3 To begin the upgrade, click OK (Windows) or press Enter (UNIX).

The Upgrade Wizard will notify you when the upgrade process is complete.

Stopping the Upgrade Wizard on Windows

Do not stop the Upgrade Wizard unless you are confident that an error has occurred, and the Upgrade Wizard or a utility it has called is hanging. Some SQL commands issued by the Upgrade Wizard or by its utilities can take considerable time to complete.

If you are not sure whether the Upgrade Wizard is hanging, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

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

To stop the Upgrade Wizard on Windows

- Do one of the following:
 - If the Upgrade Wizard has launched a separate command window in which a utility is running, then 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 might be a delay while the step completes in the RDBMS.

Stopping the Upgrade Wizard on UNIX

Do not stop the Upgrade Wizard unless you are confident that an error has occurred, and the Upgrade Wizard or a utility it has called is hanging. Some SQL commands issued by the Upgrade Wizard or by its utilities can take considerable time to complete.

If you are not sure whether the Upgrade Wizard is hanging, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

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

To stop the Upgrade Wizard on UNIX

- 1 If the Upgrade Wizard has started a utility in a child process, then 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 might be a delay while the RDBMS executes the issued SQL commands.

Related Topics

["About the Siebel Upgrade Wizard and Driver Files" on page 43](#)

Moving the Siebel Repository Files

Environments: Development and production test.

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.

Requirement: If you modified repository objects or schema definitions after completing the development upgrade, then you must regenerate the schema.ddl and custrep.dat files. See [“Regenerating the Siebel Repository Definition Files” on page 178.](#)

To move the repository files

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

Windows: *DBSRVR_ROOT\platform*

UNIX: *DBSRVR_ROOT/platform*

In these paths, *platform* is the database platform, for example, *DBSRVR_ROOT\DB2*.

- 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.

Running the Siebel Case Insensitivity Wizard

Environments: Development and production test.

Platforms: Windows, UNIX, IBM z/OS.

For Siebel CRM 8.1, query features are enhanced to provide indexes that directly support case and accent insensitive (CIAI) queries on eligible text columns. The Case Insensitivity Wizard configures specified columns for CIAI queries by defining CIAI columns and CIAI indexes in the repository. The wizard also sets the Default Insensitivity property for these columns to DB Case & Accent.

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

Overview of What the Case Insensitivity Wizard Does

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

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

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

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

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

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

CIAI Upgrade Issues for IBM DB2 Users

The Case-Insensitive and Accent-Insensitive search functionality requires changes on database schema levels (new columns, indexes, and triggers). When running the ddlimp utility on IBM DB2 databases, tables that have CIAI columns and triggers are not rebuilt. To resolve this issue, see 553429.1 (Article ID) on My Oracle Support.

Choosing the Correct Repository

If you are upgrading a development environment, then run the Case Insensitivity Wizard on the Siebel Repository. Later in the upgrade process, this repository will be renamed *Prior Customer Repository*.

If you have completed an upgrade of the development environment (upgrep, merge, upgphys), then run the Case Insensitivity Wizard on the Siebel Repository. You then must generate another schema.ddl file and use it to update your production test and production environments. Typically, you run the wizard after an upgrade is complete to revise the configuration of columns you have configured for case insensitive queries.

Running the Case Insensitivity Wizard Using an Input File

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

Windows: `SIEBEL_TOOLS_ROOT\objects`

These files list columns that are frequently used for queries and are provided as a recommendation. You can edit these files or create new input files as desired.

Requirements: See [“About the Siebel Case Insensitivity Wizard” on page 50](#) for information on how the Case Insensitivity Wizard works, eligibility criteria, and how to edit input files.

To run the Case Insensitivity Wizard using an input file

- 1 Review the input file and verify the following:
 - The syntax for all records is correct.
 - The tables and columns are eligible.
 - The specified configuration options are correct.
 - The configuration defaults are acceptable if any configuration options are omitted.
- 2 In Siebel Tools, open the repository.
- 3 Lock the tables listed in the input file.
- 4 From the Tools menu, choose Utilities, then Case Insensitivity.
The Case Insensitivity Wizard displays.
- 5 Select “Administer the columns listed in this file.”
- 6 Click Browse.
The `SIEBEL_TOOLS_ROOT\objects` directory displays, containing the default .csv input files.
- 7 Select the desired .csv file, and click Open.
- 8 In the wizard, click Next.

The Case Insensitivity Wizard validates the syntax of the file and validates the eligibility of all tables and columns. If the file contains errors or eligibility problems, then the wizard lists the records containing errors. If you continue, then the wizard skips records containing errors or eligibility problems.

If there are errors, then export the listing to a file, correct any errors in the input file, and restart the Case Insensitivity Wizard. Click Export to export the error listing to a text file.

If there are no errors, then click Next.

The Wizard displays the records in the input file.

9 Review the configuration settings, and verify that they are correct.

10 If you want to change any configuration settings, then click Export.

The Wizard exports the listing to a text file in input-file format.

11 Edit the text file, restart the Case Insensitivity Wizard, and specify the edited text file as the input file.

12 If the configuration settings are correct, then click Next.

The Wizard displays the changes it will make to repository tables and indexes.

13 If you want to save a record of the changes, then click Export.

The Wizard writes the changes to a text file.

14 Click Finish.

The Wizard configures the columns in the repository to support CIAI queries.

Running the Case Insensitivity Wizard by Selecting Columns

This procedure assumes the Object Explorer is in Types mode where objects are displayed hierarchically. An alternate way to perform the procedure is to display Object Explorer in Flat mode. Then choose the Column object and navigate to the desired columns.

Requirements: Review [“About the Siebel Case Insensitivity Wizard” on page 50](#) for information on how the Case Insensitivity Wizard works, eligibility criteria, and how to edit input files.

Running the Case Insensitivity Wizard by selecting columns

1 In Siebel Tools, open the repository.

2 In the Siebel Tools Object Explorer, select Table.

3 In the Tables list, lock the desired table, and highlight it.

4 In the Object Explorer, select Table, then Column.

5 In the Columns list, select the desired columns.

6 Right-click the highlighted columns, and choose Case Insensitivity from the pop-up menu.

The Case Insensitivity Wizard validates the eligibility of the selected columns. The Wizard lists any columns that have eligibility errors. If you continue, then the wizard skips columns containing errors.

Export the error listing to a text file for reference, correct any errors, and restart the Case Insensitivity Wizard. Click Export to export the error listing.

The Wizard displays the configuration settings it will use to configure the columns.

7 Review the configuration settings and verify they are correct.

8 If you want to change any configuration settings, then click Export.

The Wizard exports the listing to a text file in input-file format. Edit the text file, then run the Case Insensitivity Wizard and specify the text file as the input file.

9 If the configuration settings are correct, then click Next.

The Wizard displays the changes it will make to the repository tables and indexes.

10 If you want to save a record of the changes, then click Export.

The Wizard writes the changes to a text file.

11 Click Finish.

The Wizard configures the columns in the repository to support CIAI queries.

Configuring CIAI Support for Columns That Do Not Have Indexes Defined

If a column does not have any indexes defined, then the Case Insensitivity Wizard does not create new columns or indexes.

However, you can run the wizard in special mode that changes the column's Default Insensitivity property to DB Case & Accent. The Wizard performs these steps on all eligible columns in the repository. You cannot manually select columns.

Requirements: Review [“About the Siebel Case Insensitivity Wizard” on page 50](#) for information on how the Case Insensitivity Wizard works, eligibility criteria, and how to edit input files.

To configure columns without indexes

1 In Siebel Tools, open the repository.

2 From the Tools menu, choose Utilities, then Case Insensitivity.

The Case Insensitivity Wizard displays.

3 Select “Enable for all unindexed columns.”

4 Click Next.

The Wizard locates unindexed columns that meet CIAI eligibility criteria. The Wizard then displays a list of tables that must be locked.

5 Click Export to export the list of tables to a text file. Then exit the wizard.

6 Lock all the tables in the text file.

- 7 Start the Case Insensitivity Wizard again, and select “Enable for all unindexed columns.”
The Wizard locates unindexed columns and then displays a page listing how they will be configured.
- 8 Verify that for all the columns that the method is set to Database, and Index Strategy is set to None.
When the index strategy is set to None, the wizard does not create a CIAI column or indexes.
When you click Next, the wizard displays a page listing the repository changes it will make.
- 9 Verify that for all columns, Default Insensitivity will be changed to DB Case & Accent. Click Finish.
The Wizard makes the changes to the repository.

Related Topics

[“About the Siebel Case Insensitivity Wizard” on page 50](#)

[“Regenerating the Siebel Repository Definition Files” on page 178](#)

Applying Siebel Additive Schema Changes

Environments: Production test, production.

Platforms: Windows, UNIX, IBM z/OS.

As of Siebel CRM 8.0, you can run Database Configuration Utilities in Apply Additive Schema Changes mode. This generates a `schema.additive.sql` script that makes schema changes to the Siebel database schema. You do not have to take the database offline to apply this script.

Applying this script to your production database reduces the number of upgrade steps that must be performed while the production database is offline. This reduces production database downtime during the upgrade.

The recommended method for using this feature is to first test it in the production test environment as part of upgrade tuning. After verifying that `schema.additive.sql` does not adversely affect the operation of the application, you then run the script against the production database.

Requirements for Applying Siebel Additive Schema Changes

The following are requirements for applying Siebel additive schema changes:

- You must have run the Database Configuration Utilities in Apply Additive Schema Changes mode to generate the `schema.additive.sql` script.
- You must have reviewed the Apply Additive Schema Changes log file and verified that `schema.additive.sql` was generated without errors.
- If you are testing the script in the production test environment, then the schema of the Siebel database must be the same as the schema of the production environment Siebel database.

Applying Siebel Additive Schema Changes

Use the following procedure to apply Siebel additive schema changes.

To apply additive schema changes

- 1 Navigate to the directory where `schema.additive.sql` is located.
- 2 Review the script and verify that it does not make unacceptable changes to the schema.
- 3 Production environment only. Compare the script generated in the production environment with the one generated in the production test environment. If the scripts are not identical, then determine why. If you cannot determine that the differences are benign, then do not apply additive schema changes in the production environment.
- 4 Using any SQL editor, run the script against the Siebel database.

You can run `schema.additive.sql` multiple times. For example, you can run the script, revise it, and run it again.

Related Topics

[“About Siebel Additive Schema Changes Mode” on page 46](#)

Regenerating SQL Files for a Siebel Upgrade

Environments: Development and production test.

If you enter incorrect information in the Database Configuration Utilities, then the SQL files the utilities generate will be incorrect and must not be used. You must run the utilities again, enter the correct information, and regenerate the SQL files.

For example, you are upgrading a non-Unicode database. You run the Database 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 Configuration Utilities generate SQL files, all previously generated SQL files are overwritten and newly created.

To regenerate SQL files with the Database Configuration Utilities

- 1 Verify that the SQL files must be regenerated.
Typically this is caused by entering incorrect information in the Database Configuration Utilities. If you are unsure if the files must be regenerated, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle’s Application Expert Services.
- 2 Rerun the Database Configuration Utilities.

Related Topics

[“About the Siebel Database Configuration Utilities” on page 37](#)

Identifying and Dropping Obsolete Indexes for a Siebel Upgrade

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

Use this topic to identify indexes that might 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 Configuration Utilities in Prepare for Production mode, they do the following to identify obsolete indexes:

- Compares the repository schema definition in the development environment against the Siebel database physical schema definition in the production test environment.
- If an index is present in the Siebel database physical schema definition but not in the repository logical schema definition, then the utility creates an SQL drop statement and adds it to that index. The utility places this SQL statement in a file called `gen_obs_idx.sql`.

You must manually review the `gen_obs_idx.sql` file. If it contains indexes you want to remove (those containing a drop statement not followed by a create statement), then you must copy the corresponding SQL statements to another SQL file called `obs_idx.sql`. This file is executed by the Database Configuration Utilities in `upgrep` mode.

When the Upgrade Wizard is then run, all indexes, including obsolete indexes, are maintained during table rebuilds and data migration. The obsolete indexes file is executed during the Create Siebel Indexes step.

Requirements: You must have run the Database 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`

In these paths, *platform* is the database type, for example DB2.

- 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 remove an index, then copy the corresponding SQL statement(s) to the following file:

Windows: *DBSRVR_ROOT\platform\obs_idx.sql*

UNIX: *DBSRVR_ROOT/platform/obs_idx.sql*

In these paths, *platform* is the database type, for example DB2.

This file will be executed when you run the Database Configuration Wizard in upgrep mode.

Preparing for a Nondevelopment Environment Siebel Upgrade

Environments: Production test, production.

Platforms: Windows, UNIX, IBM z/OS.

If your installation does not include a development environment, then 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 Siebel upgrade in a nondevelopment environment

- 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\DB2*.
- 4 In the *platform* directory, copy the *ddl.ctf* file and paste the copy into the same directory.
- 5 Rename the copy *schema.ddl*.
- 6 In the production test environment create a new database, separate from the Siebel database. Install the Siebel database from the new release in the new database. Do not migrate any data to the new database. This database is called the reference database.
- 7 Define an ODBC for the reference database.

11 Reviewing the Siebel Upgrade Log Files

This chapter provides information on accessing, reviewing and archiving Siebel upgrade log files. This chapter includes the following topics:

- [Summarizing Siebel Log Files Using the Logparse Utility on page 159](#)
- [Reviewing Siebel Upgrade Log Files for Errors on page 161](#)
- [Manually Archiving Siebel Upgrade Log Files on page 165](#)

Summarizing Siebel Log Files Using the Logparse Utility

Environments: Development, production test, production.

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

Use the Logparse utility to analyze and summarize the log files created when you run the Upgrade Wizard.

The Logparse utility writes its findings to a summary file:

Windows: `SIEBEL_ROOT\log\process\summary.html`

UNIX: `$SIEBEL_ROOT/log/process/summary.html`

In these paths, *process* is the upgrade process you want to review, for example, `upgprep_dev_8x`.

The Logparse utility also writes a `summary.txt` file that contains the same information as `summary.html`. Use the `summary.txt` file if you do not have a browser.

After writing the output files, the Logparse utility opens `summary.html` automatically.

Command-Line Syntax for the Logparse Utility

Table 25 lists the command-line syntax for the Logparse utility.

Table 25. Command-Line Syntax for the Logparse Utility

Flag	Parameter	Description	Comment
/s	<i>install_dir</i>	Full path to the parent directory of <i>SIEBEL_ROOT</i> ($\$SIEBEL_ROOT$). For example, if <i>SIEBEL_ROOT</i> is C: \sba8x\si ebsrvr, then <i>install_dir</i> is C: \sba8x. Enclose the path in quotes.	Required
/g	<i>LANGUAGE_CODE</i>	The language in which the Upgrade Wizard ran. This is called the resource language, and typically is the language in which you ran the Siebel Installation Wizard. For example, if the resource language is English, then the language code is enu.	Required
/r	<i>PROCESS</i>	Name of the schema process for which you want to generate a summary file, for example, upgrep_dev_8x. To obtain the schema process name, look in si ebsrvr\l og. The subdirectory names in l og are the schema process names.	Required.
/l	<i>LOGPARSE_FILENAME</i>	Name of the Logparse log file.	Default value is l ogparse. l og
/n	<i>MAX_NUMBER_SQL</i>	Maximum number of longest-running SQL statements to display in the summary.	Default value is 10
/t	<i>THRESHOLD_TIME</i>	Threshold time for longest running SQL statement, in the format <i>hh:mm:ss</i> .	Default value is 00:10:00 (By default, SQL statements that run longer than 10 minutes are displayed)
/e	<i>MAX_NUMBER_ERRORS</i>	Maximum number of errors to display in the summary.	Default is 8

Running the Logparse Utility

Use the following procedure to run the Logparse utility.

To run the Logparse utility

- 1 Navigate to the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Run the following command. Use the syntax in [Table 25 on page 160](#):

```
logparse /s install_dir /g LANGUAGE_CODE /r PROCESS /l LOGPARSE_FILENAME
/n MAX_NUMBER_SQL /t THRESHOLD_TIME /e MAX_NUMBER_ERRORS
```

Windows example (Upgrade Wizard language is English):

```
logparse /s C:\sea8xx /g enu /r upgrep_dev_8xx /l logparse.log /n 10 /t 00:00:10
/e 10
```

- 3 Review the Logparse log and verify that Logparse ran without errors.

Windows: `SIEBEL_ROOT\bin\logparse.log`

UNIX: `$SIEBEL_ROOT/bin/logparse.log`

Related Topics

[“About the Siebel Database Upgrade Log Files” on page 48](#)

[“Reviewing Siebel Upgrade Log Files for Errors” on page 161](#)

Reviewing Siebel Upgrade Log Files for Errors

Environments: Development, production test, production.

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

Using the Logparse utility to review the log files created when you run Upgrade Wizard is the recommended way to verify that the upgrade process completed correctly and to identify errors that must be resolved.

The Logparse utility writes its findings to a summary file:

Windows: `SIEBEL_ROOT\log\process\summary.html`

UNIX: `$SIEBEL_ROOT/log/process/summary.html`

In these paths, *process* is the upgrade process you want to review, for example `upgrep_dev_8x`.

Logparse also writes a `summary.txt` file that contains the same information as `summary.html`. Use the `summary.txt` file if you do not have a browser.

The procedures in this topic describe how to review the `summary.html` file.

Verifying the Upgrade Environment

Before reviewing the upgrade steps for errors, verify that the upgrade environment is set up correctly.

To verify the upgrade environment

- 1 If `summary.html` is not already open in your browser, then navigate to `si_ebsrvr\log`, and locate the subdirectory for the upgrade process you want to review.
- 2 In the desired upgrade process directory, click `summary.html` to open the file in a browser.
- 3 On the `summary.html` home page, Upgrade/Install Log File Navigation Page, locate the Parameter column in the table, and click the Parameters link.

The Parameters page displays.

- 4 Review the parameters, and verify that they are correct.

License key values are encrypted. If you find errors in the parameters, then correct them, and run the upgrade process again as needed.

Determining Whether the Upgrade Process Completed Successfully

The upgrade processes consist of a series of steps. Each step must complete successfully.

If the Upgrade Wizard cannot complete a step, then it marks the step incomplete in the state log (`si_ebsrvr\log\Process\state\state.log`) and exits. You must correct the error, and run the Upgrade Wizard again. When you rerun the Upgrade Wizard, it refers to the state log and resumes at the incomplete step that contained the error.

Use the following process to identify errors:

- 1 Resolve errors for steps with a Status of Incomplete.
- 2 Review all the steps with a Status of Complete. If any contain unacceptable errors, then resolve these errors.
- 3 Restart the Upgrade Wizard, or if necessary, restore the database and rerun the upgrade process.

If you have any questions regarding how to resolve errors, then contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

To determine whether the upgrade process completed successfully

- 1 If `summary.html` is not already open in your browser, then navigate to `si_ebsrvr\log`, and locate the subdirectory for the upgrade process you want to review.
- 2 In the desired upgrade process directory, click `summary.html` to open the file in a browser.

- 3 On the summary.html home page, Upgrade/Install Log File Navigation Page, locate the Step/Error column in the table and click the Steps/Errors link.

The Steps/Errors Summary page displays.

- 4 Locate the Status column, and verify that all the steps have a status of Complete.

If the Status of all the steps is Complete, then the upgrade process completed successfully.

If the Status of any step is Incomplete, then the upgrade process did not complete successfully. The Upgrade Wizard has exited before completing all the steps in the upgrade process. You must identify the error, and correct it.

CAUTION: In some cases, the Upgrade Wizard can complete a step even though the step contains unacceptable errors. You must verify that all steps do not contain unacceptable errors.

Determining Whether Steps Contain Unacceptable Errors

There are two types of unacceptable errors:

- The Upgrade Wizard could not complete a step due to an error and exited. You can locate these errors by checking the Status field on the Steps/Errors Summary page. If the Status of a step is Incomplete, then the step contains an unacceptable error.
- The Upgrade Wizard encountered an unacceptable error but was able to complete the step. You can locate these errors by examining the detailed information for each step.

To determine whether steps contain unacceptable errors

- 1 If summary.html is not already open in your browser, then navigate to si_ebsrvr\log, and locate the subdirectory for the upgrade process you want to review.
- 2 In the desired upgrade process directory, click summary.html to open the file in a browser.
- 3 On the summary.html home page, Upgrade/Install Log File Navigation Page, locate the Step/Error column in the table, and click the Steps/Errors link

The Steps/Errors Summary page displays.

- 4 In the Name column, click the name link.

The detailed information for the step displays. An alternate method is to use the slide bar to browse down through the detailed information listing for all the steps.

- 5 IBM DB2 and Oracle Database: In the detailed information for each step, check the Errors item.

If the value for Errors is None, then the step has no unacceptable errors.

If table lists one or more errors, then these errors are unacceptable and must be corrected. Click the links to the log files to obtain more information about the errors.

- 6 Microsoft SQL Server: In the detailed information for each step, check the Errors item.

If the value for Errors is None, then the step has no errors.

If there is a table or a link to a table listing errors, then do the following:

- a Open the errors.xls spreadsheet:

Windows: `DBSRVR_ROOT\DATABASE_PLATFORM\errors.xls`

UNIX: `DBSRVR_ROOT/DATABASE_PLATFORM/errors.xls`

If an errors.rtf (Windows) or errors.txt (UNIX) file is present, then you can use either of these files. They contain the same content as errors.xls. The errors file lists benign errors that do not need to be corrected.

- b Compare the errors listed in the Logparse summary.html file to the list of acceptable errors documented in the error file.

The log files generated by Upgrade Wizard appear in the errors file as upgwi z1.log, upgwi z2.log, incrementing for additional log files.

- c If you find the error in the errors file, then the error is acceptable and no action is required.

If you find an error that is not listed in the errors file, then the error is unacceptable. You must correct the condition that caused the error.

- 7 To help resolve unacceptable errors, navigate to 477324.1 (Article ID) on My Oracle Support. Common errors and how to resolve them are listed in this document. If the error is not listed or you cannot resolve it, then contact Oracle Global Customer Support. Do not proceed with the upgrade.

Reviewing Upgrade Performance

The Logparse summary file provides the input that Upgrade Tuner uses to analyze the performance of the upgrade scripts. For example, the summary file provides Upgrade Tuner with information about SQL commands that returned 0 rows.

If you plan to use the Upgrade Tuner to tune your production upgrade, then familiarize yourself with the performance information in the summary file.

To review upgrade performance in the summary file

- 1 If summary.html is not already open in your browser, then navigate to `si_ebsrvr\log`, and locate the subdirectory for the upgrade process you want to review.
- 2 In the desired upgrade process directory, click `summary.html` to open the file in a browser.
- 3 On the `summary.html` home page, Upgrade/Install Log File Navigation Page, locate the following columns:
 - Performance Information for SQL
 - Performance Information for DDL

- 4 Click the links in these columns to display performance information about the upgrade process.
The Performance Information for SQL page displays the longest running queries and queries that returned 0 rows for SQL scripts.
The Performance Information for DDL page displays the tables created, altered, and deleted by the ddlimp utility and by SQL scripts.
In both pages, the Net Cost column lists the time required to perform each operation.

Related Topics

[“About the Siebel Database Upgrade Log Files” on page 48](#)

[“Summarizing Siebel Log Files Using the Logparse Utility” on page 159](#)

[“Manually Archiving Siebel Upgrade Log Files” on page 165](#)

Manually Archiving Siebel Upgrade Log Files

Environments: Development, production test, production.

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

After a successful installation and upgrade, you must manually save and archive the log files located in the *SIEBEL_ROOT\log\PROCESS* (Windows) directory.

By default, only nine upgrade log files are retained for subsequent retries of the Upgrade Wizard. After nine log files have been created, when the Upgrade Wizard is rerun, it overwrites log files beginning with the earliest one created and recycles the rest as necessary. (This does not apply to the state.log file.)

The number of log files retained can be increased by resetting the *si_ebel_log_archive* environment variable to 20. For example, to retain twenty (20) log files.

12 Performing the Siebel Repository Merge

This chapter provides guidelines for performing a Siebel Repository merge. This chapter includes the following topics:

- [Preparing for the Repository Merge on page 167](#)
- [Configuring Siebel Repository Objects to Inherit Upgrade Behavior on page 168](#)
- [Performing a Siebel Repository Merge on page 169](#)
- [Reviewing the Siebel Repository Merge Log Files on page 171](#)
- [Generating Siebel Enterprise Integration Manager Temporary Columns on page 174](#)
- [Reviewing Siebel Repository Object Property Conflicts on page 174](#)
- [Regenerating the Siebel Repository Definition Files on page 178](#)
- [Deleting Unneeded Siebel Repository Files on page 180](#)
- [Migrating Siebel Repository Objects to the Standard User Interface on page 180](#)
- [Running the Siebel Postmerge Utilities on page 185](#)

Preparing for the Repository Merge

Environments: Development environment only.

The process of merging repositories to create a final customized repository used in the upgrade is time-intensive and resource-intensive. As a result, a merge might fail due to environmental factors, such as space constraints. When this happens, the merge process continues, even if there is a fatal database error, and the errors might go undetected.

If the merge fails, then you must restore the database environment to its premerge state and run the repository merge again. Additionally, it is recommended, as another precaution, that you export the New Customer Repository to preserve any existing workflows.

Recovering from a Failed Merge

It is recommended that you perform one of the following two tasks to preserve the premerge environment in the event of a failed merge:

- **Perform full database backup prior to the merge.** Back up the entire database prior to the merge. If the merge fails, then you can restore the database to its premerge state, and rerun the merge operation.

- **Export the New Customer Repository prior to the merge.** Export the New Customer Repository prior to the merge to create a backup copy. If the merge fails, then delete the failed repository, then import the backed up copy of the New Customer Repository. See *Using Siebel Tools* for information on exporting and importing repositories using the Database Configuration Wizard.

NOTE: If you export the New Customer Repository, then you must truncate the following merge log tables: S_MERGE_LOG, S_MERGE_LOG_OBJ, and S_MERGE_LOG_ATTR.

Configuring Siebel Repository Objects to Inherit Upgrade Behavior

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.

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.

Specifying an Upgrade Ancestor

Use the following procedure 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.

Viewing the Descendants or Copies of an Object

Use the following procedure to view the descendants or copies of an object.

To view the descendants or copies of an object

- 1 Right-click the object.

- 2 Select View descendants from the picklist.
A dialog box appears and lists the descendants.

Related Topics

[“About the Siebel Repository Merge” on page 51](#)

[“About Inheriting Upgrade Behavior in a Siebel Upgrade” on page 57](#)

Performing a Siebel Repository Merge

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

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.

Requirement: 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, then additional memory might 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:

```
SIEBEL_TOOLS_ROOT\bin\lang_code
```

In this path, *lang_code* is the language, for example enu.

- 4 Locate the SIEBEL section, and verify that the parameters are set as shown in the following example.

Parameter	Value
EnableToolsConstraint	FALSE
SymStrPrefix	X_

- 5 Save the file and close it.
- 6 Navigate to Control Panel, System, Advanced, Performance Settings, and then Visual Effects.
- 7 Click Adjust for best performance.

- 8 Start Siebel Tools in the new Siebel release, using the following command:

```
SIEBEL_TOOLS_ROOT\bin\si ebdev /edi tseeddata
```

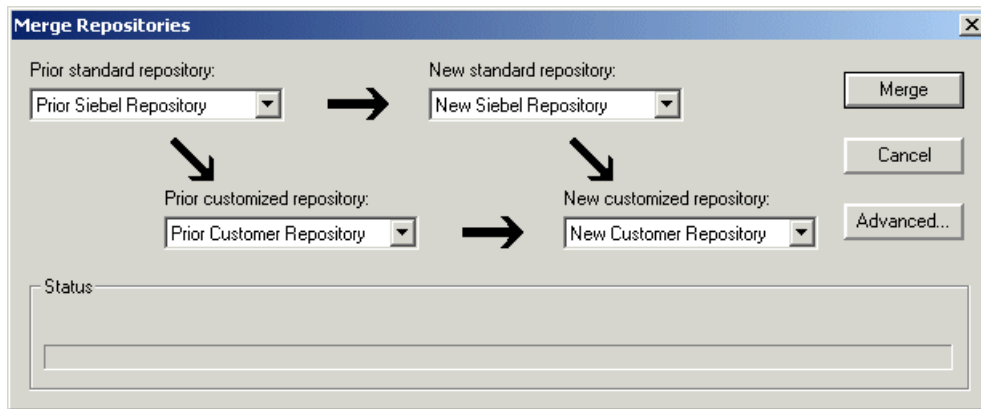
NOTE: The merge0.txt file will be generated in the *SIEBEL_TOOLS_ROOT\bin* directory.

- 9 From the Tools menu, choose View, Options, and then 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 From the File menu choose the Open Repository command to open the Prior Customer Repository.
- 12 From the Tools menu, choose Upgrade, then the Upgrade Application menu item.

The Merge Repositories dialog box appears.



The Merge Repositories dialog box provides four options:

- **Merge.** This button merges the repositories you specify to produce a New Customer Repository.
 - **Cancel.** This button cancels the repository merge and exits the Merge Repositories dialog box.
 - **Advanced.** This button opens the Merge Options dialog box.
- 13 In the Merge Repositories dialog box, choose the repositories listed in the following table.

Drop-Down List Item	Value to Choose
Prior Standard Repository	Prior x.x Siebel Repository, as appropriate for the version from which you are upgrading
Prior Customized Repository	Prior Customer Repository

Drop-Down List Item	Value to Choose
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, then set the number of errors to a large value. This will help prevent the Repository merge from aborting due to benign errors.

- **Incorporate Custom Layout.** Activate this option to help preserve field and button placement on prior custom or modified forms, views, and screens. Select a prior release and style for label placement.

After you have made your selections, click OK.

The Upgrade Check List dialog box appears.

16 In the Upgrade Check List dialog box, verify that each requirement has been met. When they have been met, select all the check boxes and click Continue.

CAUTION: The upgrade might fail if all the items in the checklist are not completed.

After the merge completes, a dialog box appears, requesting that you make a backup of the New Customer Repository.

Related Topics

[“About the Siebel Repository Merge” on page 51](#)

[Appendix C, “Siebel Incorporate Custom Layout Upgrade Option”](#)

Reviewing the Siebel Repository Merge Log Files

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

To determine whether the repository merge was successful, review the merge log files. The merge is successful if it completes without unacceptable errors:

- **Acceptable errors.** If an ancestor object is specified in an object definition, and the ancestor object is not present in the New Siebel Repository, this causes a merge error. This is an acceptable error and can be ignored.

Here is an example of an acceptable error in the merge 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.

For more detailed information, go to 477269.1 (Article ID) on My Oracle Support, This document was previously published as Siebel Troubleshooting Steps 19.

Merge errors are displayed in the Upgrade Applications Objects List view in Siebel Tools. Additional details on merge errors are located in the repository merge log:

```
SIEBEL_TOOLS_ROOT\bin\merge0.txt
```

Each time you run the merge process, the name of the merge0.txt file is incremented, for example to merge1.txt.

If your repository merge process terminates and is flagged as *Incomplete*, then navigate to the Screens menu in Siebel Tools, and choose the Application Upgrader menu item. The most common reasons for its failure are:

- The number of errors (!!ERROR) exceeds the number that was predefined in Siebel Tools when the merge was started.
- The merge process has been terminated due to a local issue on the Siebel Tools workstation, such as a scheduled reboot.
- RDBMS errors caused the process to stop.
- Memory allotment issues on the workstation on which Siebel Tools is installed.
- Network failure.

If the repository merge terminates and is flagged as Incomplete, then the merge must be restarted, as discussed in [“Performing a Siebel Repository Merge” on page 169](#).

To determine whether the repository merge was successful

- 1 In Siebel Tools, from the Screens menu, choose Application Upgrader, and then 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, then no further action is required. The merge was successful.

If the Status column indicates Completed with Errors, then you must review the errors to determine whether 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:

SIEBEL_TOOLS_ROOT\bin

If there are multiple files, then open the one with the highest number in the file name, for example merge1.txt.

8 To locate merge errors in the file, search for the !!ERROR string.

Additionally, informational messages are marked as !!INFO.

9 Use the objects displayed in the Object Differences list and the errors displayed in the log file to analyze the errors:

- If all the errors are acceptable, then the merge is considered successful.

It is advisable, however, to consider the number of acceptable errors when determining whether to re-run the merge operation.

- If the log contains unacceptable errors, then the merge has failed.

10 If the merge contains unacceptable errors, then go to 477269.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Troubleshooting Steps 19.

This document explains the meaning of many of the error messages that can appear in the log file. Use this document to correct the errors. If you cannot resolve all the errors, then contact Oracle Global Customer Support.

11 Open the workflow merge log file:

SIEBEL_TOOLS_ROOT\bin\merge0_ver.txt

If there are multiple files, then open the one with the highest number in the file name, for example merge1_ver.txt. This log file is created by the workflow premerge and postmerge steps.

12 Review the log file. If the file contains errors, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Related Topics

["About the Siebel Repository Merge" on page 51](#)

["Upgrade Planning for Siebel Workflow Designer" on page 76](#)

Generating Siebel Enterprise Integration Manager Temporary Columns

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

The repository merge process does not preserve Siebel Enterprise Integration Manager processing columns for custom mappings. You must generate the missing custom Siebel Enterprise Integration Manager processing columns again.

To generate Siebel Enterprise Integration Manager temporary columns

- 1 In Siebel Tools, from the File menu, choose Open Repository, and then New Customer Repository.
- 2 From the Tools menu, choose, Upgrade, and then Generate EIM Processing Columns.
A dialog box displays.
- 3 Click OK to generate Siebel Enterprise Integration Manager processing columns for custom mappings.
- 4 In the Object Explorer window, choose New Customer Repository and verify that the Comment field shows UpgEi mCol .

This indicates that the Siebel Enterprise Integration Manager temporary columns were created successfully.

Related Topics

[“About the Siebel Repository Merge” on page 51](#)

Reviewing Siebel Repository Object Property Conflicts

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

You can change how object property conflicts were resolved during the repository merge.

NOTE: Only critical conflicts are displayed when you are resolving conflicts following the repository merge.

How Object Property Conflicts Occur

The repository merge compares repository object properties in the Prior Siebel Repository, Prior Customer Repository, and New Siebel Repository. When the value of an object property is different in all three repositories, an object property conflict exists.

This occurs when you have changed the value of an object property in the Prior Customer Repository, and the value of this property has also changed in the new release (New Siebel Repository).

An object property conflict does not occur if you changed the value of an object property in the Prior Customer Repository, and the object property value did not change in the new release. When this happens, the merge process transfers the changed value to the New Customer Repository.

The merge process resolves object property conflicts by referring to the setting of the object's Standard Win property. For about 90% of repository objects, the merge process resolves conflicts by using the object property value in the New Siebel Repository.

Do not change the setting of the Standard Win property.

Application Upgrade Attribute List View

You can review and change how object property conflicts were resolved using the Application Upgrade Attribute List view in Siebel Tools. The Attribute Differences List in the view includes the following columns:

- **Object Name.** The name of the object.
- **Attribute.** The object property name.
- **Conflict.** The merge process puts a check mark in this field if there was an object property conflict during the merge.
- **Resolution.** Displays which property value the merge process used to resolve the conflict:
 - **Standard Value.** The property value in the New Siebel Repository was used. This value is displayed in the New Standard column.
 - **Custom Value.** The property value in the Prior Customer Repository was used. This value is displayed in the Prior Customized column.
- **Override.** Put a check mark in this column to change how the conflict is resolved. Overriding the resolution changes the property value in the merged repository. If the resolution was the Standard Value, then the attribute value is changed 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.

Requirement: The repository merge must have been successful. See [“Reviewing the Siebel Repository Merge Log Files”](#) on page 171.

Reviewing Object Property Conflicts

Use the following procedure to review object property conflicts.

To review object property conflicts

- 1 In Siebel Tools, from the Screens menu, choose Application Upgrader, and then 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.

Reviewing Object Property Conflicts at the Attribute Level

Use the following procedure to review object property conflicts at the attribute level.

To review conflicts at the attribute level

- 1 In Siebel Tools, from the Screens menu, choose Application Upgrader, and then Application Upgrade Attribute List.

By default, you can see only the critical conflicts in the Application Upgrades Attribute differences screen. You must review the critical conflicts. To view informational conflicts, right-click and select Show Non-Critical Conflicts. Noncritical conflicts include conflicts of attributes, such as Sequence number, Comments, Layout, and so on.

Application Upgrades - Attribute...

User	Operation	Status	Prior Standard Repository	Prior Customized Repository	New Standard Repository	New Customized Repository
SADMIN	Merge(NonICL)	Complete - 99 errs	Prior Siebel Repository	Prior Customer Repository	New Siebel Repository	New Customer Repository

Attribute Differences

Conflict Resolution Type	Object Name	Top Parent Name	Top Parent Type	Object
Critical	Search Specification	Product Compatibility - Default: 1	Workflow Process	WF Step I/O Argument 'Search Specification' of W
Critical	No	ISS Promotion Verify SubProcess: 1	Workflow Process	WF Step Branch 'No' of WF Step 'Violate Integrity
Critical	NewConn:42-4SKOQ	Product Compatibility - Default: 1	Workflow Process	WF Step Branch 'NewConn:42-4SKOQ' of WF Ste
Critical	NewConn:1-3MBL9	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3MBL9' of WF
Critical	NewConn:1-3MBL9	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3MBL9' of WF
Critical	NewConn:04-G6CEQ	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:04-G6CEQ' of v
	NewConn:04-G6CEQ	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:04-G6CEQ' of v
	NewConn:1-3J92M	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3J92M' of WF
	NewConn:1-3J92M	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3J92M' of WF
	NewConn:1-3JA19	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3JA19' of WF
	NewConn:1-3JA19	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3JA19' of WF
	NewConn:1-3MBLD	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3MBLD' of WI
	NewConn:1-3MBLD	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3MBLD' of WI
	NewConn:42-4ZY1P	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:42-4ZY1P' of W
	NewConn:42-4ZY1P	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:42-4ZY1P' of W
	NewConn:04-ECIY2	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:04-ECIY2' of W
	NewConn:04-ECIY2	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:04-ECIY2' of W
Critical	NewConn:1-3MBLL	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3MBLL' of WF

- 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 in 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 that changes the value of the object property in the merged repository. Avoid overriding conflicts for the following object properties. Review the following properties in the upgraded application before changing them: Left, right, top, height, and width.

Related Topics

["About the Siebel Repository Merge" on page 51](#)

Regenerating the Siebel Repository Definition Files

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

If you have modified repository objects after the development environment upgrade (upgphys) and before upgrading the production test environment, then 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, then you must regenerate the files.

Regenerating the schema.ddl File

Use this procedure to regenerate the schema.ddl file.

To regenerate the schema.ddl file

- 1 On the Siebel Server where the Siebel Database Server files are installed, navigate to the following location:

Windows: *SIEBEL_ROOT*\bin

UNIX: *\$SIEBEL_ROOT*/bin

- 2 Run the following command:

```
ddl di ct /u DatabaseOwner /p Password /c "ODBCDataSource" /d TableOwner /f
DBSRVR_ROOT\DatabasePlatform\schema.ddl /e y /a y /l Siebel LogDir\sch_di ct. log
/n "Siebel Repository" /t dcir
```

where:

- *DatabaseOwner* is the Siebel database administrator account name.
- *Password* is the Siebel database administrator account password.
- *ODBCDataSource* is the ODBC name for connecting to the database. Enclose the name in quotes.
- *TableOwner* is the Siebel table owner name.
- *DBSRVR_ROOT* is the absolute path to the Siebel Database Server installation directory.
- *DatabasePlatform* is the Siebel Database Server directory name for the database, for example Oracle. The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.

- *SiebelLogdir* is the path to the directory where you want the output log placed (log output directory). The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
- 3 After the command completes, review the output log files for errors. If the log indicates there are errors, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Regenerating the custrep.dat File

Use this procedure to regenerate the custrep.dat file.

To regenerate the custrep.dat file

- 1 On the Siebel Server where the Siebel Database Server files are installed, navigate to the following location:

Windows: *SIEBEL_ROOT*\bin

UNIX: *\$SIEBEL_ROOT*/bin

- 2 Run the following command:

```
repiexp /a e /u DatabaseOwner /p Password /c "ODBCDataSource" /d TableOwner  
/r "Siebel Repository" /f DBSRVR_ROOT\DatabasePlatform\custrep.dat  
/l SiebelLogDir\exprep.log
```

where:

- *DatabaseOwner* is the Siebel database administrator account name.
 - *Password* is the Siebel database administrator account password.
 - *ODBCDataSource* is the ODBC name for connecting to the database. Enclose the name in quotes.
 - *TableOwner* is the Siebel table owner name.
 - *DBSRVR_ROOT* is the absolute path to the Siebel Database Server installation directory. The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
 - *DatabasePlatform* is the Siebel Database Server directory name for the database, for example Oracle.
 - *SiebelLogdir* is the path to the directory where you want the output log placed (log output directory). The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
- 3 After the command completes, review the output log files for errors. If the log indicates there are errors, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Related Topics

[“About the Siebel Database Configuration Utilities” on page 37](#)

Deleting Unneeded Siebel Repository Files

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

When you are confident that the repository has been upgraded successfully, export the New Siebel Repository and Prior Customer Repository for safekeeping. You can also delete the following repositories:

- Prior Standard Repository
- New Standard Repository

Related Topics

[“About the Siebel Repository Merge” on page 51](#)

Migrating Siebel Repository Objects to the Standard User Interface

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

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

If you selected the Incorporate Custom Layout (ICL) merge option for your prior upgrade, then user interface objects in the New Customer Repository were deactivated and replaced by user interface objects from the Prior Customer Repository. Before upgrading to the next release, you must reactivate the standard user interface objects. You do this after the development upprep and before the repository merge.

The migration returns the user interface to the state it would have been after a regular merge instead of an ICL merge. The following list refers to the Prior Customer Repository from your previous upgrade.

- Customer-Created screens, views and applets are retained.
- User interface controls you added in the Prior Customer Repository are retained, but control placements might need to be reconfigured.
- User interface 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 user interface. Customizations are preserved, but you might need to revise object layouts to conform to the user interface standard in the new release.
- Standard objects in the Prior Customer Repository that were not deleted, or modified afterward are migrated to the standard user interface. No reconfiguration is required.
- Standard objects in the Prior Customer Repository that were modified afterward are not migrated to the standard user interface. Customizations are preserved, but you might need to revise object layouts to conform to the user interface in the new release.

Migrating to the standard user interface uses defined logic to select objects and then modifies these objects as follows:

- **ICL objects.** These are the screens, views, applets and child objects that received ICL handling in your previous upgrade. The ICL merge deactivated standard user interface objects in the New Customer Repository and replaced them with ICL objects from the Prior Customer Repository. Migration to the standard user interface deactivates and deletes ICL objects and activates their counterpart standard user interface objects.
- **Standard user interface objects.** These are the user interface objects that were deactivated by the ICL merge step and have `-UPG` appended to their names. They contain customizations from the previous repository merge, such as control deletion and addition. They do not contain certain user interface layout customizations such as control placement. Migration activates these objects. Standard user interface objects replace the ICL objects for the release you are upgrading from. The layouts of some of the standard user interface objects will need to be reconfigured.

Using Lag Time to Identify Postupgrade Customizations

ICL objects you have customized after the previous upgrade are not affected by the migration to the standard user interface. 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 user interface objects. If the modification times differ by more than a specified lag time, then 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 user interface 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 user interface object.

You then use Siebel Tools to migrate the repository to the standard user interface 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 user interface 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 user interface objects and ICL objects in the repository. The process then determines whether to modify the standard user interface object or the corresponding ICL object:

- Deleting the ICL object and activating the corresponding standard user interface object migrates the user interface object to standard.
- Deleting a standard user interface object and retaining the corresponding ICL object preserves the customized ICL object.

[Table 26 on page 183](#) 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 user interface 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 Table 26, Siebel Repository refers to your current Siebel Repository.

Table 26. 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 user interface 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 user interface 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	Not applicable	<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 user interface object and children of this object.
No	Not applicable	Not applicable	<ul style="list-style-type: none"> ■ Status of object is not changed from Inactive to Active. ■ Removes the read-only restriction. You can delete or modify the object.

How Logging Is Done

The migration process writes the changes to the repository in the following log file:

SIEBEL_ROOT\log\icl migration.log

The beginning of the log lists top-level objects that were affected by the ICL feature. For each object, the log then iteratively lists the operations performed on all child objects.

Specifying a Lag Time

The default lag time is 72 hours. You can revise this by editing the Siebel Tools .cfg file. The minimum is 24 hours, and the maximum is 120 hours.

Requirement: You must have installed the version of Siebel Tools that shipped in the new release.

To specify a lag time

- 1 Navigate to the `SIEBEL_TOOLS_ROOT\bin\lang_code` directory, where `lang_code` 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 User Interface

A menu option in Siebel Tools migrates ICL objects to the standard user interface. Upgrading the user interface requires a minimum of three hours to complete.

Requirements:

- You must have installed the Siebel Tools version that shipped in the new release.
- Set the lag time as desired. The default is 72 hours.
- Verify that you have backed up the upgraded database.

To migrate to the standard user interface

- 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 box, make the following selections and click Continue:
 - **Prior Customer Repository:** Select Prior Customer Repository. This is the Siebel Repository in your current release, which you have renamed in preparation for the upgrade.
 - **Prior Standard Repository:** Select Prior V7.X Siebel Repository. This is the Prior Standard Repository for your currently installed release. This repository was loaded when you performed the database upgrade for the new release.

- 4 In the ICL Migration Warning dialog box, 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, then you must restore the saved Prior Customer Repository from backup and begin the migration again.

Related Topics

[“About the Siebel Repository Merge” on page 51](#)

[Appendix C, “Siebel Incorporate Custom Layout Upgrade Option”](#)

Running the Siebel Postmerge Utilities

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

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 user interface.
- Revise multi-value group (MVG) applets so that they are shuttle-enabled.
- If you selected Incorporate Custom Layout (ICL) during the repository merge, then the utilities list the user interface objects that were affected.
- Lists user interface objects that have missing required fields.

If the postmerge utilities encounter a problem and exit before completing, then 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, then rename the log. The path to the log is as follows:

`SIEBEL_TOOLS_ROOT\reppatch\log\reputility.log`

If you do not rename the log, it will be overwritten.

- 2 In Siebel Tools, from the Screens menu, choose Application Upgrader, and then Application Upgrade Object List.
- 3 In the Application Upgrades list, select the record of the merge.
- 4 In the Application Upgrades list, right-click and select Launch Post Merge Utility from the pop-up menu.

A dialog box appears and displays the postmerge utilities log. When the utilities have finished, a message showing completion displays in the log.

Do not launch more than one instance of the postmerge utilities.

Related Topics

[“About the Siebel Postmerge Utilities” on page 59](#)

[“About the Siebel Repository Merge” on page 51](#)

[Appendix C, “Siebel Incorporate Custom Layout Upgrade Option”](#)

13 Performing the Siebel Incremental Repository Merge

This chapter provides guidelines for performing a Siebel incremental repository merge. It includes the following topics:

- [Process of Meeting the Requirements for an Incremental Upgrade to Siebel Innovation Pack 2016, Using the Incremental Repository Merge on page 187](#)
- [Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x \(SIA Repository\) and Version 15.x on page 190](#)
- [Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x \(SIA Repository\) on page 204](#)
- [Upgrading a Custom Database Schema on page 221](#)
- [About Migrating the XML Manifest Data for Siebel Open UI from Previous Releases to the New Manifest Data in the Database on page 225](#)
- [About Migrating Siebel Web Template Data on page 236](#)
- [Process of Regenerating the Siebel Repository Definition Files on page 237](#)
- [Performing a Production Test or Production Environment Migration from Siebel CRM Version 8.1.1.x \(SIA Repository\) on page 240](#)

NOTE: Use this chapter if you are upgrading from Siebel CRM version 15.0 and later to Siebel CRM version 16.0, upgrading from Siebel CRM version 8.1.1 and later (SIA repository only) to Siebel CRM version 16.0, or upgrading from Siebel CRM version 8.2.2 and later to Siebel CRM version 16.0. For more information about incremental repository merge as it applies to different installation options, see the *Siebel Installation Guide* for the operating system you are using.

Process of Meeting the Requirements for an Incremental Upgrade to Siebel Innovation Pack 2016, Using the Incremental Repository Merge

Environments: Development environment only.

Before you prepare Siebel Tools for the incremental repository merge, you must perform the following tasks:

- 1 ["Before You Begin" on page 188](#)
- 2 ["Preparing Siebel Tools for Custom ODBC Data Source Names on Oracle Database" on page 188](#)
- 3 ["Preparing Siebel Tools for Custom ODBC Data Source Names for All Databases" on page 189](#)
- 4 ["Editing the Siebel Tools Configuration File" on page 190](#)

5 “Copying Ancestor Repositories” on page 130

Before You Begin

Refer to and follow the tasks listed in [Chapter 5, “Preparing for Siebel Database Upgrade.”](#)

You must have the Siebel Enterprise Server, which includes the Siebel Gateway Name Server, the Siebel Server, and the Siebel Database Server, installed. You must have installed Siebel Innovation Pack 2016 (version 16.0). For installation information, see *Siebel Installation Guide* for the operating system you are using and other applicable documentation.

NOTE: Prior to running incremental repository merge, you must have one and only one repository in your development instance and that repository must be named *Siebel Repository*. Also, if you must delete any additional repositories, the only acceptable way to do so is to select the repository record in Siebel Tools and delete it. It is not acceptable to delete repositories using direct SQL.

About the Siebel Server and Siebel Tools on Windows

It is recommended that the Siebel Server and Siebel Tools be installed on the same Microsoft Windows computer. If your Siebel Server is installed on Windows, then the merge process automatically starts Siebel Tools to execute the incremental repository merge. For this reason, the Siebel Database Configuration Wizard must be started from the computer where Siebel Tools is installed.

If the Siebel Server is not installed on the same computer as Siebel Tools, then map a network drive from the computer where Siebel Tools is installed to the computer where Siebel Server is installed. Use this mapped drive to specify the location of the Siebel Server when you run the Siebel Database Configuration Wizard.

NOTE: Because the merge process requires extensive database resources, it is recommended that, before beginning the incremental repository merge process, you set the general cursor size to 5000.

Preparing Siebel Tools for Custom ODBC Data Source Names on Oracle Database

Platforms: Windows, UNIX, and IBM z/OS.

Use this task if you have a custom ODBC data source on Oracle Database.

This task is a step in [“Process of Meeting the Requirements for an Incremental Upgrade to Siebel Innovation Pack 2016, Using the Incremental Repository Merge” on page 187.](#)

If you create a custom ODBC data source name for Oracle Database instead of using the default ODBC data source name, then you must perform the following modifications to enable Siebel Tools for the incremental repository merge. After Siebel Tools is installed, the ODBC data source named *SSD default instance* is automatically set up by the configurations, using the Oracle driver installed by the installation.

To prepare Siebel Tools for a custom data source name on Oracle Database

- 1 Use the Siebel Tools default ODBC driver, Siebel Oracle90.
- 2 Navigate to HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\ODBC_DSN_name.
- 3 Right-click the new custom ODBC name you created, and select New, then String Value, and enter the following names:
 - ColumnsAsChar
 - ColumnSizeAsCharacter
- 4 After you create the values, select each value, right-click each one. For each value, choose Modify, then in the data field, enter the following values:
 - Set the ColumnsAsChar value to 1
 - Set the ColumnSizeAsCharacter value to 1

NOTE: The ODBC SSD default instance automatically has these values in the registry, but you have to manually create them for the ODBC data source that you create.

- 5 Click the System DSN tab of the ODBC connection that you intend to use, and then test the connection.

Preparing Siebel Tools for Custom ODBC Data Source Names for All Databases

This task is a step in [“Process of Meeting the Requirements for an Incremental Upgrade to Siebel Innovation Pack 2016, Using the Incremental Repository Merge”](#) on page 187.

If you create a custom ODBC data source name instead of using the default ODBC data source name, then you must perform the following modifications to enable Siebel Tools for the incremental repository merge.

To prepare Siebel Tools for a custom ODBC data source name for all databases

- 1 Open tools.cfg located at *SIEBEL_TOOLS_ROOT\bin\primary_lang_code*, where *primary_lang_code* is the language code for the primary language.
- 2 In the [Siebel] section of the .cfg file, do the following:
 - a Change the ServerDbODBCDataSource parameter from the default value to your custom ODBC name.
 - b (For Microsoft SQL Server and IBM DB2 only) In the [ServerDataSrc] section, change the Connect String parameter from the default value to your custom ODBC name.

Editing the Siebel Tools Configuration File

Platforms: Windows, UNIX, and IBM z/OS.

This task is a step in [“Process of Meeting the Requirements for an Incremental Upgrade to Siebel Innovation Pack 2016, Using the Incremental Repository Merge”](#) on page 187. Use this task to perform the required edits in the Siebel Tools configuration file.

To edit the Siebel Tools configuration file

- 1 Navigate to `SIEBEL_TOOLS_ROOT\bin\lang_code`, and open the tools.cfg file.
- 2 Make the following changes in the tools.cfg file:
 - a Change the SymStrPrefix value from the default value of X_ to SBL_.
CAUTION: You must change these values back to the default value after the merge but before compiling the SRF file. It is required that you set the prefix to SBL_ to differentiate between symbolic strings you create and those provided by Siebel CRM. After the merge, to enable you to create your own custom symbolic strings, you must return the prefix back to X_. To do this, see [“Editing the Siebel Tools Configuration File After the Development Environment Merge”](#) on page 239.
 - b Make sure the EnableToolsConstrain value is set to the default value of False. If it is not, then set the value to False.
 - c Set the DockRepositoryName value from the default value of Siebel Repository to Prior Customer Repository.
- 3 Save the tools.cfg file.
- 4 You do this because incremental repository merge requires this prefix to be set to SBL_ so that all the symbolic strings that are included as part of the New Siebel Repository are inserted into your repository during the merge.

Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x (SIA Repository) and Version 15.x

Environments: Development, production test, and production.

The tasks in this process enable you to merge customizations in the Prior Customer Repository (your current repository) with the New Customer Repository (the repository in the new Siebel CRM release). This process covers upgrades from Siebel CRM version 8.1.1.x, 8.2.2.x (SIA repository), and Siebel CRM version 15.x.

To perform incremental repository merge on your Siebel repository, perform the following tasks:

- 1 [“Preparing to Run the Siebel Database Configuration Wizard”](#) on page 191
- 2 [“Requirements for Running the Siebel Database Configuration Wizard on Windows”](#) on page 195

- 3 "Running the Siebel Database Configuration Wizard on Windows" on page 195
- 4 "Running the Siebel Database Configuration Wizard on UNIX" on page 196
- 5 "Preparing to Start the Siebel Upgrade Wizard" on page 197
- 6 "Preparing to Restart the Merge" on page 200
- 7 "Restarting the Merge" on page 202
- 8 "Starting the Siebel Upgrade Wizard" on page 198
- 9 "Stopping the Siebel Upgrade Wizard on Windows" on page 199
- 10 "Stopping the Siebel Upgrade Wizard on UNIX" on page 200
- 11 "Executing the RUNSTATS Command on Oracle Database During the Pause Following New Repository Creation" on page 202
- 12 "Executing the RUNSTATS Command on IBM DB2 During the Pause Following New Repository Creation" on page 203
- 13 "Executing the RUNSTATS Command on Microsoft SQL Server During the Pause Following New Repository Creation" on page 204

Preparing to Run the Siebel Database Configuration Wizard

This task is a step in "Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x (SIA Repository) and Version 15.x" on page 190. Use this task to prepare for running the Siebel Database Configuration Wizard.

Environments: Development, production test, and production.

About the Database Configuration Utilities

The Database Configuration utilities are a group of wizards that request information about the upgrade process that you want to perform. The utilities add this information to a master upgrade file and call an SQL generator. The SQL generator uses the information to create or populate SQL files. See [Table 27 on page 192](#) for a list of the information that the utilities request when you perform an upgrade.

To prepare to run Database Configuration utilities

- 1 Identify the information that you must enter when running the Database Configuration utilities.
- 2 Use [Table 27](#) as a guide to collect all the required information.

After the Database Configuration utilities exit, you run the Siebel Upgrade Wizard. The Siebel Upgrade Wizard executes the SQL files against the Siebel database. After running the Database Configuration utilities, you run the Siebel Upgrade Wizard. For information on running the Siebel Upgrade Wizard, see [“Preparing to Start the Siebel Upgrade Wizard” on page 197](#).

Table 27. Information Required for the Database Configuration Utilities

Field Name or Menu	Required Information
Enterprise Server Name	The Enterprise Server name.
Siebel Server Directory	The absolute path of the directory where the Siebel Server is installed. For UNIX, do not enter the string \$SIEBEL_ROOT.
Siebel Database Server Directory	The absolute path of the directory where the Siebel Database Server is installed. For example: C:\sba81\dbsrvr
RDBMS Platform	Choose the RDBMS type: IBM DB2, Microsoft SQL Server, or Oracle Database. NOTE: For IBM DB2 for z/OS, see <i>Siebel Database Upgrade Guide for DB2 for z/OS</i> instead of this guide.
Siebel Database Operation menu	Choose Upgrade Database. The remaining menu choices are for database installation and administration.
Environment Type	Choose Development for development environment upgrades. Choose Production for production test environment and production environment upgrades.
Upgrade Options	Choose one of the following: <ul style="list-style-type: none"> ■ Development Environment: <ul style="list-style-type: none"> ■ Upgrade Siebel Database Schema (upgrep) ■ Upgrade Custom Database Schema (upgphys) ■ Production Environment. Choose Upgrade Siebel Database Schema (upgrep and upgphys).
Siebel Industry Application	As appropriate for the environment you are upgrading from, choose SIA (for Siebel Industry Applications) or SEA (for Siebel Business Applications, also known as Siebel Cross-Industry Applications).

Table 27. Information Required for the Database Configuration Utilities

Field Name or Menu	Required Information
Current Siebel Version	<p>Choose the option for the application version that you are upgrading your Siebel database from (where you previously had installed a new database or run Incremental Repository Merge, where applicable), based on the following criteria. Do this after installing the current release of Siebel CRM as a new installation or as a migration installation from the indicated prior version.</p> <ul style="list-style-type: none"> ■ If your Siebel database is currently on version 15.5, then choose v15_5. ■ If your Siebel database is currently on version 15.0, then choose v15_0. ■ If your Siebel database is currently on version 8.1.1.14, then choose v8_1_1_14. ■ If your Siebel database is currently on version 8.1.1.11, then choose v8_1_1_11. ■ If your Siebel database is currently on version 8.1.1.10, then choose v8_1_1_10. ■ If your Siebel database is currently on version 8.1.1.0 (SIA only) to 8.1.1.9, then choose v8_1_1SIA_To_v8_1_1_9SIA. ■ If your Siebel database is currently on version 8.2.2.14, then choose v8_2_2_14. ■ If your Siebel database is currently on version 8.2.2.4, then choose v8_2_2_4. ■ If your Siebel database is currently on version 8.2.2.3, then choose v8_2_2_3. ■ If your Siebel database is currently on version 8.2.2.0 to 8.2.2.2, then choose v8_2_2SIA_To_v8_2_2_2SIA.
Siebel Tools Installation Directory	<p>The absolute path of the directory where Siebel Tools is installed, such as c:\Siebel\8.1\Tools.</p> <p>NOTE: This entry applies only to the Siebel Server installed on Windows.</p>
Siebel Tools Data Source	<p>Provide the data source name that you use to log in using Siebel Tools, such as ServerDataSrc. This entry is in <i>SIEBEL_TOOLS_ROOT\BIN\lang_code\tools.cfg</i>.</p>
Database Encoding	<p>Indicate whether your database uses a Unicode code page.</p>
ODBC Data Source Name	<p>Verify the ODBC name for connecting to the Siebel database that you are upgrading. If the ODBC name is not correct, then enter the correct name.</p>

Table 27. Information Required for the Database Configuration Utilities

Field Name or Menu	Required Information
Database User Name and Database Password	Account name and password for the Siebel administrator of the Siebel database that you are upgrading. NOTE: For more information about supported characters for Siebel passwords, see <i>Siebel Security Guide</i> . See also relevant articles on My Oracle Support, such as 1931241.1 (Article ID).
Database Table Owner and Database Table Owner Password	Account name and password for the Siebel database table owner. NOTE: For more information about supported characters for Siebel passwords, see <i>Siebel Security Guide</i> . See also relevant articles on My Oracle Support, such as 1931241.1 (Article ID).
Index Table Space Name and Table Space Name	Oracle Database and IBM DB2 only. Index tablespace name and tablespace name (4-KB tablespace name for IBM DB2).
16-KB Table Space Name, 32K Table Space Name	IBM DB2 only. The 16-KB and 32-KB tablespace names.
Database Server OS	Choose the RDBMS server operating system type.
Parallel Indexing	Oracle Database only. Select parallel indexing if you want SQL commands for index creation to include the arguments parallel and no logging. Parallel indexing causes an index to be created using parallel processing, which requires an RDBMS server with multiple processors. Verify with your database administrator whether your RDBMS server is configured for parallel processing. TIP: Oracle Library search phrase: parallel execution. Selecting parallel indexing does not cause multiple indexes to be created simultaneously, in parallel. To set up parallel indexing, you must set up parallel index-creation threads, using Siebel Upgrade Tuner. You create parallel threads as part of tuning the production upgrade files. For more information, see Chapter 18, "Tuning the Siebel Upgrade Files."
Security Group ID/Grantee	Security group or grantee name for Siebel application users. Must have select, update, insert, and delete privileges on Siebel application tables. Specify SSE_ROLE.
Verify Repository After Upgrade	Indicate whether you want to execute the steps to verify the repository during upgphys. To perform upgphys separately, select the Verify Repository After Upgrade option in the Database Configuration Wizard.

Table 27. Information Required for the Database Configuration Utilities

Field Name or Menu	Required Information
Upgrep log directory	If you select the option Verify Repository After Upgrade in the previous step, then you will have to provide the log directory of the upgrep process. The log directory is of the form <code>\$SIEBEL_ROOT/siebsrvr/Log/upgrep_dev_UpgradeNumber</code> . For example: <code>C:\ses\siebsrvr\Log\upgrep_dev_811</code> . The log directory path is a requirement for generating the seed data conflict report.
Log Output Directory	Specify a different subdirectory, under the log directory, in which to create the log files. Accept the default or enter the directory name. If the directory does not exist, then it will be created. Do not use special characters, such as spaces or slashes.
Select runupg option	Indicate whether you want to run the operation that you configured or run it at another time.

Requirements for Running the Siebel Database Configuration Wizard on Windows

This task is a step in [“Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x \(SIA Repository\) and Version 15.x”](#) on page 190.

Do the following:

- Collect the information that the Siebel Database Configuration Wizard requires.
For more information, see [“Preparing to Run the Siebel Database Configuration Wizard”](#) on page 191.
- If your Siebel Server is installed on Windows, then start the Siebel Database Configuration Wizard from the computer where Siebel Tools is installed.
The merge process automatically starts Siebel Tools to execute the incremental repository merge.
- If the Siebel Enterprise server is not installed on the same computer as Siebel Tools, then map a network drive from the computer where Siebel Tools is installed to the computer where Siebel Enterprise Server is installed. Use this mapped drive to specify the location of the Siebel Server when you run the Siebel Database Configuration Wizard.

Running the Siebel Database Configuration Wizard on Windows

This task is a step in [“Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x \(SIA Repository\) and Version 15.x”](#) on page 190. Use this task to run the Siebel Database Configuration Wizard on Windows.

Environments: Development, production test, and production.

Platforms: Windows only.

Run the Siebel Database Configuration Wizard to upgrade the Siebel database. The Wizard collects information, populates a master configuration file, and calls the SQL generator to create SQL commands. The Siebel Upgrade Wizard then uses the configuration file and SQL commands to execute the incremental repository merge process to incrementally upgrade the database to the newer release.

To run the Siebel Database Configuration Wizard on Windows

- 1 Ensure that no server tasks except the Siebel Gateway Name Service are running in the background.
To verify this step, navigate to Start, Settings, Control Panel, and then Services.
- 2 Start the Siebel Database Configuration Wizard by selecting Start, Programs, Siebel SES Configuration, and then Database Server Configuration.
- 3 In the Siebel Database Configuration Wizard, enter the information requested in each screen, and click Next.
- 4 After you have entered all the requested information, the wizard displays a screen that lists the values you entered. If you want to make changes, then click Back.
- 5 When a dialog box is displayed asking whether you want to start the Siebel Upgrade Wizard, click Yes.

The Upgrade Wizard starts.

Running the Siebel Database Configuration Wizard on UNIX

This task is a step in [“Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x \(SIA Repository\) and Version 15.x” on page 190](#). Use this task to run the Siebel Database Configuration Wizard on UNIX.

Environments: Development, production test, and production.

Platforms: UNIX only.

Run the Siebel Database Configuration Wizard to upgrade the Siebel database. The Wizard collects information, populates a master configuration file, and calls the SQL generator to create SQL commands. You then run the Siebel Upgrade Wizard to upgrade the Siebel database.

You must meet the following requirements:

- Before running the Siebel Database Configuration Wizard, you must install the language packs of the new release of Siebel CRM for each language that you deploy.
- Collect the information that the Siebel Database Configuration Wizard requires. For more information, see [“Preparing to Run the Siebel Database Configuration Wizard” on page 191](#).

To run the Siebel Database Configuration Wizard on UNIX

- 1** Verify that all servers are stopped:
 - a** Stop all Siebel Servers.
 - b** Stop the Siebel Gateway Name Server.
- 2** Make `$SIEBEL_ROOT` the current directory.
- 3** Source the environment variables from the `si ebsrvr` root directory: *install_location/si ebsrvr*

Korn shell: `si ebenv. sh`

C shell: `source si ebenv. csh`
- 4** Review the values of the following environment variables, and confirm that the settings are correct:
 - `SIEBEL_ROOT`. This path must end in `si ebsrvr`, for example, `/usr/siebel/si ebsrvr`.
 - `LANGUAGE`. This is the language in which the Siebel Database Configuration Wizard runs. The value of this variable is a language identifier string. For example, `enu` is the identifier string for U.S. English.

If either `$SIEBEL_ROOT` or `$LANGUAGE` is not set or is incorrect, then you must correct it before proceeding.
- 5** Start the Siebel Database Configuration Wizard:

`$SIEBEL_ROOT/config/config. sh -mode dbsrvr`
- 6** Enter the information requested in each screen, and click Next.

After you have entered all the requested information, the wizard displays a screen that lists the values you entered.
- 7** If you want to make changes, then click Back.

The Siebel Database Configuration Wizard exits and prompts you to start the Siebel Upgrade Wizard (`srvrupgwis`).

Preparing to Start the Siebel Upgrade Wizard

This task is a step in [“Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x \(SIA Repository\) and Version 15.x” on page 190](#). Use this task to run the Siebel Upgrade Wizard.

Environments: Development, production test, and production.

When you run the Database Configuration Utilities on Windows, there is a prompt asking whether you want to start the Siebel Upgrade Wizard. When you run the Database Configuration Utilities on UNIX, you must start the Siebel Upgrade Wizard manually.

The Siebel Upgrade Wizard executes the upgrade of the Siebel database. The Siebel Upgrade Wizard takes a master configuration file as input. This file contains environment information and a driver file name. The Siebel Upgrade Wizard executes the steps in the driver file to perform the upgrade.

Because the Siebel Upgrade Wizard performs the steps in the driver file, it lists the steps in a state log. The state log is located in the following directory:

```
si ebsrvr/LOG/process/state
```

In the directory, *process* refers to the upgrade process. For example, *upgrep_dev_811sia* indicates that the upgrade is from 8.1.1, and that the upgrep process is being run in the production test or production environment.

Starting the Siebel Upgrade Wizard

This task is a step in [“Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x \(SIA Repository\) and Version 15.x” on page 190.](#)

Use this procedure to start the Siebel Upgrade Wizard.

To start the Siebel Upgrade Wizard

1 Navigate to the following directory:

- Windows: *SIEBEL_ROOT*\bin
- UNIX: *\$SIEBEL_ROOT*/bin

2 Enter the following command:

- Windows: `si ebug /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`
- UNIX: `srvrupgwi z /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`

In these commands, *UPGRADEOPTION_ENVIRONMENT_VERSION* is the portion of the upgrade configuration file name that lists the upgrade process, the upgrade environment, and the Siebel CRM release (version) from which you are upgrading. The file is located in *SIEBEL_ROOT*\bin (UNIX: *\$SIEBEL_ROOT*/bin).

Numbers like the following examples are used for the Siebel CRM release portion of the file name:

- 811sia
- 81114

Table 28 lists an example of the file names for an upgrade from Siebel CRM version 8.1.1 (SIA).

Table 28. Example Upgrade Configuration File Names for Upgrade from Siebel 8.1.1 (SIA)

Upgrade Mode	File Name
Development environment upgrep	master_upgrep_dev_811sia.ucf
Development environment upgphys	master_upgphys_dev_811sia.ucf
Production environment upgrep and upgphys	master_upgrep_prod_811sia.ucf

Table 29 lists an example of the file names for an upgrade from Siebel CRM version 8.1.1.11 (SIA).

Table 29. Example Upgrade Configuration File Names for Upgrade from Siebel 8.1.1.11 (SIA)

Upgrade Mode	File Name
Development environment upgrep	master_upgrep_dev_81111.ucf
Development environment upgphys	master_upgphys_dev_81111.ucf
Production environment upgrep and upgphys	master_upgrep_prod_81111.ucf

- 3 To begin the upgrade, click OK (Windows), or press Enter (UNIX).

The Siebel Upgrade Wizard will notify you when the upgrade process is complete.

Stopping the Siebel Upgrade Wizard on Windows

This task is a step in “Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x (SIA Repository) and Version 15.x” on page 190.

Only stop the Siebel Upgrade Wizard if you are confident that an error has occurred and the Siebel Upgrade Wizard or a utility it has called has stopped responding. The reason for this caution is that some SQL commands issued by the Siebel Upgrade Wizard can be very time intensive and stopping and restarting the Siebel Upgrade Wizard will slow down your upgrade process.

If you are not sure whether the Siebel Upgrade Wizard has stopped responding, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle’s Application Expert Services.

To stop the Siebel Upgrade Wizard on Windows

Do one of the following:

- If the Siebel Upgrade Wizard has started a separate command window in which a utility is running, then close the command window. This step terminates the utility and stops the upgrade.
- In the Siebel Upgrade Wizard dialog box, click Cancel. The Siebel Upgrade Wizard exits when the current upgrade step is complete. There might be a delay while the step completes in the RDBMS.

Stopping the Siebel Upgrade Wizard can have different effects on the RDBMS. Before restarting the Siebel Upgrade Wizard, review the RDBMS log files. Run SQL commands to resolve errors found in the RDBMS log files.

Stopping the Siebel Upgrade Wizard on UNIX

This task is a step in [“Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x \(SIA Repository\) and Version 15.x” on page 190.](#)

Only stop the Siebel Upgrade Wizard if you are confident that an error has occurred and the Siebel Upgrade Wizard or a utility it has called has stopped responding. The reason for this caution is that some SQL commands issued by the Siebel Upgrade Wizard can be very time intensive and stopping and restarting the Siebel Upgrade Wizard will slow down your upgrade process.

If you are not sure whether the Siebel Upgrade Wizard has stopped responding, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle’s Application Expert Services.

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

To stop the Siebel Upgrade Wizard on UNIX

- 1 If the Siebel Upgrade Wizard has started a utility in a child process, then stop the child process.
- 2 Exit the shell in which the Siebel Upgrade Wizard is running.
- 3 Locate and stop any orphaned child processes started by the Siebel Upgrade Wizard.

After the processes terminate, there might be a delay while the RDBMS executes the issued SQL commands.

Preparing to Restart the Merge

This task is a step in [“Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x \(SIA Repository\) and Version 15.x” on page 190.](#)

Platforms: Windows, UNIX, and IBM z/OS.

Before restarting the merge, consider the following requirements:

If the Siebel Upgrade Wizard encounters an error and exits during an upgrade, then you can restart it after correcting the error. The Siebel Upgrade Wizard reads the state log and continues the upgrade from the last successfully completed step.

If the Siebel Upgrade Wizard stops because of errors, then verify that you have met these requirements before restarting the wizard:

- Carefully review the relevant log files to make sure that your upgrade has completed successfully up to that point.
- Back up your complete set of log files, from the beginning of the process to the point at which it stopped, to another directory.

This backup maintains a complete record of your log files, and it prevents your previous log files from being overwritten, which might prevent an accurate diagnosis of the reason for the error in the upgrade.

- If you are continuing a previous and incomplete schema upgrade, then do not change the path of the Log Output Directory that you previously selected.
- If problems with your environment prevent the upgrade from restarting, then you must restore the database from the prior base version; that is, the version from which you are upgrading. For example, environment problems might occur when table creation fails because of a database problem (insufficient storage or network problems), which causes subsequent upgrade steps to fail.

If you want to restore your database and restart the upgrade, then delete or store the upgrade log files. The files are located in the following directory:

Windows: `SI EBEL_ROOT\log\PROCESS\output`

UNIX: `$SI EBEL_ROOT/log/PROCESS/output`

Also delete the state. log file. It is located in the following directory:

Windows: `SI EBEL_ROOT\log\PROCESS\state`

UNIX: `$SI EBEL_ROOT/log/PROCESS/state`

Requirements for Restarting the Incremental Repository Merge

Platforms: Windows, UNIX, and IBM z/OS.

If the incremental repository merge stops because of errors, then you must review the log files, and restart the merge.

It is recommended that you perform the following tasks to preserve the premerge environment in the event of a failed merge.

- 1 Perform a full database backup.** Before the merge, back up the entire database. If the merge fails, then you can restore the database to its premerge state, and rerun the merge operation.

- 2 Export the new customer repository.** Before the merge, export the New Customer Repository to create a backup copy. If the merge fails, then delete the failed repository, and import the backed-up copy of the New Customer Repository. See *Using Siebel Tools* for information on exporting and importing repositories using the Siebel Database Configuration Wizard.

NOTE: If you export the New Customer Repository, then you must truncate the following merge log tables: S_MERGE_LOG, S_MERGE_LOG_OBJ, and S_MERGE_LOG_ATTR.

Restarting the Merge

This task is a step in “[Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x \(SIA Repository\) and Version 15.x](#)” on page 190.

To restart the merge

- 1** Restore the database to the state of the backup taken before the merge.
You must have already executed database statistics.
- 2** Review the irm_sav_file.ssf file located in `SIEBEL_TOOLS_ROOT\bin`. The State value should equal 7 (State=7). If it is any other value, then you must change it to 7 before starting the incremental repository merge again.
- 3** Do one of the following:
 - **On a Windows computer.** Restart the Execute Incremental Repository Merge task by marking the step as Incomplete in the state.log file located in the Siebel Server installation directory, in `si_ebsrvr\log\upgrade`.
 - **On a UNIX computer.** Execute the incremental repository merge through the command line on your Windows computer.

For more information, see “[Starting the Siebel Upgrade Wizard](#)” on page 198 to restart the upgrade process.

Executing the RUNSTATS Command on Oracle Database During the Pause Following New Repository Creation

This task is a step in “[Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x \(SIA Repository\) and Version 15.x](#)” on page 190. Use these tasks to run database checks on your specific database platform after your new repository has been created. The Upgrade Wizard pauses notifying you to execute RUNSTATS and then resume. At this point paused moment, the RUNSTATS command has to be executed.

To execute the RUNSTATS command on Oracle Database

- Connect to SQL plus, and execute the following command:

```
EXEC DBMS_STATS.gather_schema_stats (ownname => ' table_owner_name' , cascade =>true, estimate_percent => dbms_stats.auto_sample_size);
```

Example:

Invoke sqlplus from the Oracle client.

```
SQL*Plus: Release 11.2.0.1.0 Production on Wed Nov 21 19:21:46 2012
Copyright (c) 1982, 2009, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Oracle Label Security, OLAP, Data Mining,
Oracle Database Vault and Real Application Testing options
```

```
SQL> EXEC DBMS_STATS.gather_schema_stats (ownname => 'ORAZQ108' , cascade =>true, estimate_percent => dbms_stats.auto_sample_size);
PL/SQL procedure successfully completed.
```

Executing the RUNSTATS Command on IBM DB2 During the Pause Following New Repository Creation

This task is a step in [“Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x \(SIA Repository\) and Version 15.x” on page 190](#). Use this task to execute the RUNSTATS command on IBM DB2, or see your database administrator to have the command executed. The Upgrade Wizard pauses notifying you to execute RUNSTATS and then resume. At this point paused moment, the RUNSTATS command has to be executed.

To execute the RUNSTATS command on IBM DB2

- Connect to the IBM DB2 command window and execute the following commands:

```
echo "select 'runstats on table ' concat rtrim(creator) concat '.' concat
rtrim(Name) concat ' and detailed indexes all shrlevel change' concat ';' from
sysibm.systables where type='T' and (name not in
('S_ESCL_LOG' , 'S_DOCK_INIT_ITEM' ) and name not like ('S_DOCK_INITM_%' )) order by
name ;">DBname_tmp_runstats.sql
```

Example:

```
db2 connect to db204205 user siebel using db2
db2 -tf DB204205_tmp_runstats.sql | grep runstats | grep -v ^select >
DB204205_runstats.sql
db2 terminate
```

```
db2 connect to db204205 user siebel using db2
db2 -tf DB204205_runstats.sql
db2 terminate
```

Executing the RUNSTATS Command on Microsoft SQL Server During the Pause Following New Repository Creation

This task is a step in “[Process of Performing an Incremental Upgrade of the Siebel Database from Siebel CRM Version 8.1.1.x and 8.2.2.x \(SIA Repository\) and Version 15.x](#)” on page 190. Use this task to execute the RUNSTATS command on Microsoft SQL Server. The Upgrade Wizard pauses notifying you to execute RUNSTATS and then resume. At this point paused moment, the RUNSTATS command has to be executed.

To execute the RUNSTATS command on Microsoft SQL Server

- 1 Navigate to `SIEBEL_TOOLS_ROOT\bin` on the command window, and log in to `odbcsql`, using the following command:

```
/u SADMIN /p MSSQL /s ODBC_data_source_name
```

- 2 Execute the following command:

```
Exec sp_updatestats;
```

Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x (SIA Repository)

The tasks in this process enable you to perform incremental upgrade of your Siebel development environment from Siebel CRM version 8.1.1.x.

Environments: Development environment only.

For more information on Siebel Database Configuration Wizard parameters, see “[Preparing to Run the Siebel Database Configuration Wizard](#)” on page 191. For information on executing the Siebel Database Configuration Wizard, see “[Running the Siebel Database Configuration Wizard on Windows](#)” on page 195.

Perform the following tasks:

- 1 “[Upgrading the Siebel Database Schema](#)” on page 205
- 2 “[Executing Incremental Repository Merge on UNIX](#)” on page 213
- 3 “[About Siebel Repository Merge Errors](#)” on page 213
- 4 “[Reviewing the Siebel Repository Merge Log Files](#)” on page 215
- 5 “[Reviewing the Hierarchical Merge Report](#)” on page 216
- 6 “[Reviewing Siebel Repository Object Property Conflicts](#)” on page 218
- 7 “[Reviewing Log Files for All Upgrades](#)” on page 220
- 8 “[Reviewing Conflicts at the Attribute Level](#)” on page 220

- 9 [“Marking Conflict Resolution as Complete Using Siebel Tools”](#) on page 221

Upgrading the Siebel Database Schema

This task is a step in [“Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x \(SIA Repository\)”](#) on page 204. Use this task to upgrade the Siebel database schema.

NOTE: Before you upgrade the Siebel database schema using the upgrep process, it is strongly recommended to back up the entire database. This step is mentioned in [Table 30 on page 206](#), as [Step 22 on page 211](#). If this upgrade operation fails, then you can restore the database to its preupgrade state, and rerun the upgrade operation.

To upgrade the Siebel database schema

- 1 Execute the Siebel Database Configuration Wizard by specifying the following parameter values:

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

- 2 Start the Siebel Upgrade Wizard.

The Siebel Upgrade Wizard reads the steps from the driver file, which is determined by your Siebel CRM version. The Siebel Upgrade Wizard performs the commands listed in the steps. The driver file shown in [Table 30 on page 206](#) contains the following information:

- Environment: Development
- Upgrade mode: upgrep
- Database: Oracle
- Multilingual: No

- 3 After the incremental repository merge has been completed successfully, review the merge log files in `SIEBEL_TOOLS_ROOT\log`.

See [Table 30](#) for the Siebel Upgrade Wizard steps.

Table 30. Steps That the Siebel Upgrade Wizard Executes

Step	Script or Input File	Description	Comments
1 Verify Siebel Version	verify_siebel_ver.sql	Verifies whether the correct Siebel CRM version (from which you are upgrading) was selected when the Siebel Database Configuration Wizard is started.	If this step stops because of an error, then it was because the wrong Siebel CRM version was selected during the launch of the Database Configuration Wizard. Select the Current Siebel Version as explained in Table 27 on page 192 .
2 Verify Repository Name	rename_existing_repositories.sql	Renames the Siebel Repository to Prior Customer Repository.	None.
3 Drop Interface tables	dropif-db.sql	Removes all Siebel Enterprise Integration Manager tables.	None.
4 Drop triggers	trigdrop-db.sql	Removes all dynamically created triggers.	None.
5 Remove database-level functions and procedures	drop_db_func_proc.sql	Removes the exchange rate function: exrate.	None.
6 Prepare for table creation	pret_irm.sql	Performs DDL operations, such as adding columns to tables. Performs DML operations, such as revising date formats.	None.
7 Create and update tables	ddlmp utility ddl.ctl as input	The ddl.ctl file specifies the structure of tables to be created or updated.	None.

Table 30. Steps That the Siebel Upgrade Wizard Executes

Step	Script or Input File	Description	Comments
8 Prepare for index creation	preschm_irm.sql	Performs DML operations. Moves data between tables. Changes the data in the existing fields, depending on the specified conditions.	None.
9 Create indexes	ddlmp utility ddl.ctl as input	The input file specifies the structure of indexes to be created.	None.
10 Import seed data	dataimp utility seedupg0.inp as input seedupg1.inp as input seedupg_locale.inp as input	Before importing seed data, the dataimp utility deletes the existing seed data. The three seedupg files contain filters that the dataimp utility uses to prevent the deletion of seed data that you have modified or seed data that meets specified criteria. Unmodified seed data has a last update date (LAST_UPD) of 1980-01-01. The dataimp utility does not delete records in which the last update date (LAST_UPD) is later than this date.	None.
11 Upgrade data after seed data import	upg_data_afterseed.sql upg_data_afterseed_sia.sql	Updates existing seed data if any new columns have to be populated.	None.
12 Set system pref for Codepage DB	set_codepage.sql	Sets the database code page in the S_SYS_PREF table.	None.
13 Set system pref for unicode DB	set_unicode.sql	Sets the Unicode code page to UTF-8 in the S_SYS_PREF table.	None.

Table 30. Steps That the Siebel Upgrade Wizard Executes

Step	Script or Input File	Description	Comments
14 Update version component information	upd_upgcomp.sql	Updates the S_UPG_COMP table with the product release level. Also updates the location of the SRF file. The S_UPG_COMP table stores version information for application executables.	None.
15 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.	None.
16 Update Siebel database version	update_ver.sql seeduver.sql	The update_ver.sql script creates a temporary table, S_APP_VER_TEMP, which contains new version information for the database schema. The seeduver.sql script updates the S_APP_VER table with this information.	None.
17 Export Prior Customer Repository	repimexp utility Export Prior Customer Repository	Exports the existing Prior Customer Repository to create New Customer Repository.	None.

Table 30. Steps That the Siebel Upgrade Wizard Executes

Step	Script or Input File	Description	Comments
18 Import Prior Siebel, New Siebel and New Customer repositories in parallel	repimexp utility Import the Prior Siebel Repository, New Siebel Repository, and New Customer Repository.	Imports the Prior Siebel Repository, New Siebel Repository, and New Customer Repository in parallel to the repository tables.	<p>If this step produces an error while creating the New Customer Repository, check the log file in: <i>SIEBEL_ROOT\si ebsrvr \log\upgrade_path\ou tput</i>).</p> <p>For Oracle Database: If you receive an error similar to the following, then you must restart the upgrade from the previous step <i>Export Prior Customer Repository</i> by changing the state.log of that step to Incomplete.</p> <p>Error message: [DataDirect][ODBC Oracle driver][Oracle]ORA-12899: value too large for column "S_ACCELERATOR"."INACTIVE_FLG" (actual : 4, maximum: 1)</p>

Table 30. Steps That the Siebel Upgrade Wizard Executes

Step	Script or Input File	Description	Comments
19 Install SQL packages	seeduver.sql	Verifies that correct versions are set in the S_APP_VER table.	None.
	ifstrg.sql	Sets the storage parameters for Siebel Enterprise Integration Manager tables.	None.
	ifindxstrg.sql	Sets the storage parameters for Siebel Enterprise Integration Manager table indexes.	None.
	pkgseq.sql	Adds a suffix to row IDs in the S_SEQUENCE table. Ensures that row IDs are unique. The pkgldel.sql script defines s_txn_log_del_proc. The procedure periodically deletes transactions from the S_DOCK_TXN_LOG file. It also deletes rows from the S_DOCK_TXN SET table. This action prevents the need for a large rollback segment.	None.
	trgreset.sql	Ensures that denormalized rows in the S_TERR table have the correct values.	None.
	ddlseq.sql	Sets the sequence numbers for the specified tables.	None.
	pkgvis.sql	Creates a function that modifies how Oracle Optimizer performs the visibility check.	None.
	20 Create temporary indexes for merge performance	crt_temp_indexes_merge.sql	Creates temporary indexes to improve the merge performance.

Table 30. Steps That the Siebel Upgrade Wizard Executes

Step	Script or Input File	Description	Comments
21 Migrate SWT from files into the Siebel database	SWTMigration.jar	This step migrates Siebel Web template data from files into the Siebel database. For more information, see “About Migrating Siebel Web Template Data” on page 236.	None
22 Take backup of the Siebel database and update database statistics	None	Pauses the Siebel Upgrade Wizard to enable a database backup and the execution of database statistics before the merge.	At this point in the upgrade, back up the database, and execute database statistics, then click Yes to continue the upgrade. To stop at this step and restart the wizard later, click No. To resume the upgrade, see “Preparing to Start the Siebel Upgrade Wizard” on page 197.
23 Execute Incremental Repository Merge	siebdev	Starts the incremental repository merge automatically for Windows. If you are using a Windows computer and Siebel Tools is installed for Siebel Enterprise servers in UNIX environment, then you must perform this step manually.	For UNIX or Linux: You must execute the incremental repository merge process manually. (On Microsoft Windows, incremental repository merge starts automatically.) See “Executing Incremental Repository Merge on UNIX” on page 213.

Table 30. Steps That the Siebel Upgrade Wizard Executes

Step	Script or Input File	Description	Comments
24 Generate Merge Report	MergeReport.jar	<p>Generates the hierarchical merge report in HTML format.</p> <p>Also creates the file IRM_Merge*_ERROR.txt in <i>SIEBEL_TOOLS_ROOT</i>\log, which contains only the errors that are present in the IRM_Merge*.txt file.</p> <p>For more information about the hierarchical merge report, see “Reviewing the Hierarchical Merge Report” on page 216.</p>	<p>For UNIX or Linux: You must generate the hierarchical merge report manually, by running MergeReport.bat. Typically, you run this report after running incremental repository merge. (On Microsoft Windows, this report runs automatically.)</p>
25 Export Merge Data to CSV Script or Input File	ExportMrgCSV.jar	<p>Exports repository-merge log data to a CSV file for business object, business component, applet, view, and Web page object definitions. This data can be compared with the Usage Pattern Tracking (UPT) CSV files to obtain the intersection data, so that you can focus testing on these objects only.</p> <p>The Siebel Upgrade Wizard exports the log data to CSV files in the following folder:</p> <p><i>upgrep_log_directory/output/export_csv/csv</i></p>	<p>This step is executed only when you run IRM after migrating from a prior Siebel CRM release in which UPT was enabled (for example, version 15.5 or later). For information about the specific releases (including applicable patchset releases) for which UPT can be enabled, see 2145521.1 (Article ID) on My Oracle Support.</p> <p>For more information about Usage Pattern Tracking, see <i>Siebel Applications Administration Guide</i>.</p>
26 Record Upgrade History	store_history.sql	Stores the upgrade history in the S_INST_UPG_HIST table.	None.

Executing Incremental Repository Merge on UNIX

This task is a step in “[Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x \(SIA Repository\)](#)” on page 204. Use this task to execute incremental repository merge on UNIX.

Incremental repository merge is started automatically on Windows, but on UNIX you must initiate the process manually. You begin executing the incremental merge when the Upgrade Wizard pauses and notifies you to execute the manual merge.

To execute incremental repository merge on UNIX

- 1 Copy `irm_sav_file.ssf` from `$SIEBEL_ROOT\si_ebsrvr\bin` on the UNIX computer to `SIEBEL_TOOLS_ROOT\bin` on the Windows computer.
- 2 Execute the following command to start the incremental repository merge:

```
si_ebdev /u UserName /p Password /d "ToolsDataSource" /c  
"SIEBEL_TOOLS_ROOT\BIN\lang_code\tools.cfg" /i Packmode /IRM UpgDel taMerge
```
- 3 Click Yes to continue.

To pause now and resume later, click No. To resume, see “[Starting the Siebel Upgrade Wizard](#)” on page 198.

If the merge fails, see “[Requirements for Restarting the Incremental Repository Merge](#)” on page 201.

About Siebel Repository Merge Errors

This task is a step in “[Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x \(SIA Repository\)](#)” on page 204. Use this task to review Siebel Repository merge log files.

Environments: Development environment only.

Platforms: Windows, UNIX, and IBM z/OS.

Acceptable and Unacceptable Errors

To determine whether the repository merge was successful, review the merge log files. The merge is successful if it completes without unacceptable errors:

- **Acceptable errors.** If an ancestor object is specified in an object definition, and if the ancestor object is not present in the New Siebel Repository, then a merge error occurs. This error is acceptable, and you can ignore it. The following is an example of an acceptable error in the merge log file, IRM_merge0.txt:

```
!!ERROR: : CANNOT upgrade objects which have Briefing Tracking Profile Applet -  
Product marked as 'Upgrade Anc'.
```

If an object already exists in the New Customer Repository that has come from any Release Feature (RF) from Siebel CRM version 8.1.1.x or 8.2.2.x, then a merge error results stating that the object cannot be inserted because the same values already exist. This error is acceptable.

- **Unacceptable errors.** All other types of merge errors are unacceptable and indicate that the merge was not successful. For more information, see 477269.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Troubleshooting Steps 19.

Merge errors are displayed in the Upgrade Applications Objects List view in Siebel Tools. Additional details on merge errors are located in the incremental repository merge log file:

```
SIEBEL_TOOLS_ROOT\log\IRM_merge0.txt
```

Each time you run the merge process, the name of the IRM_merge0.txt file is incremented, for example, to IRM_merge1.txt.

Causes of an Incomplete Merge Process

If your repository merge process terminates and is flagged as Incomplete, then navigate to the Screens menu in Siebel Tools, and choose the Application Upgrader menu item. The most common reasons for its failure are as follows:

- The number of errors (!!ERROR) exceeds the number that was predefined in Siebel Tools when the merge was started.
- The merge process has been terminated because of a local issue on the Siebel Tools workstation, such as a scheduled reboot.
- RDBMS errors caused the process to stop.
- Memory allotment issues occurred on the workstation on which Siebel Tools is installed.
- A network failure occurred.

If the repository merge is terminated and flagged as Incomplete, then restart the incremental repository merge. For more information, see [“Process of Meeting the Requirements for an Incremental Upgrade to Siebel Innovation Pack 2016, Using the Incremental Repository Merge” on page 187.](#)

Reviewing the Siebel Repository Merge Log Files

This task is a step in “Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x (SIA Repository)” on page 204. Use this task to review Siebel Repository merge log files.

Environments: Development environment only.

Platforms: Windows, UNIX, and IBM z/OS.

To determine whether the repository merge was successful

- 1 In Siebel Tools, from the Screens menu, choose Application Upgrader, and then Application Upgrade Object List.
- 2 In the Application Upgrades list, select the record of the merge.
- 3 Review the entry in the Status column as follows:
 - **Completed.** Indicates the merge completed without errors.
 - **Completed with Errors.** Indicates the merge contains errors.

If the Status column indicates Completed, then no further action is required. The merge was successful.

If the Status column indicates Completed with Errors, then you must review the errors to determine whether the merge was successful. To review the errors, complete the remaining steps in this task.

NOTE: These errors do not indicate an incomplete merge and rerunning the merge will not correct them.

- 4 In the Object Differences list, click Query.
- 5 In the Status field, enter the following:

```
ERROR: : *
```

- 6 Press Enter to run the query.

This step displays all the objects in which the merge process encountered errors.

- 7 Open the merge log file, IRM_merge0.txt.

It is located in the following directory:

```
SIEBEL_TOOLS_ROOT\log\IRM_merge0.txt
```

If there are multiple files, then open the one with the highest number in the file name, for example, IRM_merge1.txt.

- 8 To locate merge errors in the file, search for the !!ERROR string.
Informational messages are marked as !!INFO.

- 9 Use the objects displayed in the Object Differences list and the errors displayed in the log file to analyze the errors:
 - If all the errors are acceptable, then the merge is successful.

It is advisable, however, to consider the number of acceptable errors when determining whether to rerun the merge operation.
 - If the log file contains unacceptable errors, then the merge has failed.
- 10 If the merge contains unacceptable errors, then see 477269.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Troubleshooting Steps 19.

This document explains 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, then contact Oracle Global Customer Support.
- 11 Open the workflow merge log file:

```
SIEBEL_TOOLS_ROOT\log\IRM_merge0_ver.txt
```

If there are multiple files, then open the file with the highest number in the file name, for example, IRM_merge1_ver.txt. This log file is created by the workflow premerge and postmerge steps.
- 12 Review the log file. If the file contains errors, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Reviewing the Hierarchical Merge Report

This task is a step in ["Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x \(SIA Repository\)" on page 204](#). Use this task to review the hierarchical merge log files.

The hierarchical merge report is generated in HTML format for Application Hierarchy objects, which are Application, Screen, View, Business Component, Applet, Table, Business Service, Business Object and Workflow Process. This report allows you to review the tree of a particular object and see exactly what has been changed in the Application Hierarchy. If an object on which the currently selected object depends has changed, then you can drill down on that object and get a better idea of the testing effort involved.

The report is generated in the *SIEBSRVR_ROOT\log\upgprep_dev_*\output* directory and is named MergeReport.html.

When you run Incremental Repository Merge on Microsoft Windows, the HTML report is generated automatically once the incremental repository merge has completed. On UNIX, you must generate the hierarchical merge report manually, by running MergeReport.bat. Typically, you run this report after running incremental repository merge. For Upgrade, you also must generate the hierarchical merge report manually. MergeReport.bat is located in the *SIEBEL_TOOLS_ROOT\reppatch* directory.

You are prompted to specify values for the following parameters, which are required to generate the report. Where necessary, you must enclose parameter values in double-quotes.

- Tools Installation Folder

- ODBC Data Source Name
- Database Table Owner
- Database Table Owner Password
- Table Owner User Name
- Tablespace Name
- Index Space Name
- 16k Tablespace Name
- 32k Tablespace Name
- Is DB Unicode or Not
- Grantee Role (SSE_ROLE)
- Repository Name
- Full Path of Directory in Which to Generate HTML Report
- Full Path of Log File to Run Patch
- Database Platform (Oracle, MSSQL, DB2UDB, or DB2390)
- Merge ID for Report Generation. This value can be identified in Siebel Tools. Log in to Siebel Tools, navigate to Screens menu, then Application Upgrader, Application Upgrade Object List. Select a record and get the row ID from Help, About Record.
- Generate Report for Latest Merge Run. Enter Y to generate report. Enter N to display all available Merge Log ID.
- Retain Files for Debugging. Enter N to retain files for debugging. Otherwise, enter Y.

Usability

Different sections of the report present different types of information, as follows:

- The left section of the report, the navigation section, lists the objects that are affected by the merge process. Select an object in this section to view information about the object.
- The top-right section of the report lists the objects that are dependent on the object that is selected in the navigation section.
- The bottom-right section displays the attribute-level details of the object that is selected in the top-right section.

Navigation Section

This report limits the objects that are part of the application hierarchy. Nine tabs are displayed for each object type. When a tab is selected, it expands and displays the list of objects that are affected by the merge process. Two options provide searching capability for any particular object in the report:

- Global Search, provided on the top left, can be used to search any object in all objects of the application hierarchy.

- Local search limits the search functionality on a particular tab on which a search operation is triggered.

Dependent Object Section

The dependent object section, in the top right, displays the dependent objects for the object that is selected in the navigation section. The selected object name is displayed above the dependent object section. You can further navigate to each tab to see the list of objects.

Tabs are displayed only for objects that are dependent on the selected object. For example, if Applet has been modified and no Business Service is dependent on that Applet, then the Business Service tab will not be displayed in this section. Under each tab, if an object name has no background color, it means that no attribute level information is changed as part of the merge for this object.

For example, for the View tab, responsibility information is displayed for the selected view. When the View tab is selected and a View object is selected, then the responsibility information for this view is also displayed.

The bottom-right section of the report displays attribute-level information for the selected object. You can filter the resulting records based on Critical Conflicts, Non Critical Conflicts and All Changes.

Related Topics

[“Reviewing Siebel Repository Object Property Conflicts” on page 218](#)

Reviewing Siebel Repository Object Property Conflicts

This task is a step in [“Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x \(SIA Repository\)” on page 204](#). Use this task to review Siebel Repository object property conflicts.

Environments: Development environment only.

Platforms: Windows, UNIX, and IBM z/OS.

You can change how repository object property conflicts are resolved during the repository merge.

How Repository Object Property Conflicts Occur

NOTE: Use the Hierarchical Reports feature in Siebel Tools to learn what conflicts might have occurred, and to help identify areas to target for regression testing. For more information on this feature, see *Using Siebel Tools*.

You cannot resolve conflicts from the report, but you can use the report to identify where to target regression testing and what conflicts have occurred.

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 conflict 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 a conflict does not happen, the merge process transfers the changed value to the New Customer Repository.

The merge process resolves object property conflicts by referring to the setting of the object's Standard Win property. For the majority 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.

Reviewing Object Property Conflicts

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.** Puts 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, then the displayed value is changed 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. A check mark 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.
- **Conflict Resolution Type.** Displays whether the conflict is critical or non-critical. Only critical conflicts have to be resolved.

Related Topics

["Reviewing the Hierarchical Merge Report" on page 216](#)

Reviewing Log Files for All Upgrades

This task is a step in “Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x (SIA Repository)” on page 204. The repository merge must have been successful. For information about reviewing log files, see Chapter 11, “Reviewing the Siebel Upgrade Log Files.”

Reviewing Conflicts at the Attribute Level

This task is a step in “Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x (SIA Repository)” on page 204. Use this task to review conflicts at the attribute level.

Platforms: Windows, UNIX, and IBM z/OS.

For more information on review conflicts, see “Reviewing Siebel Repository Object Property Conflicts” on page 174.

To review conflicts at the attribute level

- 1 In Siebel Tools, from the Screens menu, choose Application Upgrader, and then Application Upgrade Attribute List.

By default, you can see only the critical conflicts in the Application Upgrades Attribute differences screen. You must review the critical conflicts. To view informational conflicts, right-click and select Show Non-Critical Conflicts. Noncritical conflicts include conflicts of attributes, such as Sequence number, Comments, Layout, and so on.

The screenshot displays the 'Application Upgrades - Attribute...' window. The top section, 'Application Upgrades', contains a table with the following data:

User	Operation	Status	Prior Standard Repository	Prior Customized Repository	New Standard Repository	New Customized Repository
SADMIN	Merge(NonICL)	Complete - 99 errors	Prior Siebel Repository	Prior Customer Repository	New Siebel Repository	New Customer Repository

The bottom section, 'Attribute Differences', contains a table with the following columns: Conflict Resolution Type, Object Name, Top Parent Name, Top Parent Type, and Object. A context menu is open over the 'Conflict Resolution Type' column, showing options: New Record, Delete Record, Copy Record, Undo Record, Columns Displayed..., Sort Order..., Show Critical Conflicts (checked), Show Non-Critical Conflicts, and Show All Conflicts.

Conflict Resolution Type	Object Name	Top Parent Name	Top Parent Type	Object
Critical	Search Specification	Product Compatibility - Default: 1	Workflow Process	WF Step 1/O Argument 'Search Specification' of WF
Critical	No	ISS Promotion Verify SubProcess: 1	Workflow Process	WF Step Branch 'No' of WF Step 'Violate Integrity
Critical	NewConn:42-4SKOQ	Product Compatibility - Default: 1	Workflow Process	WF Step Branch 'NewConn:42-4SKOQ' of WF Ste
Critical	NewConn:1-3MBL9	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3MBL9' of WF
Critical	NewConn:1-3MBL9	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3MBL9' of WF
Critical	NewConn:04-G6CEQ	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:04-G6CEQ' of v
	NewConn:04-G6CEQ	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:04-G6CEQ' of v
	NewConn:1-3J9ZM	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3J9ZM' of WF
	NewConn:1-3J9ZM	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3J9ZM' of WF
	NewConn:1-3JA19	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3JA19' of WF
	NewConn:1-3JA19	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3JA19' of WF
	NewConn:1-3MBLD	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3MBLD' of WI
	NewConn:1-3MBLD	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3MBLD' of WI
	NewConn:42-4ZY1P	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:42-4ZY1P' of W
	NewConn:42-4ZY1P	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:42-4ZY1P' of W
	NewConn:04-ECIY2	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:04-ECIY2' of W
	NewConn:04-ECIY2	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:04-ECIY2' of W
Critical	NewConn:1-3MBLL	Dynamic Pricing Procedure: 1	Workflow Process	WF Branch Connector 'NewConn:1-3MBLL' of WF

- 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 in 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 that changes the value of the object property in the merged repository. Avoid overriding conflicts for the following object properties. Review the following properties in the upgraded application before changing them: Left, right, top, height, and width.

Marking Conflict Resolution as Complete Using Siebel Tools

This task is a step in “[Process of Upgrading the Siebel Development Environment from Siebel CRM Version 8.1.1.x \(SIA Repository\)](#)” on page 204. Use this task to mark conflict resolution as Complete.

To mark conflict resolution as Complete using Siebel Tools

- 1 After you have resolved all conflicts, execute the following command from the `SIEBEL_TOOLS_ROOT\bin` directory on the command prompt:

```
si ebdev.exe /u username /p password /d "ServerDataSrc" /c  
"SIEBEL_TOOLS_ROOT\bin\lang_code\tools.cfg" /l primary_language /IRM  
UpgDel taMerge /i Packmode
```

- 2 After Siebel Tools is started, select the Conflict Resolution Completed check box.
- 3 Click Finish.

Clicking Finish creates an entry in S_INST_UPG_HIST table, indicating that the merge and conflict resolution steps have been completed. If this step is not executed, then the next phase of upgrading the custom database schema (upgphys) produces an error in the first step.

Upgrading a Custom Database Schema

To upgrade a custom database schema, perform the following task. For information on the following:

- **Parameters.** See “[Preparing to Run the Siebel Database Configuration Wizard](#)” on page 191.
- **Running the Database Configuration Wizard.** See one of the following, depending on your platform:
 - “[Running the Siebel Database Configuration Wizard on Windows](#)” on page 195

- [“Running the Siebel Database Configuration Wizard on UNIX” on page 196](#)

NOTE: Before you upgrade a custom database schema using the upgphys process, it is strongly recommended to back up the entire database. If this upgrade operation fails, then you can restore the database to its preupgrade state, and rerun the upgrade operation.

To upgrade a custom Siebel database schema

- Execute the Siebel Database Configuration Wizard by specifying the following parameter values:
 - Environment Type: Development
 - Upgrade Options: Upgrade Custom Database Schema (upgphys)

Table 31 on page 222 shows the different steps executed by the upgphys process. The Script or Input File column in the table lists the SQL file or input file that is executed in each step. The Description 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 31. Steps That the upgphys Process Executes

Step	Script or Input File	Description	Solution
1 Check whether the merge and conflict resolution steps are complete	verify_irm.sql	Verifies whether the merge and conflict resolution steps are complete.	<p>If this step stops because of an error, then the conflict resolution has not been completed. For information on resolving the conflict, see “Reviewing Siebel Repository Object Property Conflicts” on page 218.</p> <p>After you resolve the conflict, restart the Siebel Upgrade Wizard, and click No when asked if you want to continue. Then start the Siebel Upgrade Wizard again from the beginning.</p>
2 Drop temporary indexes created for merge performance	drop_temp_indexes_merge.sql	Removes all the temporary indexes created to improve the merge performance.	None

Table 31. Steps That the upgphys Process Executes

Step	Script or Input File	Description	Solution
3 Export schema definition	None	Exports the schema definition to the file <i>DBSRVR_ROOT/pl atform/ schema. ddl</i> . This file is used as an input to the production test and production environments.	None
4 Sync physical and logical tables	schema.ddl	Applies the schema.ddl and synchronizes the tables.	None
5 Sync physical and logical indexes	schema.ddl	Applies the schema.ddl and synchronizes the indexes.	None
6 Export repository to a file	None	Exports the New Customer Repository to: \$DbsrvrRoot/ \$DatabasePl atform/ custrep. dat This file is used as an input to the production test and production environments.	None
7 Import New Customer Repository as Siebel Repository	custrep.dat	Truncates the repository tables and imports the previously exported New Customer Repository as Siebel Repository.	None
8 Migrate custom manifest data from XML files to the database NOTE: This step applies only to upgrades with custom Siebel Open UI manifest files for Siebel CRM version 8.1.1.9 or 8.1.1.10, and version 8.2.2.2 or 8.2.2.3.	<ul style="list-style-type: none"> ■ custom_manifest.xml ■ manifest_extensions.map 	This step migrates Siebel Open UI custom manifest data from XML files to the database. For more information, see “About Migrating the XML Manifest Data for Siebel Open UI from Previous Releases to the New Manifest Data in the Database” on page 225.	None

Table 31. Steps That the upgphys Process Executes

Step	Script or Input File	Description	Solution
9 Migrate query search controls of list applets into the Siebel database	SrchCntrlMigration.jar	This step migrates query search controls of list applets from files into the Siebel database.	None
10 Create list import package for Siebel Marketing	mktg_listimp_pkg_spec.sql mktg_listimp_pkg_body.sql	(Oracle Database only) This step creates a package to support the Siebel Marketing list import functionality for Oracle Database 12c. For more information, see <i>Siebel Marketing Installation and Administration Guide</i> .	None
11 Verify repository after upgrade	<ul style="list-style-type: none"> ■ dbchck ■ dictutl ■ mlovupgd ■ SeedConflict.jar 	The repository is optionally verified after you complete the upgphys process. This step is executed if you selected the option Verify Repository After Upgrade in the Database Configuration Wizard. For more information, see “Verifying the Repository After Upgrade” on page 224 .	None
12 Upgrade history	store_history.sql	Stores the upgrade history in the S_INST_UPG_HIST table.	None

Verifying the Repository After Upgrade

As noted in [Step 11 on page 224](#) of [Table 31 on page 222](#), the repository is optionally verified after you complete the upgphys process. This step is executed if you selected the option Verify Repository After Upgrade during database configuration. The following are the verification steps executed:

- 1 **dbchck**. Provides a report of the schema differences between the physical database and the repository definition. Unacceptable errors can occur if data types are mismatched. Acceptable errors occur if a schema object (such as a table or an index) is intentionally external to the repository.
- 2 **dictutl**. Verifies that all dock objects and rule definitions are correct.

- 3 **mlovupgd**. Validates the repository for data inconsistencies. Errors are logged into the log file and upgrade wizard continues with the next steps.
- 4 **SeedConflict**. Generates the seed conflict report in HTML format. The report contains details of all the records with unique constraint violations.

About Migrating the XML Manifest Data for Siebel Open UI from Previous Releases to the New Manifest Data in the Database

Platforms: Windows, UNIX, and IBM z/OS.

In Siebel Open UI, before Siebel CRM version 8.1.1.11 or version 8.2.2.4 (Siebel Innovation Pack 2013), files to be downloaded for an applet were configured in one of the following ways:

- Through the Applet object properties `Presentation_Model` and `Physical_Renderer` in Siebel Tools.
- By adding entries under the `Presentation Model` and `Physical Renderer` sections in the `manifest_extensions.map` file in the `SI EBSRVR_ROOT\objects` folder on the Siebel Server.

The files were then configured against the key in the `custom_manifest.xml` file or in an application-specific custom manifest file.

Starting with Siebel CRM version 8.1.1.11 and version 8.2.2.4, Siebel Open UI replaces the XML manifest file with manifest data that it stores in the Siebel database, as noted in [“Upgrading a Custom Database Schema” on page 221](#). You cannot modify this predefined manifest data, but you can use the Manifest Administration screen in the client to configure the manifest data that your customizations require. Applets are mapped directly to files, and expressions are configured to determine under what conditions the applets use a given file.

The following example illustrates how data from the XML manifest file is migrated. The example uses the applet Activity HI Calendar Applet, which is used for calendar functionality in Siebel Open UI. It has separate renderings in mobile and desktop applications.

NOTE: The migration described in this topic applies only to upgrades with custom Siebel Open UI manifest files for Siebel CRM version 8.1.1.9 or 8.1.1.10, and version 8.2.2.2 or 8.2.2.3.

Figure 2 shows the UI Objects view with two Activity HI Calendar Applet UI object entries. One of these entries has a usage type of Renderer, while the other entry has a usage type of Presentation Mode. Two object expressions are shown in the Object Expression view: Mobile and Desktop, and in the Files view, the required JavaScript file is shown.

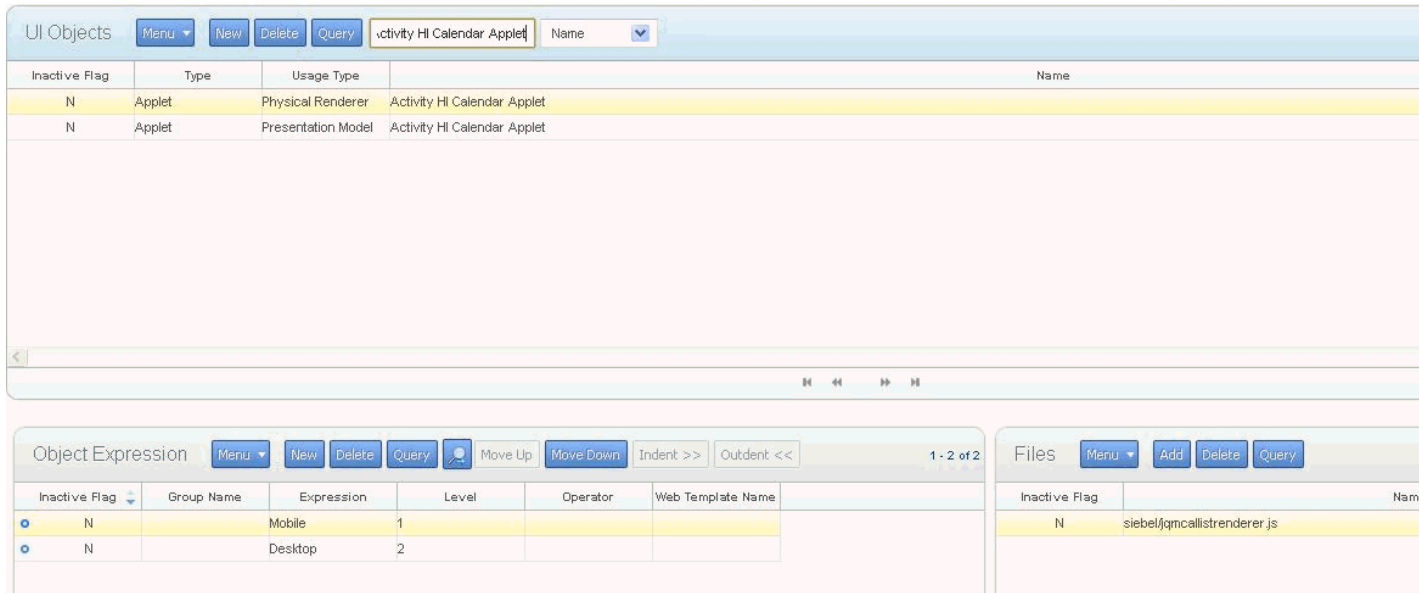


Figure 2. Activity HI Calendar Applet in UI Objects View, with Object Expressions and Files

As the following example from an earlier version shows, the two separate renderings were implemented in the XML manifest file for the applet using the CalPR key:

```
<PLATFORM Name="IsMobileApplication()">
  <KEY Name="SignViewPM"
  <KEY Name="SignViewPR"
  <KEY Name="AppletPR"
  <KEY Name="CalPR">
    <FILE Name>siebel/jqmcallsrenderer.js</FILE_NAME>
  </KEY>
  <KEY Name="GridPR"
  <KEY Name="RestOfUIPR"
</PLATFORM>
<PLATFORM Name="IsDesktopApplication()">
  <KEY Name="AppletPR">
  <KEY Name="ListPR">
  <KEY Name="GridPR">
  <KEY Name="TaskPanePR">
  <KEY Name="TreePR">
  <KEY Name="TileRenderer">
  <KEY Name="CalPR">
    <FILE Name>siebel/jqfullcalendarer.js</FILE_NAME>
  </KEY>
```

As shown in the example, separate files for mobile and desktop platforms are configured using the CalPR key. However, for the new manifest data in the Siebel database, there are two required configured entries: Mobile and Desktop. Both expressions correspond to a single user interface object, and they are evaluated sequentially by their level number. If the setting is True for either entry, then the respective file is downloaded, and further evaluation does not take place.

However, if a record is an Application type user interface object, with a usage type of Common, then all expressions set to True are considered, not just the first in Object Expression list. All True expressions are then aggregated and downloaded to the browser. For information on modifying manifest data, see *Configuring Siebel Open UI*.

Manifest File Migration Rules

The migration rules for this example are as follows. For each applet mapped to a manifest key in either of the following ways, an entry is made in the UI Objects applet.

- By using the Presentation_Model and Physical_Renderer user properties under the Applet object in Siebel Tools.
- By adding entries under the [Presentation Model] and [Physical Renderer] sections in the manifest_extensions.map file, in the *SI EBSRVR_ROOT\obj ects* folder on the Siebel Server.

Each time the manifest key is used in the legacy XML file, an entry is made in the new Manifest screen, and a record is also added to the Object Expression and Files views, using the rules shown in [Table 32](#).

Table 32. Key Usage Rules

Use of Keys in Legacy XML Files	Record Created in the Object Expression and Files Applets
Under a PLATFORM subsection within the PLATFORM_KEY_SPECIFIC section in the custom_manifest.xml	The record in the Object Expression applet with an expression value that evaluates to True only for the specified platform. The record also exists in the Files applet for any files associated with the record.
Under the KEY_COMMON section in the custom_manifest.xml	The record in the Object Expression applet with no entry in the Expression field. The record also exists the Files applet for any files associated with the record.

Table 32. Key Usage Rules

Use of Keys in Legacy XML Files	Record Created in the Object Expression and Files Applets
Under a PLATFORM section within the PLATFORM_KEY_SPECIFIC section in a custom Application Manifest XML file	<p>The following records are created in the Object Expression applet:</p> <ul style="list-style-type: none"> ■ A record for Group Expression ■ A record with an Expression value that evaluates to True only for the given platform ■ A record with an Expression value that evaluates to True only for the given application <p>The records in the Files applet are associated with the Group Expression.</p>
Under the KEY_COMMON section in a custom Application Manifest XML file	The record in the Object Expression applet with an Expression value that evaluates to True only for the given application and a record in the Files applet for the file associated with the record.

Default Configuration for Generic Applet Types and Configuration of Non-Applet User Interface Constituents

This topic applies to the following:

- **Applets that use the default configuration for generic applet types.** Such as List, Tree, and so on. In other words, applets that have not been mapped to any Manifest Key specific to an individual applet.
- **Other user interface elements.** Such as navigation user interface elements other than applets, whose keys are defined within the implementation itself, in other words, not through user properties or through the mapping file.
- **File downloads that occur during the application startup.**

For generic applet types, the UI Object applet is populated with records with well known values for the Name field. [Table 33](#) lists these values.

Table 33. Default Configuration for Generic Applet Types and Configuration of Non-Applet UI Constituents

Key in the Legacy Manifest File	UI Object Type	UI Object Usage Type	Generic UI Object Name	Description
AppletPM	Applet	Presentation Model	DEFAULT FORM APPLET	Key from the default Form Applet Presentation Model.
AppletPR	Applet	Physical Renderer	DEFAULT FORM APPLET	Key for the default Form Applet Physical Renderer.
ListPM	Applet	Presentation Model	DEFAULT LIST APPLET	Key for the default List Applet Presentation Model.
ListPR	Applet	Physical Renderer	DEFAULT LIST APPLET	Key for the default List Applet Physical Renderer.
TreePM	Applet	Presentation Model	DEFAULT TREE APPLET	Key for the default Tree Applet Presentation Model.
TreePR	Applet	Physical Renderer	DEFAULT TREE APPLET	Key for the default Tree Applet Physical Renderer.
ScreenPM	Navigation	Presentation Model	NAVIGATION CONTROL	Key for the default Navigation Control Presentation Model.
RestOfUIPM	Application	Common	PLATFORM INDEPENDENT	Key for Presentation Model implementation of parts of the user interface other than those with specific keys for their default implementation, such as Menu, Toolbar, and so on.
RestOfUIPR	Application	Common	PLATFORM DEPENDENT	Key for Physical Renderer implementation of parts of user interface other than those with specific keys for their default implementation, such as Menu, Toolbar, and so on.
NAVIGATION_TREE	Navigation	Physical Renderer	NAVIGATION TREE	Key for the default tree Navigation Control Physical Renderer.
NAVIGATION_TAB	Navigation	Physical Renderer	NAVIGATION TAB	Key for the default tab Navigation Control Physical Renderer.

Generic Applet Types from the XML Manifest File

Platforms: Windows, UNIX, and IBM z/OS.

This example of a generic Mobile applet shown in [Figure 3](#) uses the AppletPR key in the XML manifest file. This key has two entries in the Object Expression view: Mobile and Desktop. The Files view contains the JavaScript file that the manifest file maps to display the applet.

The screenshot displays three panels in the Siebel Open UI configuration tool:

- UI Objects:** A table listing various applet types. The second row is highlighted and labeled "UI Objects List Applet".
- Object Expression:** A table showing two entries for the selected applet, one for "Mobile" (Level 1) and one for "Desktop" (Level 2).
- Files:** A table showing a single entry for the JavaScript file "siebel/jgformrenderer.js".

Inactive Flag	Type	Usage Type	Name
N	Applet	Physical Renderer	DEFAULT FORM APPLET
N	Applet	Presentation Model	DEFAULT FORM APPLET
N	Applet	Physical Renderer	DEFAULT LIST APPLET
N	Applet	Presentation Model	DEFAULT LIST APPLET
N	Applet	Physical Renderer	DEFAULT TREE APPLET
N	Applet	Presentation Model	DEFAULT TREE APPLET

Inactive Flag	Group Name	Expression	Level	Operator	Web Template Name
N		Mobile	1		
N		Desktop	2		

Inactive Flag	Name
N	siebel/jgformrenderer.js

Figure 3. PLATFORM Name Is Set to "IsMobileApplication()"

Figure 4 shows the entries for the navigation controls, such as ScreenPM, in the UI Objects screen.

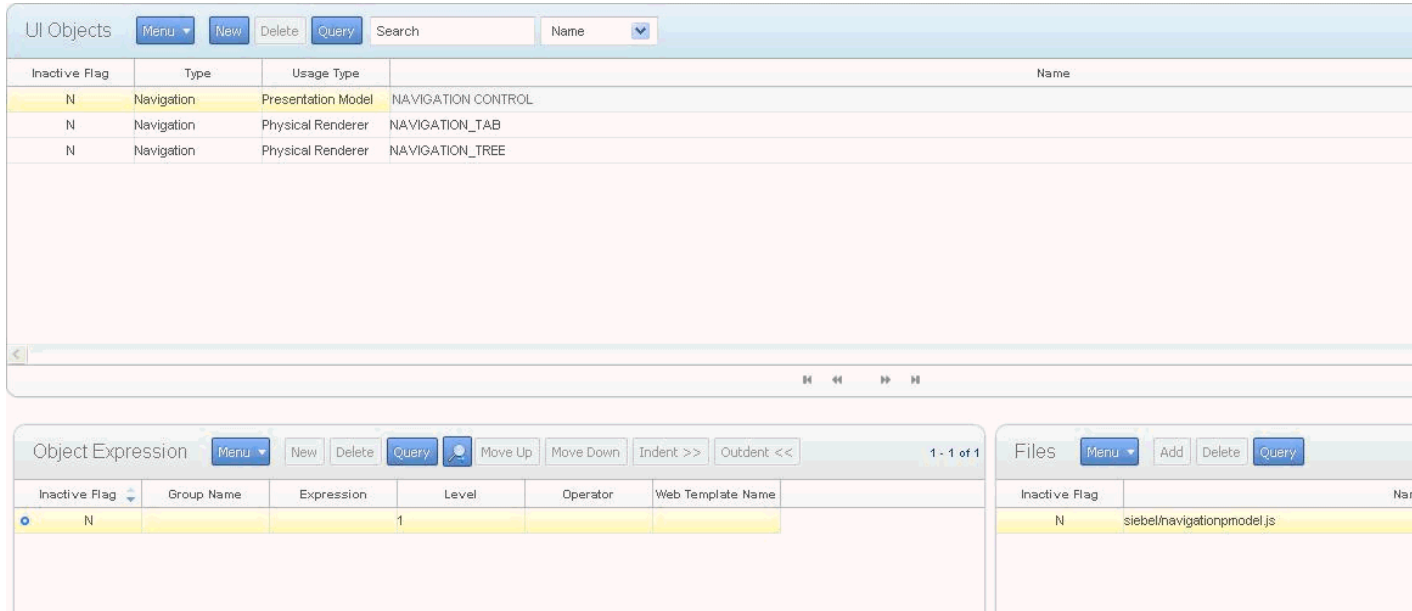


Figure 4. Navigation Control

Figure 5 shows the entries for NAVIGATION_TREE.

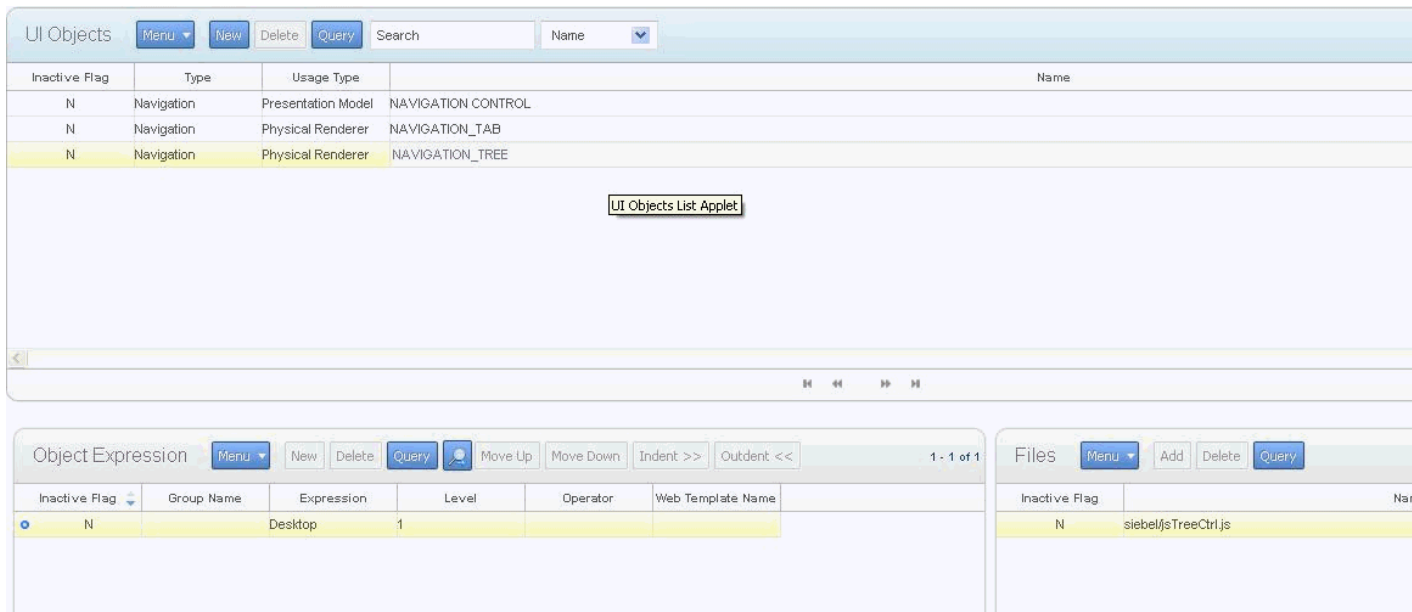


Figure 5. Navigation Tree

Figure 6 shows the entries for NAVIGATION_TAB.

The screenshot displays three tables from the Siebel Open UI configuration tool:

- UI Objects Table:**

Inactive Flag	Type	Usage Type	Name
N	Navigation	Presentation Model	NAVIGATION_CONTROL
N	Navigation	Physical Renderer	NAVIGATION_TAB
N	Navigation	Physical Renderer	NAVIGATION_TREE
- Object Expression Table:**

Inactive Flag	Group Name	Expression	Level	Operator	Web Template Name
N		Desktop	1		
- Files Table:**

Inactive Flag	Name
N	siebel/accnavigationphyrender.js

Figure 6. Navigation Tab

The files specified in the legacy manifest keys, RestOfUIPM and RestOfUIPR, are now located within the entries for UI Objects named PLATFORM_INDEPENDENT and PLATFORM_DEPENDENT. [Figure 7](#) shows the migrated legacy manifest keys.

The screenshot displays three panels from the Siebel Open UI interface:

- UI Objects Panel:** Shows a table with columns: Inactive Flag, Type, Usage Type, and Name. Two rows are visible:

Inactive Flag	Type	Usage Type	Name
N	Application	Common	PLATFORM DEPENDENT
N	Application	Common	PLATFORM INDEPENDENT
- Object Expression Panel:** Shows a table with columns: Inactive Flag, Group Name, Expression, Level, Operator, and Web Template Name. It displays a list of expressions:

Inactive Flag	Group Name	Expression	Level	Operator	Web Template Name
N	Siebel Service Portal		1	AND	
N	Siebel ePharma Mo...		2	AND	
N	Siebel ePharma Mo...		3	AND	
N	Offline Phone		4	AND	
N	Offline Tablet		5	AND	
Y	Desktop		6		
N	Desktop Phase 1 - ...		6		
N	Desktop Phase 2 - ...		7		
N	Desktop Phase 3 - ...		8		
N	Desktop Phase 4		9		
- Files Panel:** Shows a table with columns: Inactive Flag and Name. It lists various JavaScript files:

Inactive Flag	Name
N	3rdParty/DynaTreeNav/javascript/dynatree/src/quer...
N	3rdParty/candy-chat-ext.js
N	3rdParty/candy-chat/candy_bundle.js
N	3rdParty/candy-chat/libs/libs_bundle.js
N	3rdParty/ckeditor/adapters/jquery.js
N	3rdParty/ckeditor/ckeditor.js
N	3rdParty/datepicker-ext.js
N	3rdParty/jquery-ui-timepicker-addon.js
N	3rdParty/jquery.calculator/current/jquery.calculator.js
N	3rdParty/jquery.narad.min.js

Figure 7. Legacy Manifest Keys

Files configured against a number of sections in the legacy manifest XML file are now part of these two UI Objects:

- PLATFORM DEPENDENT
- PLATFORM INDEPENDENT

Table 34 shows the mapping of the legacy XML file.

Table 34. Legacy XML Mapping

Legacy XML Section	Legacy Manifest Specific Subsection	Legacy Manifest Key	UI Object Name	UI Object Expression
COMMON	Not applicable	Not applicable	PLATFORM INDEPENDENT	Not applicable
PLATFORM_COMMON	PLATFORM Name="IsDesktopApplication()"	Not applicable	PLATFORM DEPENDENT	Desktop
PLATFORM_KEY_SPECIFIC	PLATFORM Name="IsDesktopApplication()"	Not applicable	PLATFORM DEPENDENT	Desktop
KEY_COMMON	Not applicable	RestOfUIPM	PLATFORM INDEPENDENT	Not applicable
PLATFORM_KEY_SPECIFIC	PLATFORM Name="IsDesktopApplication()"	RestOfUIPR	PLATFORM DEPENDENT	Desktop

NOTE: When processing Object Expressions Records to choose files for a given UI Object, the first Expression that evaluates to True is selected, and the file associated with it is downloaded to the browser. However, there is one exception. With the UI Object records of the Application type and Common usage type, all expressions in the Object Expression list are considered, and then they are aggregated and downloaded to the browser.

Differences Between Customer Manifest Data and Oracle Manifest Data

All manifest data that replaces the previous configuration in the core_manifest.xml file is seeded in the database that is packaged with the release of Siebel CRM version. You cannot change the Oracle manifest seed data. For any UI Objects record that has been seeded in the database by Oracle, the record itself and all child Object Expression records are read-only.

If you want to customize a given UI Object, then a separate top-level UI Objects record must be created, and the required Object Expression records must be created under it. For example, if a UI Object record with a Name value of Contact List Applet is seeded in the manifest database for the Usage Type Physical Renderer, and if you want to override the configuration with a different implementation of the Physical Renderer, then you must create a new UI Objects record with a Name value of Contact List Applet and a Usage Type value of Physical Renderer, and the Object Expression record you want to create and corresponding Files record must be entered.

NOTE: If you add a UI Objects record with the same Name, Type, and Usage Type values to a UI object that is already seeded in the Manifest database, then your newly created record is given precedence by the Manifest Runtime engine, and is used when evaluating true Object Expression values when choosing a File record to download. There is one exception, however. For UI Objects records with a Type value of Application and a Usage Type value of Common, both the Oracle seeded record and your new record are considered. The sum total of all the chosen Files for both the seeded record and your new record is aggregated and downloaded to the browser.

File Selection Rules

Table 35 shows file selection rules.

Table 35. File Selection Rules

UI Object Record Values	Oracle (Siebel?) Record Present	User Created Record	File Selection Rule
Type = 'Application' and Usage Type = 'Common'	Yes	No	All files against all winning expressions under the Oracle supplied UI Objects record are aggregated and downloaded to the browser.
Type = 'Application' and Usage Type = 'Common'	Yes	Yes	All files against all winning expressions under both the Oracle supplied and user entered UI Objects record are aggregated and downloaded to the browser.
Usage Type != 'Common'	Yes	No	The file against the first winning expressions under the Oracle supplied UI Objects record is downloaded to the browser.
Usage Type != 'Common'	Yes	Yes	The file against the first winning expressions under the user entered UI Objects record is downloaded to the browser.

Checking the Post-upgphys Verification Process

After the upgphys upgrade is completed, check the oui_manifest.log file located in *SIEBEL_ROOT\si_ebsrvr\upgphys_dev_versionnumber\output* to verify whether the upgrade was successful.

Checking for Database Errors

Use the following method to check for database errors.

To check for database errors

- If you encounter database errors during the running of the upgphys upgrade, check the manifest_sqllog.txt file.

Checking for Application Errors

Use the following method to check for application errors.

To check for application errors

- If you encounter application errors during the running of the upgphys upgrade, check the oui_manifest.txt file.

About Migrating Siebel Web Template Data

Platforms: Windows, UNIX, and IBM z/OS.

As noted in [“Upgrading the Siebel Database Schema” on page 205](#), in Siebel Innovation Pack 2015 and later, a script is executed that migrates data from Siebel Web template files into table-based content in the Siebel database.

The migrated Siebel Web template data is used by Siebel CRM Composer, which is available for preview purposes in the current release. For more information about Siebel CRM Composer and Siebel Web template migration, see *Siebel CRM Composer Developer Preview Guide*.

NOTE: Your existing deployment and usage of Siebel Web templates are unaffected by this migration.

Siebel Web template data is migrated only where a corresponding Siebel Web template (SWT) file is located in the *SI EBSRVR_ROOT/webtempl /custom* directory. Before the migration, copy all Siebel Web template files located in *SI EBSRVR_ROOT/webtempl* into *SI EBSRVR_ROOT/webtempl /custom* in order to migrate data for all of the Siebel Web templates.

For each Siebel Web template, the migration script for the file in *SI EBSRVR_ROOT/webtempl /custom*. When the migration script finds the Siebel Web template file, it parses the file and processes its data, and then moves on to the next Siebel Web template. If the Web template file does not exist in the required directory, then the migration script lists the file in the List of Missing Files section of the swt_mig.log file.

For Incremental Repository Merge, Siebel Web template migration occurs during the upprep operation. For full upgrades, this migration occurs during the upgphys operation. These operations might take longer to perform than in previous releases, in part because of this migration step.

Optionally, you can run the Siebel Web template migration script manually after upgrading, as described in *Siebel CRM Composer Developer Preview Guide*. If you do so, then the migration script overwrites the previously generated table-based Siebel Web template data.

For a new Siebel database installation, the Siebel Repository now includes table-based data that was migrated from the default Siebel Web template files.

For general information about upgrade planning for Siebel Web templates, see [“Upgrade Planning for Siebel Web Template Files and Style Sheets” on page 73](#).

Related Books

Siebel CRM Composer Developer Preview Guide

Siebel Installation Guide for the operating system you are using

Process of Regenerating the Siebel Repository Definition Files

The tasks in this process enable you to regenerate the Siebel Repository definition files.

Environments: Development environment only.

If you have modified repository objects after the development environment upgrade (upgphys) and before upgrading the production test environment, then you must regenerate the schema.ddl and custrep.dat files. These files were created during the development environment upgrade (upgphys):

- **Schema.ddl.** This file contains the logical definition of the Siebel database.
- **Custrep.dat.** This file contains the definition of the 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, then you must regenerate the files.

Perform the following tasks to regenerate the Siebel Repository definition files:

- 1 [“Regenerating the schema.ddl File” on page 237](#)
- 2 [“Regenerating the custrep.dat File” on page 238](#)
- 3 [“Editing the Siebel Tools Configuration File After the Development Environment Merge” on page 239](#)

Regenerating the schema.ddl File

This task is a step in [“Process of Regenerating the Siebel Repository Definition Files” on page 237](#). Use this procedure to regenerate the schema.ddl file.

To regenerate the schema.ddl file

- 1 On the Siebel Server where the Siebel Database Server files are installed, navigate to the following location:

- **Windows.** *SIEBEL_ROOT\bin*
- **UNIX.** *\$SIEBEL_ROOT/bin*

- 2 Run the following command:

```
ddl di ct /u DatabaseOwner /p Password /c "ODBCDataSource" /d TableOwner /f  
DBSRVR_ROOT\DatabasePlatform\schema.ddl /e y /a y /l SiebelLogDir\sch_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 quotation marks.
 - *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 the Windows path syntax. On UNIX systems, use the UNIX path syntax.
 - *SiebelLogdir* is the path to the directory where you want the output log placed; that is, the log output directory. The example shows the Windows path syntax. On UNIX systems, use the UNIX path syntax.
- 3 After the command completes, review the output log files for errors. If the log indicates there are errors, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Regenerating the custrep.dat File

This task is a step in "Process of Regenerating the Siebel Repository Definition Files" on page 237. Use this procedure to regenerate the custrep.dat file.

To regenerate the custrep.dat file

- 1 On the Siebel Server where the Siebel Database Server files are installed, navigate to the following location:

Windows: *SIEBEL_ROOT\bin*

UNIX: *\$SIEBEL_ROOT/bin*

- 2 Run the following command:

```
repi mexp /a e /u DatabaseOwner /p Password /c "ODBCDataSource" /d TableOwner  
/r "Siebel Repository" /f DBSRVR_ROOT\DatabasePlatform\custrep.dat  
/l SiebelLogDir\exprep.log
```

where:

- *DatabaseOwner* is the Siebel database administrator account name.
 - *Password* is the Siebel database administrator account password.
 - *ODBCDataSource* is the ODBC name for connecting to the database. Enclose the name in quotation marks.
 - *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 has finished, review the output log files for errors. If the log indicates there are errors, then create a service request (SR) on My Oracle Support, or contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Editing the Siebel Tools Configuration File After the Development Environment Merge

This task is a step in "Process of Regenerating the Siebel Repository Definition Files" on page 237. Use this task to edit the Siebel Tools configuration file after successfully performing your Development environment repository merge.

CAUTION: You must set the prefix to back to X_ from changed value of SBL_ to enable you, after the merge, to create your own custom symbolic strings.

To change the SymStrPrefix value in tools.cfg file

- 1 Navigate to *SIEBEL_TOOLS_ROOT\bin\lang_code*, and open the tools.cfg file.
- 2 Make the following changes in the tools.cfg file:
 - a Change the SymStrPrefix value from SBL_ back to the default value of X_.
 - b Make sure the Set EnableToolsConstrain value is set to the default value of False. If it is not, then set the value to False.
 - c Change the DockRepositoryName value from Prior Customer Repository to Siebel Repository.

- 3 Save the tools.cfg file.

Performing a Production Test or Production Environment Migration from Siebel CRM Version 8.1.1.x (SIA Repository)

Environments: Production test and production.

To upgrade your production environment use the following task. For information on Siebel Database Configuration Wizard parameters, see [“Preparing to Run the Siebel Database Configuration Wizard” on page 191](#). For information on running the Database Configuration Wizard, see [“Running the Siebel Database Configuration Wizard on Windows” on page 195](#) or [“Running the Siebel Database Configuration Wizard on UNIX” on page 196](#), depending on your platform.

Requirements: You must copy the schema.ddl and the custrep.dat files to your production test environment and production environment to conduct upgrades in each of these environments. For more information, see [“Process of Regenerating the Siebel Repository Definition Files” on page 237](#).

To perform a production upgrade

- 1 In the Siebel Database Configuration Wizard, select the following options:
 - Environment Type: Production
 - Upgrade Options: Upgrade Custom Database Schema (upgrep and upgphys)
- 2 Start the Siebel Upgrade Wizard.

For information on running the Siebel Upgrade Wizard, see [“Preparing to Start the Siebel Upgrade Wizard” on page 197](#).

The Siebel Upgrade Wizard reads the steps from the driver file that relate to your Siebel CRM version. The Siebel Upgrade Wizard performs the commands listed in those steps.

14 Siebel Postmerge Development Tasks

This chapter provides guidelines for developers following a successful Siebel Repository merge. This chapter includes the following topics:

- [Reviewing Objects Deleted from the Siebel Repository on page 241](#)
- [Reviewing Obsolete Objects in the Siebel Repository on page 242](#)
- [Upgrading to the Siebel Symbolic String Model on page 243](#)
- [Dropping IBM DB2 8-KB Tablespace and Buffers After a Siebel Merge on page 243](#)
- [Updating Siebel Enterprise Application Integration \(EAI\) on page 244](#)

Reviewing Objects Deleted from the Siebel Repository

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

If an object you have deleted from the Prior Customer Repository exists in the New Customer Repository, then the repository merge does not delete the object from the New Customer Repository. After the merge, you must review these objects and verify that they do not adversely affect the operation of the application.

Requirements: The repository merge must have been successful. See [“Reviewing the Siebel Repository Merge Log Files” on page 171](#).

To generate a list of deleted objects

- 1 In Siebel Tools, from the Screens menu, choose Application Upgrader, and then Application Upgrade Object List.
- 2 In the Application Upgrades list, select the record of the successful merge.
- 3 Click Query.
- 4 Enter your query criteria in the Object Differences list:
 - Click in the In Prior Standard field so that a check mark appears.
 - Click in the Added to New Customized field so that a check mark appears.
 - Click in the In Prior Customized field so that a check mark appears. Then click in it again so that no check mark appears.

- 5 Press Enter to run the query.

Deleted objects appear in the Object Differences list. You can filter the objects displayed by using the Top Parent Type and Object Type fields.

- 6 Review the list carefully to determine that deleted objects that have been restored to the merged repository will not have an adverse effect on upgraded applications.

Related Topics

[“About the Siebel Repository Merge” on page 51](#)

Reviewing Obsolete Objects in the Siebel Repository

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

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.

Requirement: The repository merge must have been successful. See [“Reviewing the Siebel Repository Merge Log Files” on page 171](#).

To generate a list of obsolete objects

- 1 In Siebel Tools, from the Screens menu, choose Application Upgrader, and then Application Upgrade Object List.
- 2 In the Application Upgrades list, select the record of the successful merge.
- 3 Click Query.
- 4 Enter your query criteria in the Object Differences list:
 - a Click in the In Prior Standard field so that a check mark appears.
 - b Click in the In New Standard field so that a check mark appears. Then click in the field again so that no check mark appears.
 - c Click in the In Prior Customized field so that a check mark appears.
 - d Click in the Attribute field so that a check mark appears. Then click in the field again so that no check mark appears.
- 5 Press Enter to run the query.

All obsolete objects appear in the Object Differences list. You can filter the objects displayed by using the Top Parent Type and Object Type fields.

- 6 Review the list carefully to determine that obsolete objects that have been deleted will not have an adverse effect on upgraded applications.

Related Topics

[“About the Siebel Repository Merge” on page 51](#)

Upgrading to the Siebel Symbolic String Model

Environments: Development environment only.

The symbolic string model is object-oriented. As of Siebel CRM version 7.8.2, 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 user interface.

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, then it is recommended that you do so before you run the string conversion or consolidation process.

Instructions for converting or consolidating to the symbolic strings model are found in *Using Siebel Tools*.

Related Topics

[“About the Siebel Repository Merge” on page 51](#)

[“Upgrade Planning for Siebel String Translation” on page 80](#)

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

Environments: Development environment only.

Databases: IBM DB2 only.

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)

Environments: Development environment only.

Platforms: Windows and UNIX only.

If you use Enterprise Application Integration (EAI), then 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, then deactivate the user keys or fields that cause the error.
- 4 If you receive the error "List Of" in the XML Parent Element, then manually remove the value "List Of" from the XML Parent Element.

15 Postupgrade Tasks for the Siebel Database

This chapter provides tasks to be completed within the Siebel database, and the Siebel File System following a successful upgrade and repository merge. This chapter includes the following topics:

- [Reapplying Schema Customizations in the Siebel Database on page 245](#)
- [Migrating Address Data After a Direct SEA to SIA Upgrade on page 246](#)
- [Validating Dock Objects and Rule Definitions in the Siebel Database on page 247](#)
- [Verifying an Upgraded Oracle Database After a Siebel Upgrade on page 249](#)
- [Setting Oracle Database Parameters After a Siebel Upgrade on page 250](#)

Reapplying Schema Customizations in the Siebel Database

Environments: Development environment only.

In the current release, tables are obsolete or have been replaced by new tables. If you added extension columns or foreign key (FK) columns to tables that are obsolete in the current release, then you must reapply these changes to the new tables.

Reviewing Obsolete Tables

The upgrade process generates a report that you can review for information about tables that are either obsolete. This report, `xtndobstbl.txt`, lists the following:

- Custom columns in obsolete tables
- Custom foreign key columns pointing to obsolete tables
- Siebel Enterprise Integration Manager 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 can continue to use them. These tables are supported as is (for instance, with docking or Siebel Enterprise Integration Manager).
- **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 `xtndobstbl.txt`, then no action is required.

If this file lists any tables, then reapply the custom extensions and foreign key columns to those tables in the current release using Siebel Tools. See *Configuring Siebel Business Applications*.

Migrating Address Data After a Direct SEA to SIA Upgrade

Upgrades from: SEA repository to SIA repository.

Platforms: Windows and UNIX only.

See [Table 36](#) for a list of schema differences between the SEA and SIA repository along with recommended data migration approaches.

Schema Differences Between SEA and SIA Repositories

[Table 36](#) displays schema differences between the SEA repository and the SIA repository.

Table 36. Schema Differences Between SEA and SIA Repositories

SEA Repository	SIA Repository	SEA to SIA Migration Approach
Account Addresses are stored in the S_ADDR_ORG table. Other addresses are stored in the S_ADDR_PER table.	The S_ADDR_ORG table is obsolete. Address data is stored in the S_ADDR_PER table.	Data is converted automatically through upgrade scripts.
Organization Relationships are stored in the S_ORG_REL table. Other party relationships are stored in the S_PARTY_REL table.	All party relationships, including organizations, are stored in S_PARTY_REL. The S_ORG_REL table is obsolete.	Manual data migration required. For instructions see 476479.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 312.

Table 36. Schema Differences Between SEA and SIA Repositories

SEA Repository	SIA Repository	SEA to SIA Migration Approach
Contact relationships are stored in the S_CONTACT_REL table. Other party relationships are stored in the S_PARTY_REL table.	All party relationships, including organizations, are stored in the S_PARTY_REL table. The S_CONTACT_REL table is obsolete.	Manual data migration required. For instructions see 476479.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 312.
Household relationships are stored in the S_PER_ORG_UNIT table.	Household relationships are stored in the S_PARTY_PER table.	Manual data migration required. For instructions see 476479.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 312.

Validating Dock Objects and Rule Definitions in the Siebel Database

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

Changes to visibility rules and dock objects require the assistance of Oracle Advanced Customer Services.

If you deploy Siebel Business Applications to mobile users with local databases, then you can run the DICTUTL utility to verify that all dock objects and rule definitions are correct. Dock objects allow mobile users to synchronize their local databases with the Siebel Server. Rules determine which data users synchronize. For more information about dock objects and rules, see *Siebel Tools Online Help* and *Siebel Remote and Replication Manager Administration Guide*.

To verify that all dock object and rule definitions are correct

- 1 Navigate to the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Type the following command using the parameters specified in [Table 37](#):

```
dictutl /C ODBC_DATASOURCE /U USERNAME /P PASSWORD /D TABLEOWNER /N
"REPOSITORY_NAME" /A y 2> logfile.log
```

Table 37. 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, then break the file into two files and examine each separately.
 - b Query for the word error.
 - c If you locate errors, then write down the exact error text.
 - d Query for the word syntax.
 - e If you locate syntax errors, then write down the exact error text.
 - f To determine whether the errors must be resolved, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

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

- Errors: 0
- Syntax OK

Related Topics

["Preserving Siebel Dock Objects and Visibility Rules" on page 85](#)

Verifying an Upgraded Oracle Database After a Siebel Upgrade

Environments: Development, production test, production.

Databases: Oracle only.

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, then you must run the Siebel `ddlmp` utility to verify the schema layout of the upgraded database.

NOTE: The `upgphys` process might recreate descending (DESC) indexes as ascending (ASC) indexes. To recreate descending indexes, use the `/9` option while running `ddlmp`.

This also converts the Siebel database to character-based length for char and varchar data. For additional information about the following procedure, see 477966.1 (Article ID) My Oracle Support.

To verify the upgraded Oracle Database

- 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 /I SiebelLogDir/ddlctl_veri fy_RDBMS.log
```

where:

- `TableOwner` is the Siebel table owner name, also called the schema qualifier name.
 - `TablePassword` is the Siebel table owner password.
 - `ODBCDataSource` is the ODBC name for connecting to the database. Enclose the name in quotes.
 - `DBSRVR_ROOT` is the absolute path to the Siebel Database Server installation directory.
 - `DatabasePlatform` is the Siebel Database Server directory for the Oracle Database.
 - `Tablespace` is the Oracle tablespace name for the Siebel database.
 - `IndexSpace` is the Oracle index space name for Siebel database.
 - `SiebelLogDir` is the path to the directory where you want the output log placed (log output directory).
- 3 After the command completes, review the output log files for errors. If the log indicates there are errors, then create a service request (SR) on My Oracle Support. Alternatively, you can phone Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers remain the same and are listed on My Oracle Support.

Related Topics

[“About Upgrading Your RDBMS in the Siebel Environment” on page 35](#)

Setting Oracle Database Parameters After a Siebel Upgrade

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 781927.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 582.
- For a full list of recommended settings for your postupgrade production environment, see the chapter on configuring the RDBMS in *Siebel Installation Guide* for the operating system you are using.

16 Reviewing the Siebel User Interface

This chapter provides information on reviewing your Siebel user interface following a repository merge. This chapter includes the following topics:

- [Troubleshooting Postmerge Siebel User Interface Problems on page 251](#)
- [Reviewing Siebel Grid-Based Applets on page 252](#)
- [Reviewing Siebel User Interface Navigation on page 254](#)
- [Reviewing Siebel Multi-Value Group Shuttle Applets on page 255](#)
- [Revising Siebel UI Rich Text Controls on page 258](#)
- [Reviewing New Siebel UI Aggregate Categories on page 259](#)
- [Revising Siebel Visibility Filters to Display Correctly on page 259](#)
- [Assigning a Category and Type to Siebel Chart Views on page 260](#)
- [Assigning a Category and Type to Siebel Explorer Views on page 261](#)
- [Setting Up Navigation to Inaccessible Siebel Detail Views on page 262](#)
- [Eliminating Obsolete Siebel UI Fields on page 263](#)
- [Reviewing Siebel UI Objects Affected by Incorporate Custom Layout on page 263](#)
- [Reviewing Required Fields in the Siebel User Interface on page 264](#)
- [Assigning an Item Identifier to Siebel Web Template Items on page 265](#)

Troubleshooting Postmerge Siebel User Interface Problems

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Basic Troubleshooting Guidelines

If views or screens do not display, then 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.
- 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, then 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.

About the Postmerge Utilities Log

After the repository merge, you must run the postmerge utilities. These utilities do the following:

- Validate user interface components to verify they were migrated correctly to the new repository.
- Modify UI objects to implement new user interface features. For example, they modify form applets and multi-value group applets.
- 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 might be incorrectly configured, so must be reviewed.
- **ERROR.** These messages indicate that a problem has been found that must be corrected.

The postmerge utilities log is located here:

```
SIEBEL_TOOLS_ROOT\reppatch\log\reputility.log
```

For more information on troubleshooting user interface problems, see 477269.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Troubleshooting Steps 19.

Related Topics

[“About the Siebel Postmerge Utilities” on page 59](#)

[“About the Siebel Repository Merge” on page 51](#)

[Appendix C, “Siebel Incorporate Custom Layout Upgrade Option”](#)

Reviewing Siebel Grid-Based Applets

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Start Invalid Web Template Item Mapping Clean-up.”

If a Prior Customer Repository applet that will be converted to grid-based layout has been customized by adding new fields or controls, then 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, then ICL converts many form applets to grid-based layout. The postmerge utilities then deactivate the grid-based layout and activate the Prior Customer Repository, flow-based form of the applet.

The reputility.log file lists applets that were converted by the postmerge utilities. Review the applets listed in the section of the log referenced above and revise layouts as needed.

The log is located here:

```
SIEBEL_TOOLS_ROOT\reppatch\log\reputility.log
```

Table 38 lists common issues and corrective actions for grid-based applets.

Table 38. Common Grid-Based Applet Issues

Log Examples	What To Do
STATUS:: Succeed Update AWTI List Mgmt Lists Entry Applet Base Status	The utility mapped a control to a different location. Action: Review the modified Applet. Verify that the new location for the control works.
STATUS:: Succeed Delete AWTI List Mgmt Lists Entry Applet Base Status ESN	To avoid possible overlapping, the utilities deleted a locale record for an applet Web template item. Action: Review the modified Applet Web Template and re-create the locale record.
WARNING:: Grid -> Grid merge/upgrade, Skip	Only flow-based form applets are converted to the grid-based Web template. Action: None required.
WARNING:: Upgrade Ancestor "Account Form Applet" Not Found in New Siebel Repository, Skip	The utilities did not find the upgrade ancestor's applet Web template in the New Siebel Repository. Action: Review the applet for invalid Web template items.
WARNING:: Applet Web Template Not Found in New Siebel Repository, Skip	The utility is trying to find the Applet Web Template from New Siebel Repository, but it is not found. Action: Review the applet for invalid Web template items.
WARNING:: No List Column Found, Applet "Program Expenditure List Applet", AWT "Edit", AWTI "Type", Control "Type", Skip	No List Column found for the Applet Web Template Item. Action: Delete the applet Web template item.

Table 38. Common Grid-Based Applet Issues

Log Examples	What To Do
WARNING:: No Control Found, Applet "Expense Item Entry Applet", AWT "Edit", AWTI "Site", Control "Site", Skip	No Control is found for the Applet Web Template Item. Action: Delete the applet Web Template.
WARNING:: Button Control, Applet "Expense Item Entry Applet", AWT "Edit", AWTI "NewQuery", Control "NewQuery", Invoke Method "NewQuery", Skip	If a control has an Invoke Method action defined on it, then the utility treats it as a button and will not remap it. Action: Review the applet and verify that the button control displays correctly.

Related Topics

- ["Troubleshooting Postmerge Siebel User Interface Problems" on page 251](#)
- ["About the Siebel Postmerge Utilities" on page 59](#)
- ["About the Siebel Repository Merge" on page 51](#)
- [Appendix C, "Siebel Incorporate Custom Layout Upgrade Option"](#)

Reviewing Siebel User Interface Navigation

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: "User Interface Navigation 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_ROOT\reppatch\log\reputility.log
```

Table 39 shows examples of common issues and corrective actions.

Table 39. Common User Interface Navigation Issues

Log Examples	What To Do
WARNING:: Project <i>projname</i> is not found in this Repository	The postmerge utilities exclude certain Siebel Tools projects from UI Navigation cleanup. One of the excluded projects was not found in the repository. Action: None required.
WARNING:: Ignoring Screen View Record. View Definition Not Found [View: Account Briefing View]	The utility is unable to read the view definition to determine where it should display. Action: Remove the invalid screen view record that references the invalid view.
WARNING:: Error Writing Category Record, Ignoring Changes [Name: <i>catname</i>]	The utility could not update or insert a record in the Siebel database. Possible causes are that the record already exists or there is a database access problem. Action: Verify that a duplicate <i>catname</i> record does not already exist, and then check database access.
WARNING:: Error Writing Screen View Record, Ignoring Changes [View: <i>viewname</i>]	The utility could not create a screen view record. Possible cause is that the category does not exist. This error is often a consequence of an error updating or inserting a category record. Action: Verify that the screen view category exists, and then check database access.

Related Topics

- ["Troubleshooting Postmerge Siebel User Interface Problems" on page 251](#)
- ["About the Siebel Postmerge Utilities" on page 59](#)
- ["About the Siebel Repository Merge" on page 51](#)

Reviewing Siebel Multi-Value Group Shuttle Applets

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: "Multivalue Group Shuttle Applet Upgrade."

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_ROOT\reppatch\log\reputility.log`

Table 40 lists common issues and corrective actions for MVG applets.

Table 40. Common MVG Applet Issues

Log Entry Example	What To Do
<p>WARNING:: [APPLET: Account Address Mvg Applet (NB)] MVG Applet is Inactive. Ignoring Applet.</p>	<p>This applet is inactive. Action: No action required.</p>
<p>WARNING:: [APPLET: Primary Employee Mvg Applet] [Applet Web Template: Base] Applet Web Template is Inactive. Ignoring Applet Web Template.</p>	<p>The applet Web template is inactive. Action: No action required.</p>
<p>WARNING:: [APPLET: FINS Application Contact Mvg Applet - ACAPS] [CONTROL METHOD INVOKED: ExecuteQuery] Has a Non Standard Control Type.</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 does not have the correct attributes. Action: Revise the control definition and map it to the Edit List mode.</p>
<p>WARNING:: [APPLET: State Model - State Mvg Applet] [CONTROL METHOD INVOKED: UndoQuery] Has a Non Standard Control Type.</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 does not have the correct attributes. Action: Revise the control's definition and map it to the Edit List mode.</p>
<p>WARNING:: [APPLET: LOY Account Address Assoc Applet] [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. Action: Redefine a Cancel button and map it to the Edit List mode.</p>

Table 40. Common MVG Applet Issues

Log Entry Example	What To Do
<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. The user interface standard is that when MVGs first display, they are editable.</p> <p>Action: For MVGs that users can edit, verify that the Web template specified for base mode and edit-list mode are the same. If not, change the base mode Web template so that it is the same as the edit-list template.</p>
<p>WARNING:: [APPLET: Activity Order Mvg Applet] [CONTROL: ExecuteQuery]</p> <p>Control is at an unexpected location. Expected Location is 107</p>	<p>The utilities tried to map a Go button to the default location 107 in Edit List mode. This is part of enabling Popup Inline Query. However, another control is already mapped to this location.</p> <p>Action: Move the control at location 107 to another location, and then map the Go button to location 107.</p>

Related Topics

[“Troubleshooting Postmerge Siebel User Interface Problems” on page 251](#)

[“About the Siebel Postmerge Utilities” on page 59](#)

[“About the Siebel Repository Merge” on page 51](#)

Revising Siebel UI Rich Text Controls

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: "Issue 1: Rich Text Controls (RTC) That Need to Have Properties Reconfigured."

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_ROOT\repatch\log\reputility.log
```

To revise Rich Text Control definitions

- 1 Start Siebel Tools. Set the Object Explorer to Flat.
- 2 From the Views menu, choose Options, then the Object Explorer menu item.
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, then 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, and then Control User Prop.
- 7 In the Controls list, query for the value you wrote down for RTC Body Field. This is the control name.
- 8 For the control, select Control User Prop.
- 9 Define the following control user properties on the control. Assign the values you wrote down for the applet user properties:
 - RTC Graphic Field
 - RTC Link Field

Related Topics

["Troubleshooting Postmerge Siebel User Interface Problems" on page 251](#)

[“About the Siebel Postmerge Utilities” on page 59](#)

[“About the Siebel Repository Merge” on page 51](#)

Reviewing New Siebel UI Aggregate Categories

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Issue 2: New Aggregate Category Records That Should Be Renamed.”

If you have created new views or have modified existing views, then 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_ROOT\reppatch\log\reputility.log
```

Related Topics

[“Troubleshooting Postmerge Siebel User Interface Problems” on page 251](#)

[“About the Siebel Postmerge Utilities” on page 59](#)

[“About the Siebel Repository Merge” on page 51](#)

Revising Siebel Visibility Filters to Display Correctly

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Issue 3: Views that need an applet in View Web Template Item ID 1.”

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_ROOT\reppatch\log\reputility.log
```

To revise visibility filters to display correctly

- 1 Query for one of the views listed in the reputeility.log.
- 2 For the view, select Base, and then View Web Template Item.
- 3 Set the Item Identifier for the first applet in the list to 1.
- 4 Refine the query to display the Parent Category listed in the log.

Related Topics

["Troubleshooting Postmerge Siebel User Interface Problems" on page 251](#)

["About the Siebel Postmerge Utilities" on page 59](#)

["About the Siebel Repository Merge" on page 51](#)

Assigning a Category and Type to Siebel Chart Views

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputeility log section: "Issue 4: Chart Views Needing Migration to Aggregate Type."

The Chart menu item 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, then the chart menu item displays as a view tab.

The postmerge utilities review the repository and verify that Chart views have been defined correctly. The reputeility log lists the screen views that require revision. Review the section of the log referenced above and make the needed revisions.

The log is located here:

```
SIEBEL_TOOLS_ROOT\reppatch\log\reputeility.log
```

To assign a category and type to chart views

- 1 Set Siebel Tools Object Explorer to Types.
- 2 Query for the screen.
- 3 In the Object Explorer, select Screen View.
- 4 Query for the screen views listed in the log.
- 5 Verify that all the views are assigned to the same category (Category Name).
The Category Name can be null.
- 6 Verify that all the views are of type Aggregate View.

Related Topics

[“Troubleshooting Postmerge Siebel User Interface Problems” on page 251](#)

[“About the Siebel Postmerge Utilities” on page 59](#)

[“About the Siebel Repository Merge” on page 51](#)

[Appendix C, “Siebel Incorporate Custom Layout Upgrade Option”](#)

Assigning a Category and Type to Siebel Explorer Views

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Issue 5: Explorer Views Needing Migration to Aggregate Type.”

NOTE: This topic does not apply to 7.x upgrades that used the Incorporate Custom Layout (ICL) feature.

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_ROOT\reppatch\log\reputility.log
```

Assigning a New Web Template to a View

Use the following procedure to assign a new Web template to a view.

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.

Assigning a New Type to a Screen View

Use the following procedure to assign a new type to a screen view.

To assign a new type to a screen view

- 1 In Siebel Tools, query for the screen.
- 2 In Object Explorer, select Screen View.

- 3 For the screen views listed in the log, change the Type to Aggregate View.
- 4 Repeat for each screen listed in the log section.

Related Topics

[“Troubleshooting Postmerge Siebel User Interface Problems” on page 251](#)

[“About the Siebel Postmerge Utilities” on page 59](#)

[“About the Siebel Repository Merge” on page 51](#)

[Appendix C, “Siebel Incorporate Custom Layout Upgrade Option”](#)

Setting Up Navigation to Inaccessible Siebel Detail Views

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Issue 6: Categories Where Parent Applets Are Missing Drilldowns to a Detail View.”

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, then 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_ROOT\reppatch\log\reputility.log`

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:
 - Set the Type value to Detail View.
 - Set the Parent Category value to the Parent Category listed in the log.
- 3 Start the application and navigate to the screen.
- 4 Try to navigate to the detail views listed in the query.

TIP: Use the Web Layout Editor in Tools to identify a detail view containing third-level view tabs that provide navigation to all the detail views. Verify that you can navigate to this view from a drilldown in the screen.

- 5 When you have identified the inaccessible detail view containing third-level view tabs, review the drilldown definitions in Siebel Tools for the list applet in the screen. Define a drilldown to the detail view if one does not exist.

Related Topics

[“Troubleshooting Postmerge Siebel User Interface Problems” on page 251](#)

[“About the Siebel Postmerge Utilities” on page 59](#)

[“About the Siebel Repository Merge” on page 51](#)

[Appendix C, “Siebel Incorporate Custom Layout Upgrade Option”](#)

Eliminating Obsolete Siebel UI Fields

Upgrades from: All Supported Siebel releases. This topic applies only to upgrades using the Incorporate Custom Layout (ICL) feature. For more information, see [Appendix C, “Siebel Incorporate Custom Layout Upgrade Option.”](#)

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: “Issue 7: Fields deprecated from Business Components.”

Some of the business component fields in your installed release might be obsolete in the new release. If you have customized existing applets or created new ones, and you selected ICL during the merge, then the user interface might contain obsolete fields or controls. If a business component field is not available for an applet field or control after the merge, then 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_ROOT\reppatch\log\reputility.log
```

Related Topics

[“Troubleshooting Postmerge Siebel User Interface Problems” on page 251](#)

[“About the Siebel Postmerge Utilities” on page 59](#)

[“About the Siebel Repository Merge” on page 51](#)

[Appendix C, “Siebel Incorporate Custom Layout Upgrade Option”](#)

Reviewing Siebel UI Objects Affected by Incorporate Custom Layout

Upgrades from: All Supported Siebel releases. This topic applies only to upgrades using the Incorporate Custom Layout (ICL) feature. For more information, see [Appendix C, “Siebel Incorporate Custom Layout Upgrade Option.”](#)

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility log section: "Issue 8: List of the items affected by PCL."

If you selected Incorporate Custom Layout (ICL) for the merge, then 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 user interface.

The list has two parts:

- **User interface elements changed between prior Siebel release and current Siebel release.** This part lists the screens, views, and applets that have user interface changes in the new release. If you selected during the merge, then these changes are not reflected in the user interface. Instead, the user interface for the release you are upgrading from has been preserved.
- **User interface 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, then the user interface for the customizations is the same as the release from which you are upgrading.

The log is located here:

```
SIEBEL_TOOLS_ROOT\reppatch\log\reputility.log
```

Related Topics

["Troubleshooting Postmerge Siebel User Interface Problems" on page 251](#)

["About the Siebel Postmerge Utilities" on page 59](#)

["About the Siebel Repository Merge" on page 51](#)

[Appendix C, "Siebel Incorporate Custom Layout Upgrade Option"](#)

Reviewing Required Fields in the Siebel User Interface

Upgrades from: All Supported Siebel releases. This topic applies only to upgrades using the Incorporate Custom Layout (ICL) feature. For more information, see [Appendix C, "Siebel Incorporate Custom Layout Upgrade Option."](#)

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Reputility.log section: "Issue 9: List of Required Fields Missing from the UI."

If you select Incorporate Custom Layout (ICL) during the repository merge, then the Web templates for the user interface you are upgrading from are used to display the user interface after the merge. The Web templates for the new release are not used.

The Web templates used might not contain all the fields required by the new release. For example, a required field might 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_ROOT\reppatch\log\reputility.log
```

Assigning an Item Identifier to Siebel Web Template Items

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

All View Web Template items must have a unique item identifier. Do the following to ensure this:

- See 476806.1 (Article ID) on My Oracle Support and resolve duplicate item identifiers. This document was previously published as Siebel Technical Note 400.
- In Siebel Tools, query View Web Templates for a null Item Identifier. Replace any null Item Identifiers with a number. If an item is part of a group of items that have the same grandparent (GParent), then the number must be unique in the group.

Related Topics

[“Troubleshooting Postmerge Siebel User Interface Problems” on page 251](#)

[“About the Siebel Postmerge Utilities” on page 59](#)

[“About the Siebel Repository Merge” on page 51](#)

[Appendix C, “Siebel Incorporate Custom Layout Upgrade Option”](#)

17 Postupgrade Tasks for Siebel Business Applications

This chapter provides tasks to be performed on Siebel Business Applications following a successful upgrade and repository merge. This chapter includes the following topics:

- [Activating License Keys on page 267](#)
- [Generating Siebel Reporting Relationships on page 268](#)
- [Setting Up Siebel Global Time Zone Support on page 269](#)
- [Displaying Regions in Siebel Marketing on page 270](#)
- [Configuring Siebel Marketing Purchase Orders for Display on page 270](#)
- [Upgrading Siebel Attribute Pricing on page 271](#)
- [Verifying Aggregate Discounts in Siebel Pricer on page 272](#)
- [Upgrading Inbound Siebel Workflows on page 273](#)

Activating License Keys

In Siebel Innovation Pack 2016, the license keys for Siebel CRM base applications that were previously provided in seed data in the Siebel database are inactive. License keys entered by customers are unchanged. A new utility is provided for activating or deactivating the license keys that you require. You run the License Key Activation utility after installing a new Siebel database, running Incremental Repository Merge (for migration installations), or completing a full database upgrade.

You can find license key information for Siebel Business Applications at Oracle's license codes site. For the Siebel license keys, see

<http://licensecodes.oracle.com/siebel.html>

The License Key Activation utility is supported on all operating systems and databases for Siebel Business Applications.

To start the License Key Activation utility

- 1 On the computer where you installed Siebel Server, navigate to the following location:

```
SI EBSRVR_ROOT\bin
```

- 2 Run the following program, according to your operating system:

- Microsoft Windows: `licensekeymodule.bat`
- UNIX: `licensekeymodule.sh`

- 3 Enter valid data for the following fields:
 - **Siebel Server Location.** The installation path for this Siebel Server.
 - **ODBC DSN.** The ODBC data source for the Siebel database.
 - **Table Owner.** The table owner for the Siebel database.
 - **Username.** The user name for logging into the Siebel database.
 - **Password.** The password for this user.
 - **DB Platform.** The RDBMS platform, either ORACLE, DB2UDB, DB2390, or MSSQL.
 - **Log folder.** The folder in which the log file licenseKeys.log is created. This log file shows database connection information for troubleshooting purposes, and lists all of the license keys that were activated or deactivated in each session.
- 4 Click Login.

The license key activation screen appears, which lists Siebel CRM license keys.
- 5 For each license key module whose activation status you want to change, click the Active Flag check box to activate or deactivate this license key.
- 6 To apply your selections to the Siebel database, click Apply. Or, to reset any changes you have made in this screen, or since you last clicked Apply, click Reset.

CAUTION: After you have clicked Apply, the Reset button does not reset activation settings to their original state. However, you can change the activation status and click Apply again.
- 7 To exit the utility, click the X in the top right corner.

Related Books

Siebel Installation Guide for the operating system you are using

Generating Siebel Reporting Relationships

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

The Generate Reporting Relationships process needs to be executed after the upgrade process and whenever the denormalized hierarchy structure (S_PARTY_RPT_REL) becomes out of sync with the data in the normalized tables (S_PARTY). Tables can become out of sync in the following cases:

- After upgrading, the organizational hierarchy (even if there is only one organization) must be established to maintain appropriate visibility in the views cited above.
- When Siebel Enterprise Integration Manager is used to import or update any of the hierarchies (positions, organizations, or access groups).

There are three visibility hierarchies: position, organization, and access groups. These hierarchies are denormalized and maintained in the table S_PARTY_RPT_REL. These denormalized hierarchies are necessary for executing visibility modes that go up or down a hierarchy. For example:

- **Manager view mode.** My Team's Accounts View displays all accounts on which managers and their subordinates are working.
- **Suborganizations view mode.** All Contacts across My Organizations View displays all contacts who associated with either my organization or any of my organization's suborganizations.

The Generate Reporting Relationships process rebuilds the denormalized relationships in the S_PARTY_RPT_REL table so that the hierarchical view modes display the correct information. The basic operation of the function is to empty the S_PARTY_RPT_REL table and then walk through each S_PARTY record to re-create the denormalized hierarchical structures in the table. This process generates a large number of transactions for Siebel Remote users and regional nodes.

This operation is time and CPU/memory-intensive. The process might take several minutes, depending on the size and complexity of your organizational structures. Do not perform this when you are running other memory-intensive processes.

To generate reporting relationships

- 1 If you have an active Siebel Remote environment, then confer with an Oracle Administrator. The Administrator must arrange for the Transaction Processor component to be paused before performing this procedure.
- 2 In the Siebel Web Client application, navigate to Administration - Group, and then the Positions view.
- 3 Click the Generate Reporting Relationships button in the Position List Applet NB. Note that generating the reporting relationship might cause a large number of Siebel Remote transactions to be generated.
- 4 When this has completed, restart the Transaction Processor.

Setting Up Siebel Global Time Zone Support

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

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.

It is strongly recommended that you operate your production environment with global time zone enabled.

For information on setting up and managing UTC, see *Siebel Global Deployment Guide*.

Displaying Regions in Siebel Marketing

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

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 Navigate to Administration – Location, and then Marketing Regions.
- 2 Create new marketing regions to correlate to the previous LOV values for the Region field.
For more information, see *Siebel Marketing Installation and Administration Guide*.
- 3 Rename the Region LOV field and add it to the applets in the user interface.
This provides backward compatibility.

Configuring Siebel Marketing Purchase Orders for Display

Environments: Development, production test, production.

At Siebel CRM 8.0, the relationship between campaign and order is changed from M:1 to M:M and is displayed in a new view, Marketing Purchase Order.

The upgrade process does not set up campaign-related purchase orders created in previous releases so that they display in the new view.

If you want to display purchase orders from previous releases, then you must run a script after the upgrade. The script changes the order type from Purchase Order to Marketing Purchase Order for campaign-related purchase orders. You can also use the script to change the order type of campaign-related sales orders to Marketing Purchase Order. This allows these purchase orders to display in the new view.

To configure marketing purchase orders

- 1 Run the following script against the Siebel database:

Windows: `DBSRVR_ROOT\common\MktgPurchaseOrder.sql`

UNIX: `DBSRVR_ROOT/common/MktgPurchaseOrder.sql`

For campaign-related purchase orders, this script sets order type to Marketing Purchase Order. This allows these purchase orders to display in the Marketing Purchase Order view.

- 2 To configure campaign-related sales orders to display in the new view, open the script with a text editor, and locate the following section:

```
where ORDER_TYPE_ID =  
(select ROW_ID from S_ORDER_TYPE  
where NAME = 'Purchase Order')
```

Make the following change:

```
where ORDER_TYPE_ID =  
(select ROW_ID from S_ORDER_TYPE  
where NAME = 'Sales Order')
```

- 3 After making the change, run the script again.

Upgrading Siebel Attribute Pricing

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

NOTE: For information on changes to Copy and Revise functionality for Quotes and Orders, see 473867.1 (Article ID). This article was previously published as Siebel Alert 1218.

At Siebel CRM 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 the Administration-Business Service screen, and then to Simulator view.
- 4 Create a new record.
- 5 Click Run to start the business method.

Related Topics

[“Upgrade Planning for Siebel Pricer and Order Management” on page 81](#)

Verifying Aggregate Discounts in Siebel Pricer

Environments: Development, production test, production.

Platforms: Windows, UNIX, IBM z/OS.

NOTE: For information on changes to Copy and Revise functionality for Quotes and Orders, see 473867.1 (Article ID). This article was previously published as Siebel Alert 1218.

At Siebel CRM 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 CRM 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 CRM 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, then 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 preupgrade bundle factor. Verify that it has the lowest sequence number. If not, then 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, then revise the numbers in all three columns as required.

Related Topics

[“Upgrade Planning for Siebel Pricer and Order Management” on page 81](#)

Upgrading Inbound Siebel Workflows

Environments: Development, production test, production.

Change inbound workflows that contain a “String” type process property to pass the value into type Binary; otherwise, the workflow presents the following error message:

```
Output argument '<Value>' in step 'Read from File' contains data that cannot be passed to string type property 'InputXML'. Data type: 'MEMBLOCK'; String representation of data body: '<?xml version="1.0" encoding="UTF-8"?>?'
```

Related Topics

[“Upgrade Planning for Siebel Workflow Designer” on page 76](#)

18 Tuning the Siebel Upgrade Files

This chapter provides guidelines for preparing your Siebel Business Applications data for the Siebel database upgrade. This chapter includes the following topics:

- [Starting and Stopping Siebel Upgrade Tuner on page 275](#)
- [Managing Parallel Threads Using Siebel Upgrade Tuner on page 277](#)
- [Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner on page 280](#)
- [Transferring UNIX Files for Use by Siebel Upgrade Tuner on page 283](#)
- [Rolling Back Siebel Upgrade Tuner Changes on page 286](#)

Starting and Stopping Siebel Upgrade Tuner

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

Platforms: Windows and UNIX only.

Databases: All databases.

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.

Requirements

- In the production test environment, you must have completed the production upgrep step.
- You must have run the Logparse utility on the production upgrep log files. See [“Summarizing Siebel Log Files Using the Logparse Utility” on page 159](#).
- 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 283](#).

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
```

In this command, *LOGPARSE_SUMMARY_LOCATION* is the location of the *summary.xml* file generated by the Logparse 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 and then Exit.
- 3 Click Yes in the pop-up window that asks you to confirm if you want to save and exit.
Upgrade Tuner applies your changes to the upgrade files and exits.

Discarding Your Session Changes and Exiting Upgrade Tuner

Use this procedure to discard the changes you have made in the current session and exit Upgrade Tuner. No changes are made to upgrade files.

Parallelize Table or Index Creation Pages

Changes are discarded and do not display the next time you start Upgrade Tuner, if:

- You moved tables or indexes between threads, these changes are discarded.
- You created new threads, the threads are discarded.

Deactivate 0-Row SQLs

Changes are discarded and do not display the next time you start Upgrade Tuner, if:

- You deactivated a command, this change is discarded. The next time you start Upgrade Tuner, no check mark displays in the Inactive column. The command remains active in the SQL file.
- You activated a command, this change is discarded. The next time you start Upgrade Tuner, a check mark displays in the Inactive column, and the command remains inactive in the SQL file.

To discard your changes and exit Upgrade Tuner

- 1 Click Cancel.
- 2 Click Yes in the pop-up window that asks if you to confirm you want to discard your changes and exit.

Related Topics

[“About Tuning Siebel Production Upgrade Files” on page 63](#)

Managing Parallel Threads Using Siebel Upgrade Tuner

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.

Requirements

- In the production test environment, you must have completed the production upgrep step.
- You must have run the Logparse utility on the production upgrep log files. See [“Summarizing Siebel Log Files Using the Logparse Utility” on page 159](#).
- 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 283](#).
- Start Upgrade Tuner. See [“Starting and Stopping Siebel Upgrade Tuner” on page 275](#).

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 the Serial Thread column head.
- **Contents of a thread.** Click the column head for that thread. To reverse the sort-order, click the column head again.
- **All items in all threads sorted by cost.** Click the Cost per Table column head. To reverse the sort-order, click 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.

Creating Parallel Thread 1 and Parallel Thread 2

Use the following procedure to create Parallel Thread 1 and Parallel Thread 2.

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.

Creating Additional Parallel Threads

Use the following procedure to create additional parallel 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.

Deleting a Thread Other Than Parallel Thread 1 or Parallel Thread 2

Use the following procedure to delete a thread other than Parallel Thread 1 or 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.

Deleting Parallel Thread 1 and Parallel Thread 2

Use the following procedure to delete Parallel Thread 1 and Parallel Thread 2.

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 might need to experiment with the number of parallel threads to optimize upgrade performance.

Related Topics

[“About Tuning Siebel Production Upgrade Files” on page 63](#)

Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner

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

Platforms: Windows and UNIX only.

Databases: All databases.

The upgrade scripts support all the tables in the Siebel data model. This support means the tables might 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 upgrade.

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, `db2\upgrade\V8_x`.

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.

Requirements

- In the production test environment, you must have completed the production upgrep step.
- You must have run the Logparse utility on the production upgrep log files. See [“Summarizing Siebel Log Files Using the Logparse Utility” on page 159](#).
- 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 283](#).
- 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 275](#).

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 the SQL File column head. This action sorts the file names alphabetically. To reverse the sort order, click 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 Not applicable.
- 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 Not applicable 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 Not applicable.
- When you save and exit, Upgrade Tuner activates the command in the SQL file.

- The next time you start Upgrade Tuner, Not applicable 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 upgrep 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 upgrep.

Related Topics

[“About Tuning Siebel Production Upgrade Files” on page 63](#)

Transferring UNIX Files for Use by Siebel Upgrade Tuner

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

Platforms: UNIX only.

Upgrade Tuner is part of the Siebel Server and runs only on Windows. To tune production upgrade files on a UNIX computer 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.

Requirements for the UNIX Host

- In the production test environment, you must have completed the production upgrep step.
- You must have run the Logparse utility on the production upgrep log files. See [“Summarizing Siebel Log Files Using the Logparse Utility” on page 159](#).

Requirements 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.** Copy the following scripts using FTP to the Windows computer 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 computer.
&Username	This value is the user name used to open an FTP session with the UNIX computer (for example sadmin).
&WindowsTempDir	This value is the full path of the target directory on the Windows computer. 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	<code>\$SIEBEL_ROOT/ilog/upgprep_prod_VERSION</code> For example, <code>\$SIEBEL_ROOT/ilog/upgprep_prod_8x/summary.xml</code>
master_upgprep_prod_VERSION.ucf	<code>\$SIEBEL_ROOT/bin</code> For example, <code>\$SIEBEL_ROOT/bin/master_upgprep_prod_8x.ucf</code>
schema*.ddl	<code>DBSRVR_ROOT/DBPLATFORM</code> For example, <code>DBSRVR_ROOT/Oracle/schema.ddl</code> , <code>schema_t1.ddl</code> , <code>schema_t2.ddl</code>
driver_upgprep_prod_VERSION.ucf	<code>DBSRVR_ROOT/DBPLATFORM/upgrade/VERSION</code> For upgrades from Siebel CRM version 8.1.1.10 on Oracle Database, an example would be <code>DBSRVR_ROOT/Oracle/upgrade/v8_1_1_10/driver_upgprep_prod_v81110.ucf</code> .
*.sql	<code>DBSRVR_ROOT/DBPLATFORM/upgrade/VERSION</code> For example, for upgrades from Siebel CRM version 8.1.1.10 on Oracle Database, an example would be <code>DBSRVR_ROOT/Oracle/upgrade/v8_1_1_10/preschm.sql</code> .

- 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.** Use FTP to copy 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 computer.

Related Topics

[“About Tuning Siebel Production Upgrade Files” on page 63](#)

Rolling Back Siebel Upgrade Tuner Changes

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

Platforms: Windows and UNIX only.

Databases: All databases.

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 deployments on UNIX. You can roll back upgrade files to a previous version on the UNIX host. You do not have to transfer the files to a Windows host and rerun Upgrade Tuner.

Upgrade File Versions

Before Upgrade Tuner saves changes to the upgrade files, it does the following:

- If this your first Upgrade Tuner session, Upgrade Tuner saves the current driver and SQL files to .orig, for example dri ver_upgrep_prod_8x.ucf.orig.
- If this is the second session or all later sessions, Upgrade Tuner saves the current driver and SQL files to .old, for example dri ver_upgrep_prod_8x.ucf.old.

To roll back, you replace the upgrade file with the .old or .orig version.

Guidelines for Rolling Back Upgrade Files

Use the following guidelines to roll back upgrade files:

- To roll back to the previous Upgrade Tuner session, replace the driver or SQL file with the .old version.
- To roll back to the original version of the file, before any Upgrade Tuner modifications, replace the driver or SQL file with the .orig version.
- You can roll back the driver and SQL files separately. For example, you can roll-back to the original driver file while retaining the most recent changes to SQL files.

- You do not have to roll back all SQL files at once. For example, you can roll back some SQL files while retaining the most current version of others.
- You do not have to roll back all the commands in an SQL file. You can edit the file and activate or deactivate as many commands as desired.
- Manually editing the `dri ver_upgrep_prod_VERSION.ucf` file is not recommended.
- If you roll back to a session with fewer threads, then you do not have to delete any `schema_t#.ddl` or `schema_i#.ddl` thread-files. The deletion is unnecessary because Upgrade Tuner removes from the driver file the steps that execute the deleted threads.

The following procedures use `dri ver_upgrep_prod_8x.ucf` and `pret.sql` as examples.

Rolling Back to the Previous Session

Use this procedure to discard your most-recent session and roll back to the previous session.

To roll back to the previous session

- 1 Save a copy of `dri ver_upgrep_prod_8x` and `pret.sql` to new names.
- 2 Copy `dri ver_upgrep_prod_8x.old` to `dri ver_upgrep_prod_8x`.
- 3 Copy `pret.sql.old` to `pret.sql`.
- 4 Restart Upgrade Tuner.

Rolling Back to the Original Upgrade Files

Use this procedure to discard all Upgrade Tuner changes and roll back to the original upgrade files.

To roll back to the original upgrade files

- 1 Save a copy of `dri ver_upgrep_prod_8x` and `pret.sql`.
- 2 Copy `dri ver_upgrep_prod_8x.orig` to `dri ver_upgrep_prod_8x`.
- 3 Copy `pret.sql.org` to `pret.sql`.

Activating or Deactivating SQL Commands Manually

Use this procedure to activate or deactivate individual zero-row SQL commands by editing the SQL file.

To activate or deactivate a command by editing an SQL file

- 1 Save a copy of the SQL file to a new name.
- 2 Open the `.sql` file (not the copy) and locate the desired SQL command.
Commands begin `Run_SQL_#`, for example `Run_SQL_100`.

- 3 Edit the command as follows:
 - To activate the command, delete the element (Execute=N)
 - To deactivate a command, add the element (Execute=N)
Insert the element on a line by itself after "Run_SQL_# =".
- 4 Save the file.

Related Topics

["About Tuning Siebel Production Upgrade Files" on page 63](#)

A

Implementing Siebel High-Availability Upgrade Using Oracle GoldenGate to Upgrade to the Current Release

This appendix describes how to implement the Siebel High-Availability (HA) upgrade using Oracle GoldenGate. It includes the following topics:

- [Overview of Oracle GoldenGate Implementation on page 290](#)
- [About Siebel High Availability Upgrade Files on page 291](#)
- [Extracting Oracle GoldenGate Files on page 291](#)
- [Example of Implementing a Siebel High-Availability Upgrade on page 298](#)
- [Limitations When Performing a Siebel Upgrade with Oracle GoldenGate on page 300](#)

Overview of Oracle GoldenGate Implementation

You use the standard Siebel upgrade process detailed in this guide to upgrade your Siebel database from one supported Siebel CRM version to another version. The duration of the upgrade process and the potential for having your live environment unavailable during the process depend on the database size and other factors. Performing a high-availability upgrade using Oracle GoldenGate enables you to keep your Siebel CRM production environment available to your customers during the upgrade process while also causing the least interruption to your business users because you do not have to shut down your Siebel environment during the upgrade. [Figure 8](#) provides an overview of the Oracle GoldenGate architecture.

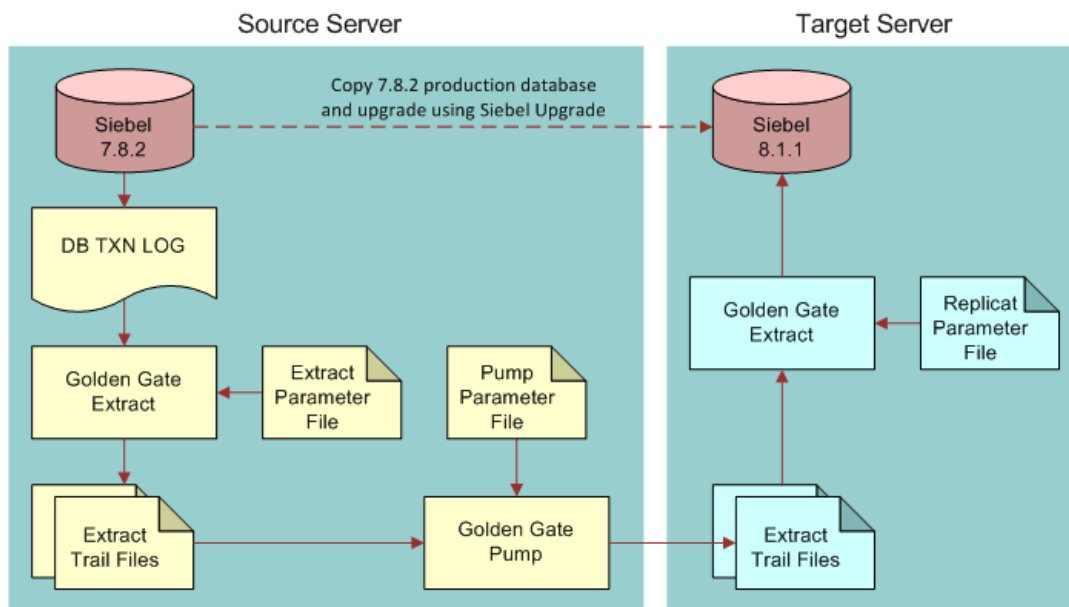


Figure 8. Siebel Upgrade Using Oracle GoldenGate

[Figure 8](#) shows two servers with Oracle GoldenGate software installed on each of them. Each server has a database instance: one is the source system Siebel CRM version 7.8.2, and the other has the target system, the current release of Siebel CRM. The directory structure and files reflect this environment.

For example, other upgrade situations might involve a single server hosting both databases (either in the same instance or in two different instances) with one instance of Oracle GoldenGate software. In this situation, the Oracle GoldenGate pump process and its parameter files are not required. The extract files created by the Oracle GoldenGate Extract process can be used by the Replicat process. In this case, the file prefix value in the replicat OBEY file (such as r782_811.oby) must be changed to the one used in the Extract process.

Related Topics

For more information on Oracle GoldenGate, see Oracle Technology Network (<http://www.oracle.com/technetwork/indexes/documentation/index.html>).

About Siebel High Availability Upgrade Files

Oracle GoldenGate software is implemented with a set of files that enable you to decrease the downtime of your Siebel production database during an upgrade. These files are provided in the Siebel Database Server installation directory:

`SI EBEL_ROOT/dbsrvr/COMMON`

These files are compressed with names like the following, depending on the database platform:

- GoldenGateSIA782_SIA811_HA_Upg_ORA.zip
- GoldenGateSIA782_SIA811_HA_Upg_ORA.tar.gz
- GoldenGateSIA782_SIA811_HA_Upg_DB2UDB.tar.gz
- GoldenGateSIA782_SIA811_HA_Upg_DB2UDB.tar.gz
- GoldenGateSIA782_SIA811_HA_Upg_MSSQL.tar.gz
- GoldenGateSIA782_SIA811_HA_Upg_MSSQL.tar.gz

Extracting Oracle GoldenGate Files

The ZIP or tar files include a number parameter files and OBEY files, which can be used with Oracle GoldenGate software. For more information, see “About Siebel High Availability Upgrade Files” on page 291.

NOTE: Some of the files are specific to a database platform. Depending on your database platform, database-specific ZIP or tar files must be used to extract the Oracle GoldenGate parameter files.

To extract compressed Oracle GoldenGate files

- 1 From your `dbsrvr` directory, extract the files from the following ZIP or tar file:

`GoldenGateSIA782_SIA811_HA_Upg_*. *`

- 2 View the following files after extracting them:

`SIA8119/dbsrvr/COMMON`

- GoldenGateSIA782_SIA811_HA_Upg_ORA.zip (Windows only)
- GoldenGateSIA782_SIA811_HA_Upg_DB2UDB.zip (Windows only)
- GoldenGateSIA782_SIA811_HA_Upg_MSSQL.zip (Windows only)
- GoldenGateSIA782_SIA811_HA_Upg_ORA.tar.gz (UNIX and Linux only)

- GoldenGateSIA782_SIA811_HA_Upg_DB2UDB.tar.gz (UNIX and Linux only)
- GoldenGateSIA782_SIA811_HA_Upg_MSSQL.tar.gz (UNIX and Linux only)

SI A8119/dbsrvr/Gol denGate/SI A782_811/SourceServer/GLOBALS

- dirprm
 - d_782sia.prm
 - e_782sia.oby
 - e_782sia.prm
 - mgr.prm
 - p_782sia.oby
 - p_782sia.prm
 - s782sia_trandata.oby
 - siebelMacro.mac
- dirrpt
 - extractReports
 - pumpReports

SI A8119/dbsrvr/Gol denGate/SI A782_811/TargetServer/GLOBALS

- dirprm
 - mgr.prm
 - r782_811.oby
 - r782_811.prm
 - siebelMacro.mac
- dirrpt
 - replicatReports

- 3 Assuming a directory structure illustrated in [Step 2 on page 291](#) . . . /SI A8119/dbsrvr/COMMON, run tar or extract the files to the following directory, depending on your operating system:

- **Windows.** cd. . . /SI A8119/dbsrvr
- **UNIX.** tar -xvf . /COMMON/Gol denGateSI A782_SI A811_HA_Upg_ORA. tar. gz

See [Table 41](#) for a description of the source server files and directories.

Table 41. Oracle GoldenGate Source Server Files and Directories

File or Directory Name	Description	Required or Optional Changes
GLOBALS	Global Parameters file for the Oracle GoldenGate processes. This is an empty file.	You can add the required parameters for your environment.
d_782sia.prm	Parameters file for the Oracle GoldenGate executable defgen. This file contains the list of Siebel CRM version 7.8.2 tables from which data is extracted from the Siebel CRM version 7.8.2 database log file by Oracle GoldenGate.	You can do the following: <ul style="list-style-type: none"> ■ Change the login information for the source database. ■ Change the table owner and schema name for the tables. ■ Add your custom tables (X_* tables).
e_782sia.oby	The OBEY file used to create the Oracle GoldenGate Extract process.	You can do the following: <ul style="list-style-type: none"> ■ (Optional) Extract the file size. ■ (Optional) Add start command.
e_782sia.prm	The parameter file for the Oracle GoldenGate Extract process. This file contains the list of Siebel 7.8.2 application tables from which data is extracted from the Siebel 7.8.2 database log file by Oracle GoldenGate. The tables include the following: <ul style="list-style-type: none"> ■ Siebel Enterprise Integration Manager tables ■ Siebel repository tables ■ Other tables related to Siebel Remote log files and other similar files 	You can do the following: <ul style="list-style-type: none"> ■ (Optional) Report options. ■ Add custom tables designated by the (X_*) prefix.

Table 41. Oracle GoldenGate Source Server Files and Directories

File or Directory Name	Description	Required or Optional Changes
mgr.prm	The parameter file for the Oracle GoldenGate Manager process. The same file is included in both source and target di rprm directories.	You can do the following: <ul style="list-style-type: none"> ■ (Optional) Port and Port List numbers. ■ (Optional) other parameters.
p_782sia.oby	The OBEY file used to create the Oracle GoldenGate pump process.	You can do the following: <ul style="list-style-type: none"> ■ (Optional) pump file size. ■ (Optional) Add the start command.
p_782sia.prm	The parameter file for the Oracle GoldenGate pump process.	No changes required.
s782sia_trandata.oby	The OBEY file that contains the Oracle GoldenGate ADD TRANDATA command. Use this command to get a list of the tables whose data will be extracted from the Siebel database 7.8.2. This command also requests supplemental log information for the tables to be executed by the database server. For some tables, additional columns are included because those columns are required to populate the tables in the upgraded target database.	You can do the following: <ul style="list-style-type: none"> ■ Change the login information for the source database. ■ Change the table owner and schema name for the tables. ■ Add custom tables added by the customer (X_* tables).

Table 41. Oracle GoldenGate Source Server Files and Directories

File or Directory Name	Description	Required or Optional Changes
siebelMacro.mac	<p>Contains the Oracle GoldenGate macros that are used to define the following:</p> <ul style="list-style-type: none"> ■ Source database login information ■ Target database login information ■ Source database table owner and schema name ■ Target database table owner and schema name ■ Target server name ■ Target manager port number ■ Primary language for the source Siebel database <p>These macros are used in the parameter files for the Extract process, pump, and Replicat process. For more information, see "Oracle GoldenGate Macros" on page 297.</p>	
extractReports	The directory where the report files for the Extract process are stored.	Ensure that this directory is created in the di rrpt directory of Oracle GoldenGate on the source server.
pumpReports	The directory where the report files for the pump process are stored.	Ensure that this directory is created in the di rrpt directory of Oracle GoldenGate on the source server.

See [Table 42](#) for a description of some of the target server files and directories.

Table 42. Oracle GoldenGate Target Server Files and Directories

File or Directory Name	Description or Usage	Changes to Make
GLOBALS	Global parameters file for the Oracle GoldenGate processes. This file has only one line, which is required: checkpoint table GGS_CHECKPOINT.	<ul style="list-style-type: none"> ■ (Optional) Add the required parameters for your environment. ■ Change the checkpoint table name if you have a defined checkpoint table with any other name. ■ Add a checkpoint table in the target database by using the command: ADD CHECKPOINTTABLE table_owner_name.GGS_CHECKPOINT
mgr.prm	The parameters file for the Oracle GoldenGate Manager process. This file is included in both the source and the target di rprm directories.	<ul style="list-style-type: none"> ■ (Optional) provide the port and the post list numbers. ■ (Optional) provide the other required parameters.
r782_811.oby	The OBEY file to create the Oracle GoldenGate Replicat process.	(Optional) Add the start command.
r782_811.prm	This file contains the list of Siebel SIA 8.1.1 tables for which the transaction data that was extracted from Siebel 7.8.2 SIA database is replicated in the target Siebel 8.1.1 SIA database, applying the required upgrade logic to the table.	<ul style="list-style-type: none"> ■ (Optional) Add the report options. ■ Add any additional upgrade logic for Siebel database tables and custom tables (X_*).

Table 42. Oracle GoldenGate Target Server Files and Directories

File or Directory Name	Description or Usage	Changes to Make
siebelMacro.mac	<p>Contains the Oracle GoldenGate macros that define following:</p> <ul style="list-style-type: none"> ■ Source database login information ■ Target database login information ■ Source database table owner and schema name ■ Target database table owner and schema name ■ Target server name, ■ Target manager port number ■ Primary language for the source Siebel database <p>This file has the same values as the file on the source server. See Table 41 for more information.</p> <p>These macros are used in the parameter files for the Extract, pump, and Replicat process.</p>	<p>You can copy the file from the instance of Oracle GoldenGate on the source database after the macro values have been changed for this file in the source server.</p>
replicatReports	<p>The directory where the report files for the Replicat process are stored.</p>	<p>Ensure that this directory is created in the Oracle GoldenGate <code>di rrpt</code> directory on the target server.</p>

NOTE: The standard Siebel upgrade process, described in this guide, does not use the ZIP files, tar files, or the extracted files and directories. Oracle GoldenGate software does not use the ZIP files, tar files, extracted files, and directories directly. These files must be located in the `di rprm` directory for the Oracle GoldenGate software installation.

The extracted files must be copied to the `di rprm` directory in the Oracle GoldenGate software installation and modified to reflect your environment. For example, modify the following: database login and password information, table owner and schema names, port numbers and the target server name and so on. Make sure that the directories mentioned above have been created under the `di rrpt` directory where Oracle GoldenGate software has been installed.

Oracle GoldenGate Macros

The Oracle GoldenGate macros identify the following:

- Source database table owner and schema name.

- Target database table owner and schema name.
- Source database login and password information.
- Target database login and password information.
- The remote host name. In other words, the server name where the Oracle GoldenGate instance for the target database is located.
- The port number used by the Oracle GoldenGate Manager process for the Oracle GoldenGate instance in the target database.
- Optionally, the macro determines the string value for the DB_LAST_UPD_SRC column in the target database.
- The language code for the primary language of your Siebel CRM installation on the source database.

Example of Implementing a Siebel High-Availability Upgrade

This topic provides an example of how to implement a Siebel high-availability upgrade, using Oracle GoldenGate software:

- 1 Install Oracle GoldenGate software on the source and target servers, and configure them.
For more information on Oracle GoldenGate, see Oracle Technology Network (<http://www.oracle.com/technetwork/indexes/documentation/index.html>).
- 2 Install the current release of Siebel CRM, including the Siebel Server, Siebel Database Server, and Siebel Gateway Name Server.
- 3 From the installation of the current release of Siebel CRM, extract the Oracle GoldenGate files from the installed ZIP or tar file as described in “[Extracting Oracle GoldenGate Files](#)” on page 291, and copy the extracted files to the di rprm directory of the Oracle GoldenGate installation.
- 4 Make all the required changes to those files. Provide database login and password information, as well as table owner, target server name, and so on. Create the required directories under the di rrpt directory.
- 5 Customize the Oracle GoldenGate parameter files to include any custom tables that you have previously created (X_*), including any logic required for those tables.
- 6 Using the Oracle GoldenGate executable defgen and the modified parameter file d_782sia.prm, generate the schema definitions for Siebel 7.8.2 tables.
This parameter file defines the output file as Siebel782.dat. This file is generated under the di rdat directory. Copy this file to the target server's di rdat directory.
- 7 Create a new database instance on the target server to host the Siebel 7.8.2 database that you will upgrade.
- 8 Ensure that all disconnected users synchronize their local database with the server database.
- 9 Bring the production database to an idle state by disconnecting all users and stopping all Siebel Server processes.

- 10** Take a full dump of the database. Using the dump file, create a new copy of the database in the target server.
This database will be upgraded to the current release of Siebel CRM.
- 11** Start the Oracle GoldenGate Extract and pump process on the source server to capture all new transactions in the Siebel 7.8.2 database. Start the Siebel application and allow users to connect and continue to use the application.
- 12** Ensure that the Oracle GoldenGate processes are running and extract files are created on the source server and are created in the target server by the Oracle GoldenGate pump process.
- 13** Start the standard Siebel upgrade process on the target server to upgrade the Siebel 7.8.2 database and Siebel Business Applications to the current release of Siebel CRM.
- 14** Monitor the Oracle GoldenGate Extract process and pump process to ensure that they are capturing data from the source server. Move the data to the target server continuously.
- 15** After the upgrade to Siebel 8.1.1.x is complete and the upgrade is validated, start the Oracle GoldenGate Replicat process on the target server to apply the captured transactions in the extract files to the upgraded database for the current release of Siebel CRM.
- 16** After the Oracle GoldenGate Replicat process has retrieved all the transactions in the source Siebel 7.8.2 database, all disconnected users must synchronize their local database with the server database and stop using the Siebel application in disconnected mode.
- 17** Bring the production database to an idle state again, by disconnecting all users and stopping all Siebel Server processes.
- 18** Check that the last few identified transactions on the source database are reflected in the upgraded database for the current release of Siebel CRM. Start all the required Siebel server processes and start the Siebel application. The upgraded Siebel CRM database and Siebel applications are now ready for use.
- 19** Migrate all users so that they can start using the new upgraded database and Siebel applications for the current release of Siebel CRM. Shut down the Siebel 7.8.2 database, and stop the Oracle GoldenGate Extract, pump, and Replicat processes.
- 20** If some users are disconnected (Siebel Mobile Web Client users), then create a new version of the local database for these users, for the current release of Siebel CRM. After those users initialize their local database with the new local database extract, they can start using the application in disconnected mode.

The upgrade of Siebel 7.8.2.x to the current release of Siebel CRM, using Oracle GoldenGate with minimum downtime of the Siebel application (version 7.8.2), is now complete.

Limitations When Performing a Siebel Upgrade with Oracle GoldenGate

The following limitations apply when performing a Siebel upgrade with Oracle GoldenGate software:

- **You must not perform any of the following operations in the Siebel 7.8.2 database after the target database has been created when preparing to upgrade to the current release:**
 - Creating or modifying schemas, business components, applets, views, and so on in the Siebel Repository
 - Creating or modifying workflows, assignment rules, audit rules, and so on
- **Siebel Enterprise Integration Manager tables are not replicated.** This limitation is because all Siebel Enterprise Integration Manager operations on the source database result in changes to the application tables, and those transactions from the application tables are replicated.
- **Transactions in the docking transactions log tables are not replicated.** This limitation is because disconnected users are expected to get a newly installed local database from the upgraded Siebel database following the completed upgrade when Oracle GoldenGate has applied all the Siebel CRM version 7.8.2 transactions to the database for the current release.
- **Log files and tables with data that are temporary are not replicated.**
- **For LOV-based column values that are used in the upgrade steps, the Oracle GoldenGate Replicat parameter file contains values in English (ENU) from the S_LST_OF_VAL table.** For customers whose primary language is not English, some of these steps must be changed to use values in the customer's installed language.
- **Using the OBEY files to define the Extract, pump, and Replicat processes and using the GGSCI command does not start those processes.** You must run the start command from within the Oracle GoldenGate secure command interface (GGSCI) when you are ready to start the processes.

B

Overview of Performing a Siebel Database Upgrade

This appendix provides process information on how to upgrade your Siebel database. It includes the following topics:

- [Roadmap for Siebel Database Environment Upgrade on page 301](#)
- [Process of Upgrading a Siebel Development Environment on page 308](#)
- [Process of Upgrading a Siebel Production Test Environment on page 315](#)
- [Process of Tuning Siebel Upgrade Performance on page 320](#)
- [Process of Upgrading a Siebel Production Environment on page 324](#)

Roadmap for Siebel Database Environment Upgrade

The following roadmap provides an overview of tasks required to upgrade your Siebel environment. Use this roadmap to determine the applicability of each task to each environment. For example, if a task has *No* listed in the Prod column, then that task is not required for production test, or production environment upgrades. All tasks required for the production test upgrade are also required for the production environment upgrade. The development environment has more required tasks. Print and review this roadmap as you plan and work through your upgrade.

NOTE: Links to other topics in this guide are provided throughout this chapter. These links provide additional information that cannot be provided in these brief overviews. You must, however, perform the upgrade in the proper order. For information on general planning, see this chapter as well as [Chapter 3, “Siebel Database Upgrade Planning,”](#) and [Chapter 4, “Application Planning for a Siebel Upgrade.”](#) To begin the process of upgrading your Siebel database, see [Chapter 5, “Preparing for Siebel Database Upgrade,”](#) then refer to each subsequent chapter.

For a more detailed view of the upgrade steps for each Siebel database environment, see the following topics:

- [“Process of Upgrading a Siebel Development Environment” on page 308](#)
- [“Process of Upgrading a Siebel Production Test Environment” on page 315](#)
- [“Process of Tuning Siebel Upgrade Performance” on page 320](#)
- [“Process of Upgrading a Siebel Production Environment” on page 324](#)

Figure 9 displays the basic tasks for upgrading the Siebel database environments.

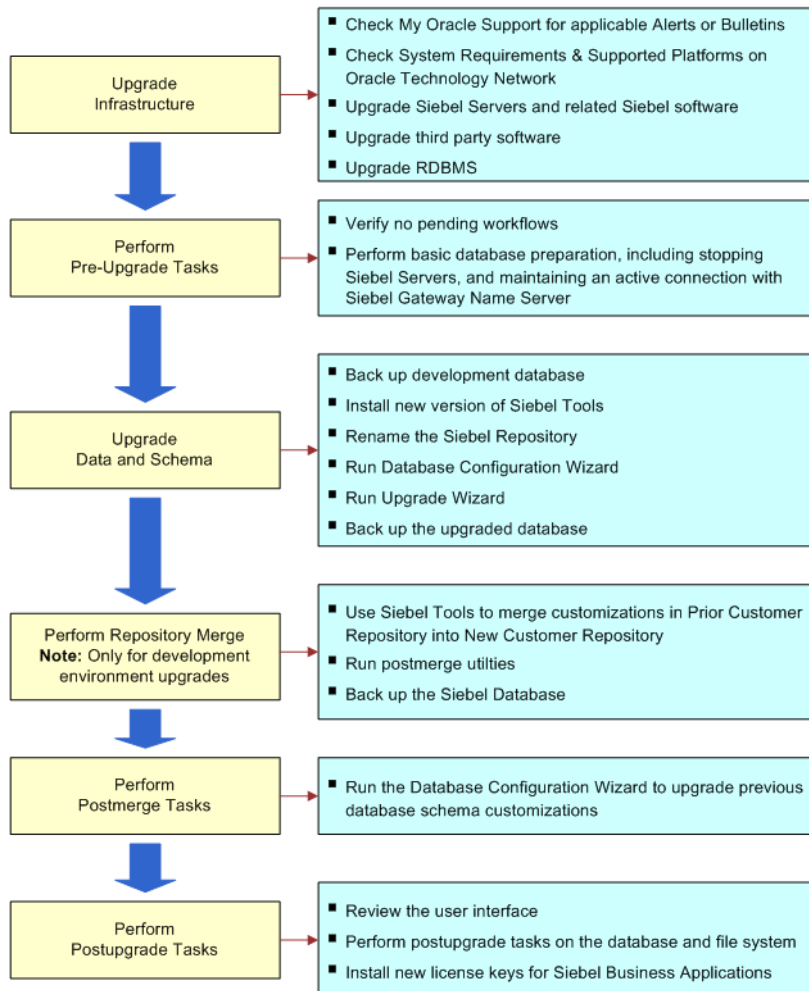


Figure 9. Upgrading the Siebel Database Environment

This roadmap includes the following topics:

- “Evaluating the Environment” on page 303
- “Planning the Upgrade” on page 303
- “Configuring the Environment” on page 304
- “Testing the Environment” on page 307
- “Implementing the Environment” on page 308

Evaluating the Environment

Whether you are beginning your upgrade with the development environment or the production test environment, it is advisable to begin by evaluating the process to determine resource allocation and other concerns. Table 43 lists information relevant to this phase of the upgrade.

Table 43. Evaluating the Environment

Upgrade Tasks	Where to Find Documentation	Dev?	Prod?
1 Check My Oracle Support for recently published articles regarding your upgrade.	My Oracle Support (http://support.oracle.com)	Yes	Yes
2 Check <i>Siebel Release Notes</i> on My Oracle Support.	My Oracle Support (http://support.oracle.com) See 557191.1 (Article ID).	Yes	Yes
3 Review hardware and third-party software requirements.	The Certifications tab on My Oracle Support.	Yes	Yes
4 If you need help evaluating your environment, then contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.	Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services	Yes	Yes

Planning the Upgrade

Review the *Siebel Database Upgrade Guide* to plan and further determine the scope of the upgrade process. Table 44 lists chapters in the *Siebel Database Upgrade Guide* to consider during the planning phase.

Table 44. Planning the Upgrade

Upgrade Tasks	Where to Find Documentation	Dev?	Prod?
1 Review the <i>Siebel Database Upgrade Guide</i> for new features in the current release.	Chapter 1, "What's New in This Release"	Yes	Yes
2 Review how to perform the Siebel database upgrade and how the upgrade works step-by-step.	Chapter 2, "Overview of Siebel Database Environments," and Chapter 3, "Siebel Database Upgrade Planning"	Yes	Yes

Table 44. Planning the Upgrade

Upgrade Tasks	Where to Find Documentation	Dev?	Prod?
3 Review Siebel database upgrade planning and about upgrade considerations for your Siebel user interface.	Chapter 3, "Siebel Database Upgrade Planning"	Yes	Yes
4 Review planning for Siebel Business Applications prior to upgrade.	Chapter 4, "Application Planning for a Siebel Upgrade"	Yes	Yes

Configuring the Environment

This is the most time-consuming and resource-intensive phase of the Siebel database upgrade. Many tasks in this phase are performed only during a development environment upgrade. Perform these tasks only if you are upgrading a development environment. All other tasks are performed during all upgrades. [Table 45](#) lists information relevant to this phase of the upgrade.

Table 45. Configuring the Environment

Upgrade Tasks	Where to Find Documentation	Dev?	Prod
1 Copy the ancestor repositories.	For details, see "Copying Ancestor Repositories" on page 130.	Yes	No
2 Upgrade the servers. Verify that you have met all requirements for the upgrade. You might be required to upgrade the Siebel Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE).	For more information on server upgrade and installation, see the <i>Siebel Installation Guide</i> for the operating system you are using. See also the Certifications tab on My Oracle Support.	Yes	Yes
3 Upgrade third-party software. You might need, for example, to upgrade operating system software.	See the Certifications tab on My Oracle Support.	Yes	Yes
4 Upgrade the RDBMS.	See the Certifications tab on My Oracle Support. For details on upgrading the RDBMS, see "About Upgrading Your RDBMS in the Siebel Environment" on page 35.	Yes	Yes

Table 45. Configuring the Environment

Upgrade Tasks	Where to Find Documentation	Dev?	Prod
<p>5 Perform any preupgrade tasks on your specific database.</p>	<p>For details on IBM DB2, see “Perform Preupgrade Tasks for IBM DB2” on page 310.</p> <p>For details on Oracle Database, see “Perform Preupgrade Tasks for Oracle Database” on page 310.</p> <p>For details on Microsoft SQL Server, see “Perform Preupgrade Tasks for Microsoft SQL Server” on page 310.</p>	Yes	Yes
<p>6 Preparing the Siebel database for an upgrade.</p> <p>Tasks include verifying that you have no pending workflows, stopping the Siebel Server, and so on.</p>	<p>For an overview, see “Perform Preupgrade Tasks for the Siebel Database” on page 310.</p> <p>For details on basic database preparation, see Chapter 5, “Preparing for Siebel Database Upgrade.”</p>	Yes	Yes
<p>7 Preparing Siebel Business Applications data for an upgrade.</p>	<p>See Chapter 9, “Preparing Siebel Application Data for Upgrade.”</p>	Yes	Yes
<p>8 Prepare your developers for the upgrade.</p> <p>Tasks include backing up development databases, ensuring that all developers’ projects are checked in and unlocked.</p>	<p>See “Prepare Developers for the Upgrade” on page 311.</p>	Yes	No
<p>9 Use the previous version of Siebel Tools to rename your Siebel Repository.</p>	<p>For an overview, see “Upgrade Siebel Database Schema (upgrep)” on page 311.</p> <p>For task details, see “Renaming the Siebel Repository” on page 129.</p>	Yes	No
<p>10 Upgrade the Siebel database schema.</p> <p>Tasks include running the Database Configuration Wizard (upgrep) and the Upgrade Wizard.</p>	<p>For an overview, see “Upgrade Siebel Database Schema (upgrep)” on page 311.</p> <p>For specific task details, including running the Database Configuration Utilities, see Chapter 10, “Upgrading the Siebel Database.”</p>	Yes	Yes
<p>11 Prepare for the repository merge.</p>	<p>For details, see “Perform Repository Merge” on page 312.</p>	Yes	No

Table 45. Configuring the Environment

Upgrade Tasks	Where to Find Documentation	Dev?	Prod
<p>12 Use the newly installed version of Siebel Tools to perform the repository merge.</p>	<p>For details, see “Perform Repository Merge” on page 312.</p>	Yes	No
<p>13 Upgrade custom database schema. Tasks include running the Database Configuration Wizard to upgrade previous database schema customizations, running the Upgrade Wizard, resolving errors with upgrade log files, and backing up the Siebel database.</p>	<p>For an overview, see “Upgrade Custom Database Schema (upgphys)” on page 313. For task details, see Chapter 10, “Upgrading the Siebel Database.”</p>	Yes	No
<p>14 Initialize and extract developers’ local database.</p>	<p>For an overview, see “Initialize and Extract Developers’ Local Databases” on page 313. For task details, see <i>Configuring Siebel Business Applications.</i></p>	Yes	No
<p>15 Review the user interface. Tasks include reviewing potential object property conflicts, noting inherited behavior, and performing other tasks on the Siebel user interface.</p>	<p>For an overview, see “Review the User Interface” on page 313. For task details, see Chapter 16, “Reviewing the Siebel User Interface.”</p>	Yes	Yes
<p>16 Create a new SRF file to assist in user interface testing.</p>	<p>For details, see <i>Using Siebel Tools.</i></p>	Yes	Yes
<p>17 Perform postmerge development tasks.</p>	<p>For an overview, see “Perform Postmerge Development Tasks” on page 314. For task details, see Chapter 14, “Siebel Postmerge Development Tasks.”</p>	Yes	No

Table 45. Configuring the Environment

Upgrade Tasks	Where to Find Documentation	Dev?	Prod
<p>18 Perform postupgrade tasks on database and file system.</p> <p>These tasks are done following a completed upgrade.</p>	<p>For an overview, see “Perform Postupgrade Tasks for Database and File System” on page 314.</p> <p>For task details, see Chapter 15, “Postupgrade Tasks for the Siebel Database.”</p>	Yes	Yes
<p>19 Perform postupgrade tasks for applications configuration.</p> <p>Tasks include installing activating and deploying any workflows, as well as performing other postupgrade configuration tasks.</p>	<p>For an overview, see “Perform Postupgrade Tasks for Applications Configuration” on page 320.</p> <p>For task details, see Chapter 17, “Postupgrade Tasks for Siebel Business Applications.”</p>	Yes	Yes

Testing the Environment

In this phase, perform system tests and prepare for the transition to the next database environment. [Table 46](#) lists information relevant to this phase of the upgrade.

Table 46. Testing the Environment

Upgrade Tasks	Where to Find Documentation	Dev?	Prod?
<p>1 Perform system tests.</p> <p>Use available test data to perform unit testing.</p>	<p>For an overview, see “Perform System Tests” on page 315.</p>	Yes	Yes
<p>2 Prepare for the transition to the next environment. If you upgraded a development environment, then you will move to the production test environment, if you upgraded the production test environment then you proceed to the production environment.</p> <p>Tasks include creating a new SRF file, and regenerating Siebel repository definition files, if necessary.</p>	<p>For an overview, see “Prepare for Transition to Production Test Environment” on page 315.</p> <p>For details on creating an SRF file, see <i>Developing and Deploying Siebel Business Applications</i>.</p> <p>For details on regenerating Siebel repository definition files, see “Regenerating the Siebel Repository Definition Files” on page 178.</p>	Yes	Yes

Implementing the Environment

Implementing your environment occurs when you have fully upgraded and tested your production test environment and then move to your live, production environment. [Table 47](#) lists information relevant to this phase of the upgrade.

NOTE: Perform these tasks only if you are transitioning from a production test to a production environment.

Table 47. Implementing the Environment

Upgrade Tasks	Where to Find Documentation	Dev?	Prod?
1 Apply the latest maintenance release (if any).	Check with your Oracle representative for any information on Siebel CRM maintenance releases which might be applicable to your release.	No	Yes
2 Perform an architecture and infrastructure review to ensure the production environment meets the requirements of the upgrade.	For help with an architecture and infrastructure review, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.	No	Yes

Process of Upgrading a Siebel Development Environment

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

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 when doing the upgrade.

NOTE: Links to other topics in this guide are provided throughout this chapter. These links are meant to provide additional information that cannot be provided in these brief overviews. You must, however, perform the upgrade in the proper order. For information on general planning, see this chapter as well as [Chapter 3, "Siebel Database Upgrade Planning"](#), and [Chapter 4, "Application Planning for a Siebel Upgrade."](#) To begin the process of upgrading your Siebel database, see [Chapter 5, "Preparing for Siebel Database Upgrade"](#), then refer to each subsequent chapter.

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

NOTE: If you are performing a development environment upgrade on IBM z/OS, then see *Siebel Database Upgrade Guide for DB2 for z/OS*.

Search for Articles on My Oracle Support

- Check My Oracle Support for recently published articles regarding the upgrade. For more information, see [“Naming Conventions Used in This Guide” on page 21](#).

Upgrade the Servers

Verify that you have identified all of the Siebel releases that are required for the upgrade. Along with the current release, Siebel Innovation Pack 2016, also install the latest Siebel Patchset release. For more information on server upgrade and installation, see the *Siebel Installation Guide* for the operating system you are using. See also the Certifications tab on My Oracle Support.

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 Siebel Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE).
For information on upgrading these Siebel Enterprise components, see *Siebel Installation Guide* for the operating system you are using.
- 2 Choose the option to install the Siebel Database Server along with the Siebel Gateway Name Server and Siebel Server.
- 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, then you can run a script to migrate configuration parameters to the upgraded Siebel Enterprise. (Do not perform this step for a migration installation, which is described in the *Siebel Installation Guide* for the operating system you are using.)
- 5 If you are upgrading to Siebel CRM versions 8.1 or 8.2, then copy the ancestor repositories. In Siebel CRM versions 8.1 and 8.2, the ancestor repository files are located in an Ancestor Repositories media directory that you access from Oracle Software Delivery Cloud. For more information, see [“Preparing for the Repository Merge” on page 167](#).

Upgrade Third-Party Software

- Upgrade third-party software as required due to dependencies on Oracle’s Siebel software or other installed software. For example, you might need to upgrade the following software:
 - Operating system software. Some database upgrades require newer versions of operating systems, such as AIX or Windows.

Upgrade the RDBMS

- If required, then upgrade the RDBMS version. Refer to the vendor’s documentation to perform the upgrade:
 - For information on supported RDBMS systems, see the Certifications tab on My Oracle Support.

- For information on how upgrading your RDBMS affects the upgrade process, see [“About Upgrading Your RDBMS in the Siebel Environment”](#) on page 35.

Perform Preupgrade Tasks for the Siebel Database

These steps apply to all database types.

- 1 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 2 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 3 Stop Siebel Servers, but make sure the Siebel Gateway Name Server is running.
NOTE: The Gateway Name Server must be running for the Database Configuration Wizard to proceed.
- 4 Perform the tasks in [Chapter 5, “Preparing for Siebel Database Upgrade.”](#)

Perform Preupgrade Tasks for IBM DB2

- 1 Perform the tasks in [Chapter 6, “Preparing an IBM DB2 Database for a Siebel Upgrade.”](#)
- 2 If you are disabling archive logging, then allow time for a cold backup when reenabling archive logging.
- 3 Apply runstats to all tables, reorganize any that are out of cluster, or that are clustered less than 90%, then execute runstats on those tables again. This will improve performance and reduce the number of tasks that will need to be done after the upgrade is complete.

Perform Preupgrade Tasks for Oracle Database

- 1 Perform the tasks in [Chapter 7, “Preparing an Oracle Database for a Siebel Upgrade.”](#)
- 2 Run the Oracle Analyze command on the Siebel database. Highly fragmented indexes can cause the upgrade to fail.

Perform Preupgrade Tasks for Microsoft SQL Server

- 1 Perform the tasks in [Chapter 8, “Preparing a Microsoft SQL Server Database for a Siebel Upgrade.”](#)
- 2 Run Microsoft SQL Server statistics. This will improve upgrade performance.

Perform Preupgrade Tasks for Application Data

- 1 Perform the tasks in 477519.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 521. The following table shows the applicability of the tasks in the article.

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

- 2 Perform the tasks in the My Oracle Support article listed in the following table.

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

- 3 Perform the tasks in [Chapter 9, "Preparing Siebel Application Data for Upgrade."](#)

Prepare Developers for the Upgrade

To prepare developers for the upgrade, do the following.

- 1 Back up the development database.
- 2 Verify that all developers have checked in their projects and that all projects are unlocked.
- 3 Verify that all developers have disconnected from the database. The only open account must be the account that is performing the upgrade.
- 4 Install the new version of Siebel Tools on development workstations.

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

Upgrade Siebel Database Schema (upgrep)

- 1 (Optional) ["Changing the Siebel Database Configuration Utilities Language"](#) on page 132.
- 2 Run the Database Configuration Wizard:
 - ["Preparing to Run the Siebel Database Configuration Wizard"](#) on page 132.
 - ["Running the Siebel Database Configuration Wizard on Windows"](#) on page 136.
 - ["Running the Siebel Database Configuration Wizard on UNIX"](#) on page 141.

Choose the following settings:

- **Environment Type:** Development
- **Upgrade Options:** Upgrade Siebel Database Schema (upgrep)
- 3 Edit generated SQL files as required by articles on My Oracle Support, and by the applicable version of *Siebel Release Notes* on My Oracle Support.
- 4 Resume the upgrade: [“Starting the Siebel Upgrade Wizard” on page 145.](#)
- 5 Review the upgrade logs and resolve errors:
 - [“Reviewing Siebel Upgrade Log Files for Errors” on page 161.](#)
- 6 If the upgrade contains unacceptable errors, then do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c If errors occurred because you entered incorrect information in the Database Configuration Wizard, then see [“Regenerating SQL Files for a Siebel Upgrade” on page 155.](#)
 - d Rerun the Database Configuration Wizard.
- 7 Back up the upgraded database.

Prepare for Repository Merge

- 1 [“Preparing for the Repository Merge” on page 167.](#)

NOTE: If you are upgrading from Siebel CRM version 8.1.1.x (SIA repository), version 8.2.2.x, or version 15.x to version 16.0, then use [“Process of Meeting the Requirements for an Incremental Upgrade to Siebel Innovation Pack 2016, Using the Incremental Repository Merge” on page 187.](#)
- 2 [“Migrating Siebel Repository Objects to the Standard User Interface” on page 180.](#)
- 3 Set the Upgrade Ancestor property for copied objects. See [“Configuring Siebel Repository Objects to Inherit Upgrade Behavior” on page 168.](#)

Perform Repository Merge

- 1 [“Performing a Siebel Repository Merge” on page 169.](#)

NOTE: If you are upgrading from Siebel CRM version 8.1.1.x (SIA repository), version 8.2.2.x, or version 15.x to version 16.0, then use [“Process of Meeting the Requirements for an Incremental Upgrade to Siebel Innovation Pack 2016, Using the Incremental Repository Merge” on page 187.](#)
- 2 [“Reviewing the Siebel Repository Merge Log Files” on page 171.](#)
- 3 If the repository merge contains unacceptable errors, then do the following:
 - a Restore the backup of the upgraded database.
 - b Correct the errors.
 - c Rerun the repository merge.

- 4 ["Running the Siebel Postmerge Utilities"](#) on page 185.
- 5 ["Generating Siebel Enterprise Integration Manager Temporary Columns"](#) on page 174.
- 6 (Optional) Configure case insensitive queries in the New Customer Repository. See ["Running the Siebel Case Insensitivity Wizard"](#) on page 149.
- 7 Back up the Siebel database.

Upgrade Custom Database Schema (upgphys)

- 1 Run the Database Configuration Wizard:
 - ["Preparing to Run the Siebel Database Configuration Wizard"](#) on page 132.
 - ["Running the Siebel Database Configuration Wizard on Windows"](#) on page 136.
 - ["Running the Siebel Database Configuration Wizard on UNIX"](#) on page 141.Choose the following settings:
 - **Upgrade Options:** Upgrade Custom Database Schema (upgphys)
 - **Environment Type:** Development
- 2 Run the Upgrade Wizard: ["Starting the Siebel Upgrade Wizard"](#) on page 145.
- 3 Review the upgrade log files and resolve errors: ["Reviewing Siebel Upgrade Log Files for Errors"](#) on page 161.
- 4 If the upgrade contains unacceptable errors, then 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.
- 5 ["Manually Archiving Siebel Upgrade Log Files"](#) on page 165.
- 6 Back up the upgraded database.

Initialize and Extract Developers' Local Databases

- 1 Install the new release of Siebel Tools on developers' computers.
- 2 Initialize and extract the local database on development computers. For more information, see *Configuring Siebel Business Applications*.

Review the User Interface

- 1 ["Reviewing Siebel Repository Object Property Conflicts"](#) on page 174.
- 2 Review the following topics on user interface upgrade before proceeding:
 - ["About Inheriting Upgrade Behavior in a Siebel Upgrade"](#) on page 57
 - ["About the Siebel Postmerge Utilities"](#) on page 59

- For ICL upgrades, see [Appendix C, “Siebel Incorporate Custom Layout Upgrade Option.”](#)
- 3 If you performed an ICL upgrade and chose Label on Top, then revise the cascading style sheet. See [“Deleting Unneeded Siebel Repository Files”](#) on page 180.
- 4 Create a new SRF file to assist in user interface testing. For information on creating an SRF file, see *Developing and Deploying Siebel Business Applications*.
- 5 If you customized style sheet or Web template files in the previous release, then reimplement those customizations in the new release as desired.

Carefully review the user interface in the new release before reimplementing customizations to these files.
- 6 Perform the tasks in [Chapter 16, “Reviewing the Siebel User Interface.”](#)
- 7 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 topic in *Configuring Siebel Business Applications* and convert the remaining flow-based applets as required.

Perform Postmerge Development Tasks

- 1 Perform the tasks in [Chapter 14, “Siebel Postmerge Development Tasks.”](#)
- 2 Resolve any business component and join conflicts.
- 3 [“Deleting Unneeded Siebel Repository Files”](#) on page 180.

Perform Postupgrade Tasks for Database and File System

- 1 Perform the tasks in [Chapter 15, “Postupgrade Tasks for the Siebel Database.”](#)
- 2 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.
- 3 If you exported data from interface tables before the upgrade, then review the database and import the data as desired. See *Siebel Enterprise Integration Manager Administration Guide*.
- 4 Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide*.
- 5 Run database statistics.

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

Perform Postupgrade Tasks for Applications Configuration

- 1 If applicable, then review the results of the Person and Organization merge. Make configuration changes as required.
- 2 Perform the tasks in [Chapter 17, “Postupgrade Tasks for Siebel Business Applications.”](#)
- 3 Verify the function of interfaces in integrated applications.

- 4 Activate and deploy workflows.
- 5 If you have set up integrations for transferring data to or from third-party applications using Siebel Enterprise Application Integration, then verify the integrations are configured correctly. For information on EAI, see *Overview: Siebel Enterprise Application Integration*.
- 6 If you have used Siebel Enterprise Integration Manager to set up batch processing jobs, then verify Siebel Enterprise Integration Manager is configured correctly. For information on Siebel Enterprise Integration Manager, see *Siebel Enterprise Integration Manager Administration Guide*.

Perform System Tests

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

Prepare for Transition to Production Test Environment

- 1 Create a new SRF file. For information on creating an SRF file, see *Developing and Deploying Siebel Business Applications*.
- 2 If you revised the repository after running upgphys, then you must regenerate the repository definition files. See [“Regenerating the Siebel Repository Definition Files” on page 178](#).

Process of Upgrading a Siebel Production Test Environment

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

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

NOTE: Links to other topics in this guide are provided throughout this chapter. These links are meant to provide additional information that cannot be provided in these brief overviews. You must, however, perform the upgrade in the proper order. For information on general planning, see this chapter as well as [Chapter 3, “Siebel Database Upgrade Planning,”](#) and [Chapter 4, “Application Planning for a Siebel Upgrade.”](#) To begin the process of upgrading your Siebel database, see [Chapter 5, “Preparing for Siebel Database Upgrade,”](#) then refer to each subsequent chapter after that.

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

NOTE: If you are performing a Production Test environment upgrade on IBM z/OS, then see *Siebel Database Upgrade Guide for DB2 for z/OS*.

Search for Articles on My Oracle Support

- Check My Oracle Support for recently published articles regarding the upgrade. For more information, see [“Naming Conventions Used in This Guide” on page 21](#).

Upgrade the Servers

Verify that you have identified all of the Siebel releases that are required for the upgrade. Along with the current release, Siebel Innovation Pack 2016, also install the latest Siebel Patchset release. For more information on server upgrade and installation, see the *Siebel Installation Guide* for the operating system you are using. See also the Certifications tab on My Oracle Support.

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 Siebel 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.

NOTE: You need only to install a new Siebel Server instance if you are using a different computer from that on which you performed the Production Test Upgrade. If you are using the same Siebel Server installation as for the Production Test upgrade, then the existing files can be leveraged instead.

- 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, then you can run a script to migrate configuration parameters to the upgraded Siebel Enterprise. (Do not perform this step for a migration installation, which is described in the *Siebel Installation Guide* for the operating system you are using.)

Upgrade Third-Party Software

- Upgrade third-party software as required due to dependencies on Oracle’s Siebel software or other installed software. For example, you might need to upgrade the following software:

Operating system software. Some database upgrades require newer versions of AIX or Windows.

Upgrade the RDBMS

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

- For information on supported RDBMS systems, see the Certifications tab on My Oracle Support.

- For information on how upgrading your RDBMS affects the upgrade process, see [“About Upgrading Your RDBMS in the Siebel Environment”](#) on page 35.

Perform Preupgrade Tasks for the Siebel Database

These steps apply to all database types.

- 1 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 2 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 3 Perform the tasks in [Chapter 5, “Preparing for Siebel Database Upgrade.”](#)

Perform Preupgrade Tasks for IBM DB2

- 1 Perform the tasks in [Chapter 6, “Preparing an IBM DB2 Database for a Siebel Upgrade.”](#)
- 2 Make sure that runstats is current in your environment and that tables with clustering indexes are not out of cluster. Reorganize any tables that are out of cluster or that are clustered less than 90% before the upgrade if possible.

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

Perform Preupgrade Tasks for Oracle Databases

- 1 Perform the tasks in [Chapter 7, “Preparing an Oracle Database for a Siebel Upgrade.”](#)
- 2 Run the Oracle Analyze command on the Siebel database. Highly fragmented indexes can cause the upgrade to fail.

Perform Preupgrade Tasks for Microsoft SQL Server

- 1 Perform the tasks in [Chapter 8, “Preparing a Microsoft SQL Server Database for a Siebel Upgrade.”](#)
- 2 Run Microsoft SQL Server statistics. This will improve upgrade performance.

Perform Preupgrade Tasks for Application Data

- 1 Perform the tasks in 477519.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 521. The following table shows the applicability of the tasks in the article.

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

- 2 Perform the tasks in [Chapter 9, "Preparing Siebel Application Data for Upgrade."](#)

Prepare the Siebel Database for Upgrade

- 1 Refer to 477772.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 586, and includes instructions on how to set up the Siebel database and Siebel Servers in the production test environment.
- 2 Stop the Siebel Servers and the Siebel Gateway Name Server.
- 3 Close all database connections. The only database connection must be the account performing the upgrade.
- 4 Copy application files to the environment:
 - Custom SRF file.
 - Reports files.
- 5 Copy the custrep.dat and schema.ddl files to the environment. See ["Moving the Siebel Repository Files" on page 149](#).

If you have made changes to the Siebel repository since performing the development upgphys, then you must regenerate the schema.ddl and custrep.dat files. See ["Regenerating the Siebel Repository Definition Files" on page 178](#).

- 6 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.

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

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

Upgrade the Siebel Database (upgrep and upgphys)

- 1 Run the Database Configuration Wizard:
 - [“Preparing to Run the Siebel Database Configuration Wizard”](#) on page 132.
 - [“Running the Siebel Database Configuration Wizard on Windows”](#) on page 136.
 - [“Running the Siebel Database Configuration Wizard on UNIX”](#) on page 141.Choose the following settings:
 - **Environment Type:** Production
 - **Upgrade Options:** Upgrade Siebel Database Schema (upgrep and upgphys)See [“Performing a Production Test or Production Environment Migration from Siebel CRM Version 8.1.1.x \(SIA Repository\)”](#) on page 240.
- 2 Refer to 475427.1 (Article ID) on My Oracle Support for any instructions specific to your upgrade scenario. This document was previously published as Siebel Alert 1322.
- 3 Resume the upgrade. See [“Starting the Siebel Upgrade Wizard”](#) on page 145.
- 4 Review the upgrade logs and resolve errors. See [“Reviewing Siebel Upgrade Log Files for Errors”](#) on page 161.
- 5 If the upgrade contains unacceptable errors, then do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c If errors occurred because you entered incorrect information in the Database Configuration Wizard, then see [“Regenerating SQL Files for a Siebel Upgrade”](#) on page 155.
 - d Rerun the Database Configuration Wizard.
- 6 See [“Manually Archiving Siebel Upgrade Log Files”](#) on page 165.
- 7 Back up the upgraded database.

Perform Postupgrade Tasks for Database and File System

- 1 Perform the tasks in [Chapter 15, “Postupgrade Tasks for the Siebel Database.”](#)
- 2 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.
- 3 If you exported data from interface tables before the upgrade, then review the database and import the data as desired.
- 4 Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide*.
- 5 Run database statistics.

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

Perform Postupgrade Tasks for Applications Configuration

- 1 Perform the tasks in [Chapter 17, "Postupgrade Tasks for Siebel Business Applications."](#)
- 2 Verify the function of interfaces in integrated applications.
- 3 Activate and deploy workflows.
- 4 If you have set up integrations for transferring data to or from third-party applications using Siebel Enterprise Application Integration, then verify that the integrations are configured correctly. For information on EAI, see *Overview: Siebel Enterprise Application Integration*.
- 5 If you have used Siebel Enterprise Integration Manager to set up batch processing jobs, then verify Siebel Enterprise Integration Manager is configured correctly. For information on Siebel Enterprise Integration Manager, see *Siebel Enterprise Integration Manager Administration Guide*.
- 6 If you customized style sheet or Web template files in the previous release, then reimplement those customizations in the new release as desired.

Carefully review the user interface in the new release before reimplementing customizations to these files.

Perform System Tests

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

Process of Tuning Siebel Upgrade Performance

Platforms: Windows and UNIX only.

NOTE: This process is optional.

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

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

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

NOTE: Links to other topics in this guide are provided throughout this chapter. These links are meant to provide additional information that cannot be provided in these brief overviews. You must, however, perform the upgrade in the proper order. For information on general planning, see this chapter as well as [Chapter 3, “Siebel Database Upgrade Planning,”](#) and [Chapter 4, “Application Planning for a Siebel Upgrade.”](#) To begin the process of upgrading your Siebel database, see [Chapter 5, “Preparing for Siebel Database Upgrade,”](#) then refer to each subsequent chapter.

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

- [“About Siebel Additive Schema Changes Mode” on page 46](#)
- **478308.1 (Article ID) on My Oracle Support.** This document was previously published as Siebel Technical Note 616. This article describes the strategies for minimizing production environment downtime during an upgrade. The steps that follow are intended primarily for use with the baseline guidelines described in 478308.1 (Article ID).
- [Chapter 18, “Tuning the Siebel Upgrade Files.”](#) This chapter provides detailed information on how to use Upgrade Tuner.

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

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

Set Up the Target Database

- 1 Back up and remove the upgraded production test database.
This preserves seed data and metadata changes you have made that must be migrated to the production environment. This database is called the **production-test final database**.
- 2 In the production test environment, install a recent backup of your production database.
This database has not been upgraded and is called the **target database**. You will use it to perform test upgrades as part of tuning upgrade performance.
- 3 Define an ODBC connection to the target database.
- 4 Verify that the target database and RDBMS server are configured for optimum upgrade performance:
 - [Chapter 6, “Preparing an IBM DB2 Database for a Siebel Upgrade”](#)
 - [Chapter 7, “Preparing an Oracle Database for a Siebel Upgrade”](#)
 - [Chapter 8, “Preparing a Microsoft SQL Server Database for a Siebel Upgrade”](#)
- 5 [“Preparing Siebel Tables and Views for Upgrade” on page 83.](#)
- 6 Run statistics on the target database. This optimizes query performance.

- 7 **UNIX.** [“Securing AIX Memory Allocation Segment Space for the Siebel Database” on page 86.](#)
- 8 Perform the tasks in [Chapter 9, “Preparing Siebel Application Data for Upgrade.”](#)

Test Additive Schema Changes

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

Perform these steps on the target database.

- 1 Review [“About Siebel Additive Schema Changes Mode” on page 46.](#)
- 2 (Optional) [“Changing the Siebel Database Configuration Utilities Language” on page 132.](#)
- 3 In the production test environment, run the Database Configuration Wizard:
 - [“Preparing to Run the Siebel Database Configuration Wizard” on page 132](#)
 - [“Running the Siebel Database Configuration Wizard on Windows” on page 136](#)
 - [“Running the Siebel Database Configuration Wizard on UNIX” on page 141](#)

- 4 Choose Apply Additive Schema Changes mode.

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

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

- 5 **UNIX.** Run the Siebel Upgrade Wizard.

See [“Starting the Siebel Upgrade Wizard” on page 145.](#)

Specify `master_additive_gen.ucf` as the input file.

The Upgrade Wizard generates the `schema_additive.sql` script. It does not run the script against the target database.

- 6 Run the SQL script against the target database.

See [“Applying Siebel Additive Schema Changes” on page 154.](#)
- 7 Thoroughly review the operation of all applications. Verify that applying the additive schema changes has not adversely affected the operation of all applications.
- 8 If applying additive schema changes adversely affects the operation of the applications, then do the following:
 - a For help diagnosing the cause of the problem and help in taking corrective action create a service request (SR) on My Oracle Support. Alternatively, you can phone Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers remain the same and are listed on My Oracle Support.
 - b Restore the target database from backup before proceeding with further upgrade tuning.
 - c Do not use the additive schema changes feature during your production upgrade.

- 9 If applications function normally, then continue with upgrade tuning. Consider using the additive schema changes feature during your production upgrade.

Upgrade the Target Database (upgrep and upgphys)

- 1 (Optional) [“Changing the Siebel Database Configuration Utilities Language” on page 132.](#)
- 2 In the production test environment, run the Database Configuration Wizard:
 - [“Preparing to Run the Siebel Database Configuration Wizard” on page 132](#)
 - [“Running the Siebel Database Configuration Wizard on Windows” on page 136](#)
 - [“Running the Siebel Database Configuration Wizard on UNIX” on page 141](#)

Choose the following settings:

- **Upgrade Options:** Upgrade Siebel Database Schema (upgrep and upgphys)
 - **Environment Type:** Production
- 3 In the production test environment, run Siebel Upgrade Wizard.
See [“Starting the Siebel Upgrade Wizard” on page 145.](#)
Note the time required to upgrade the database.
 - 4 Review the upgrade log files for errors: [“Reviewing Siebel Upgrade Log Files for Errors” on page 161.](#)
 - 5 If the upgrade contains errors that prevented completion or adversely affected performance, then correct the errors and rerun the upgrade.
 - 6 [“Manually Archiving Siebel Upgrade Log Files” on page 165.](#)

Tune the Upgrade Files

- 1 Evaluate upgrade performance, particularly the time required to complete the upgrade.
- 2 Do one of the following:
 - If the time required to complete the upgrade is acceptable, then no further tuning is needed. Perform the steps in [“Restore the Production-Test Final Database” on page 324.](#)
 - If the time required to complete the upgrade is too long, then perform the remaining steps in this topic to continue tuning upgrade performance.
 - If the time required to complete the upgrade is too long and you cannot tune further, then contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle’s Application Expert Services to apply advanced tuning.
- 3 Carefully review target database and RDBMS server configuration. Adjust as needed to further improve upgrade performance.
- 4 Run Upgrade Tuner to tune the upgrade files:
 - **UNIX.** [“Transferring UNIX Files for Use by Siebel Upgrade Tuner” on page 283.](#)
 - [“Starting and Stopping Siebel Upgrade Tuner” on page 275.](#)

- [“Managing Parallel Threads Using Siebel Upgrade Tuner” on page 277.](#)
- [“Managing Zero-Row SQL Commands Using Siebel Upgrade Tuner” on page 280.](#)

To roll back changes you have made in previous Upgrade Tuner sessions, see [“Rolling Back Siebel Upgrade Tuner Changes” on page 286.](#)

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.

- 1 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.
- 2 In the production test environment, apply additive schema changes again as desired.
- 3 Perform another test upgrade and evaluate upgrade performance.
- 4 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.

- 1 In the production test environment, delete and remove the target database.
- 2 In the production test environment, restore the production-test final database from backup.

Process of Upgrading a Siebel Production Environment

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

To perform this process, you must be able to execute ODBC commands on the production database from within the production environment. For more information, see [“About the Siebel Database Configuration Utilities” on page 37.](#) For help executing ODBC commands within the production test environment, create a service request (SR) on My Oracle Support.

NOTE: Links to other topics in this guide are provided throughout this chapter. These links are meant to provide additional information that cannot be provided in these brief overviews. You must, however, perform the upgrade in the proper order. For information on general planning, see this chapter as well as [Chapter 3, “Siebel Database Upgrade Planning,”](#) and [Chapter 4, “Application Planning for a Siebel Upgrade.”](#) To begin the process of upgrading your Siebel database, see [Chapter 5, “Preparing for Siebel Database Upgrade,”](#) then refer to each subsequent chapter.

This topic lists the tasks required to transition your production test environment to production. Print this topic and use it as a checklist when doing the upgrade. Complete the steps in the order shown.

NOTE: If you are performing a production environment upgrade on IBM z/OS, then see *Siebel Database Upgrade Guide for DB2 for z/OS.*

Search for Articles on My Oracle Support

- Check My Oracle Support for recently published articles regarding the upgrade. For more information, see [“Naming Conventions Used in This Guide” on page 21](#).

Upgrade the Servers

Verify that you have identified all of the Siebel CRM releases that are required for the upgrade. Along with the current release, Siebel Innovation Pack 2016, also install the latest Siebel Patchset release. For more information on server upgrade and installation, see the *Siebel Installation Guide* for the operating system you are using. See also the Certifications tab on My Oracle Support.

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 Siebel 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 the 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, then you can run a script to migrate configuration parameters to the upgraded Siebel Enterprise. (Do not perform this step for a migration installation, which is described in the *Siebel Installation Guide* for the operating system you are using.)

Upgrade Third-Party Software

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

Apply Additive Schema Changes

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

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

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

- 1 Review [“About Siebel Additive Schema Changes Mode” on page 46](#).

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

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

- 3 (Optional) [“Changing the Siebel Database Configuration Utilities Language” on page 132.](#)

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

- [“Preparing to Run the Siebel Database Configuration Wizard” on page 132](#)
- [“Running the Siebel Database Configuration Wizard on Windows” on page 136](#)
- [“Running the Siebel Database Configuration Wizard on UNIX” on page 141](#)

- 5 Choose Apply Additive Schema Changes mode.

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

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

- 6 UNIX. Run the Siebel Upgrade Wizard.

See [“Starting the Siebel Upgrade Wizard” on page 145.](#)

Specify `master_additive_gen.ucf` as the input file.

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

- 7 Run the SQL script against the production database.

See [“Applying Siebel Additive Schema Changes” on page 154.](#)

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

- 9 If you are applying additive schema changes adversely affects the operation of the application, then do the following:

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

Upgrade the RDBMS

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

- For information on supported RDBMS systems, see the Certifications tab on My Oracle Support.
- For information on how upgrading your RDBMS affects the upgrade process, see [“About Upgrading Your RDBMS in the Siebel Environment” on page 35.](#)

Perform Preupgrade Tasks for the Siebel Database

These steps apply to all database types.

- 1 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 2 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 3 Stop the Siebel Servers.
- 4 Verify that the Siebel Gateway Name Server is still running.
- 5 Stop all other Siebel Servers.
- 6 Close all database connections. The only open connection must be the account performing the upgrade.
- 7 Perform the tasks in [Chapter 5, "Preparing for Siebel Database Upgrade."](#)

Perform Preupgrade Tasks for IBM DB2

- 1 Perform the tasks in [Chapter 6, "Preparing an IBM DB2 Database for a Siebel Upgrade."](#)
- 2 Make sure that all transactional tables with clustering indexes are at least 90% clustered and stats are current.

NOTE: You might be required to perform runstats on a small number of transactional tables and indexes following a successful upgrade. All Siebel Enterprise Integration Manager tables will have been replaced and might need statistics gathered.

Perform Preupgrade Tasks for Oracle Database

- 1 Perform the tasks in [Chapter 7, "Preparing an Oracle Database for a Siebel Upgrade."](#)
- 2 Run the Oracle Analyze command on the Siebel database. Highly fragmented indexes can cause the upgrade to fail.

Perform Preupgrade Tasks for Microsoft SQL Server

- 1 Perform the tasks in [Chapter 8, "Preparing a Microsoft SQL Server Database for a Siebel Upgrade."](#)
- 2 Run Microsoft SQL Server statistics. This will improve upgrade performance.

Perform Preupgrade Tasks for Application Data

- 1 Perform the tasks in 477519.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 521. The following table shows the applicability of the tasks in the article.

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

- 2 Perform the tasks in [Chapter 9, “Preparing Siebel Application Data for Upgrade.”](#)

Perform Preupgrade Tasks for the User Interface

- Copy application files to the environment:
 - Custom SRF file.
 - Reports files.

Upgrade the Siebel Database (upgrep and upgphys)

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

- Run the Database Configuration Wizard in the production environment.
- In the utilities, enter the information for the production environment instead of the production test environment. For example, you enter the ODBC connection for the production environment.

This information configures the driver file to run against the production database rather than the production test database. It also configures the driver file to use the upgrade SQL files you generated for the production test upgrade.
- Run the Upgrade Wizard. The Upgrade Wizard uses the SQL files in the production test environment to upgrade the database in the production environment.

This approach has several advantages:

- You do not have to generate upgrade SQL files in the production environment, and then manually transfer the customizations to them from the production test environment.
- You do not lose any changes to the SQL files that were made by Siebel Upgrade Tuner in the production test environment.
- You do not have to run the Database Configuration Wizard in Prepare for Production mode again.
- With some exceptions, you do not have to perform again the database-related configuration tasks required by *Siebel Release Notes* on My Oracle Support or Articles on My Oracle Support.

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

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

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

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

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

- 4 (Optional) [“Changing the Siebel Database Configuration Utilities Language” on page 132.](#)
- 5 Run the Database Configuration Wizard:
 - [“Preparing to Run the Siebel Database Configuration Wizard” on page 132](#)
 - [“Running the Siebel Database Configuration Wizard on Windows” on page 136](#)
 - [“Running the Siebel Database Configuration Wizard on UNIX” on page 141](#)
 - a Choose the following settings when you run the utility:
 - **Upgrade Options.** Upgrade Siebel Database Schema (upgrep and upgphys)
 - **Environment Type.** Production
 - b Enter the information for the production environment instead of the production test environment.
 - c Enter the name of the ODBC for connecting to the production database.
 - d When prompted whether you want to run the Siebel Upgrade Wizard, answer No and exit.

This updates the master UCF file with the production environment configuration. When you run the Siebel Upgrade Wizard, the SQL commands will be executed on the production environment database.
- 6 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, then do not continue. Instead, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle’s Application Expert Services.
- 7 Perform the production database upgrade. See [“Starting the Siebel Upgrade Wizard” on page 145.](#)

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, then these changes are included.

- 8 Review the upgrade logs and resolve errors: [“Reviewing Siebel Upgrade Log Files for Errors” on page 161.](#)
- 9 If the upgrade contains unacceptable errors, then do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Siebel Upgrade Wizard.
- 10 [“Manually Archiving Siebel Upgrade Log Files” on page 165.](#)
- 11 Use the Siebel Application Deployment Manager to migrate administrative data such as LOVs and responsibilities from production test to production.
- 12 Back up the upgraded production environment database.

Perform Postupgrade Tasks for Database and File System

- 1 Perform the tasks in [Chapter 15, “Postupgrade Tasks for the Siebel Database.”](#)
- 2 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.
- 3 If you exported data from interface tables before the upgrade, then review the database and import the data as desired.
- 4 Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide*.
- 5 Run database statistics.

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

Perform Postupgrade Tasks for Applications Configuration

- 1 If applicable, review the results of the Person and Organization merge. Make configuration changes as required.
- 2 Perform the tasks in [Chapter 17, “Postupgrade Tasks for Siebel Business Applications.”](#)
- 3 Verify the function of interfaces in integrated applications.
- 4 Activate and deploy workflows.
- 5 If you have set up integrations for transferring data to or from third-party applications using Siebel Enterprise Application Integration, then verify that the integrations are configured correctly. For information on Siebel EAI, see *Overview: Siebel Enterprise Application Integration*.
- 6 If you have used Siebel Enterprise Integration Manager to set up batch processing jobs, then verify Siebel Enterprise Integration Manager is configured correctly. For information on Siebel Enterprise Integration Manager, see *Siebel Enterprise Integration Manager Administration Guide*.
- 7 If you customized the style sheet or Web template files in the previous release, then reimplement those customizations in the new release as desired.

Perform System Tests

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

C

Siebel Incorporate Custom Layout Upgrade Option

This appendix describes the Siebel Incorporate Custom Layout Upgrade Option. It includes the following topic:

- [“About the Siebel Incorporate Custom Layout Upgrade Option” on page 333](#)

About the Siebel Incorporate Custom Layout Upgrade Option

Upgrades from: Siebel 7.x using ICL.

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

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

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

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

Overview of Incorporate Custom Layout

If you are upgrading from Siebel CRM version 7.x or later, then 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, then a regular merge adds the control to Applet-A in the New Customer Repository.

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

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

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

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

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

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

Who Should Use ICL?

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

Use the following factors to assess whether to use ICL:

- ICL can be used only for upgrades from Siebel CRM version 7.x onward.
- If your user interface 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 user interface objects that specify an upgrade ancestor, then ICL is not recommended. This is because ICL ignores the Upgrade Ancestor property. For user interface objects, if the configuration of an upgrade ancestor changes in the new release, then 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, then ICL might not be a good choice.

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

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

UI Objects That ICL Affects

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

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

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

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

[Step 48 on page 336](#) compares a regular merge and an ICL merge. The columns list the status of a repository object:

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

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

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

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

Exceptions listed in [Table 48](#) are described in a section following the table.

Table 48. 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 "Table Exceptions" on page 336 .
Standard	Standard	Standard, Inactive	Standard, Inactive	Standard, Inactive
Standard	Customized	Standard	Customized	Customized
Standard	Customized	Standard, Inactive	Customized, Inactive	Customized, Inactive
Standard	Customized	Revised	Revised (conflict) See "Table Exceptions" on page 336 .	Customized
Not applicable	New	Not applicable	New	New
Not applicable	Not applicable	New	New	New
Standard	Deleted	Standard	Standard	Standard
Standard	Deleted	Standard, Inactive	Standard, or Inactive	Standard, or Inactive
Standard	Deleted	Revised	Revised	Revised

Table Exceptions

Exceptions listed in [Table 48 on page 336](#) are as follows:

- In an ICL merge, user interface navigation changes at Siebel CRM version 7.7 are implemented in screens copied-in from the Prior Customer Repository.

- A conflict exists when an object is different in all three repositories. The merge process typically resolves the conflict in favor of the New Siebel Repository. After the merge, you can review conflicts and change the resolution.

Some UI Objects Are Not Eligible for ICL Handling

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

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

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

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

Upgrade Behavior Property

An object property called Upgrade Behavior determines whether a UI object is preservable:

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

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

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

ICL Upgrade Path

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

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

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

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

- For upgrades from Siebel X and later, the Preserve status is effective, and the object is eligible for ICL handling. Customizations will be preserved.

For example:

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

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

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

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

How ICL Affects the Overall UI

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

Screens

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

Views

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

Applets

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

- When a Web template is used for an applet. Exception: for upgrades prior to Siebel CRM version 7.7, the postmerge utilities provide a new Web template for MVG applets that the postmerge utilities have converted to shuttle-enabled.

- All Web template item properties are preserved:
 - All control types are preserved.
 - The controls that are visible are preserved.
 - No new controls from the New Siebel Repository are added or deleted. Exception: for upgrades prior to Siebel CRM version 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 CRM version 7.7, many flow-based form applets are converted to grid-based.
- Applet Drilldowns and their child objects are preserved:
 - Drilldown properties are preserved.
 - The columns that have drilldowns is preserved
 - The destination views of drilldowns are preserved

The following features are not preserved:

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

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

- Siebel CRM version 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 CRM version 7.7, MVG dialog boxes display only an OK button to close them. After an ICL merge, MVG dialog box applets will have both OK and Cancel buttons.
- Siebel CRM version 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 CRM version 7.7, applets have three default buttons: New, Delete, and Query. If you select an ICL merge, then applets will keep their existing default buttons.
- In Siebel CRM version 7.7, the Reset button was removed from applets. If you select an ICL merge, then applets keep the Reset button.

Navigation

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

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

How ICL Handles Deleted and Obsolete Objects

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

In a regular merge, if an object is obsolete (inactive) in the New Siebel Repository, then 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, then they are handled as follows:

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

ICL and the Upgrade Ancestor Property

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

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

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

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

How an ICL Merge Works

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

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

Additional Merge Step

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

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

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

Postmerge Utilities and Grid-Based Applets

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

- At Siebel CRM version 7.7, many form applets in the New Siebel Repository that were formerly flow-based are grid-based. After an ICL merge, flow-based form applets from the Prior Customer Repository will replace some of these grid-based applets in the New Customer Repository, particularly for upgrades from releases prior to Siebel CRM version 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 CRM version 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 elements 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, then you must reimplement your customizations in the new Web template files.

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

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

These Web template files support the UI layout of the release you are upgrading from. They are similar but not identical to the Web template files included in that release. For example, if you selected the 7.8.2 and Label on Top ICL options, then the postmerge utilities copy Web template files from the `782` and `TopLabel` subdirectories of `repatch`.

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

ICL and HTML Style Sheets

The postmerge utilities do not copy-in a new version of the HTML style sheets, `main.css` and `printmain.css`. The style sheets for the new release are used to render all UI objects, including those copied-in from the Prior Customer Repository. If you customized the style sheets in the release you are upgrading from, then 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 CRM version 7.x to Siebel CRM version 7.7, then performing an ICL merge when upgrading to the next release is not recommended.

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

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

A Siebel 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, then 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 51](#)

[“About Inheriting Upgrade Behavior in a Siebel Upgrade” on page 57](#)

[“About the Siebel Postmerge Utilities” on page 59](#)

[“Reviewing Siebel Repository Object Property Conflicts” on page 174](#)

D

Siebel Case Insensitivity Wizard

This appendix provides information about the Siebel Case Insensitivity Wizard. It includes the following topics:

- [Overview of What the Case Insensitivity Wizard Does on page 345](#)
- [How CIAI Columns and Indexes Are Implemented in the Database on page 346](#)

Overview of What the Case Insensitivity Wizard Does

Environments: Development environment only.

Platforms: Windows, UNIX, IBM z/OS.

Databases: All databases.

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

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

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

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

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

```
SELECT column_list FROM S_CONTACT  
  
WHERE LAST_NAME_CI indicates SMITH
```

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

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

Related Topics

[“About the Siebel Case Insensitivity Wizard” on page 50](#)

Related Books

Configuring Siebel Business Applications

How CIAI Columns and Indexes Are Implemented in the Database

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

Case Insensitivity Wizard Modes

You can run the Case Insensitivity Wizard in one of two modes:

- **Configure specified columns.** You can specify the columns you want to configure for CIAI queries by using an input file or by selecting the files manually:
 - **Use an input file.** The Case Insensitivity Wizard reads the input file and configures the columns in the file for CIAI queries. Using an input file allows you to control which configuration options the wizard uses. Oracle provides recommended input files. These files contain columns frequently used in searches. Use an input file if you want to enable large numbers of columns or to use methods or index strategies other than the defaults.
 - **Select columns manually.** You can use the Tools Object Explorer to manually select and configure specific columns for CIAI queries. The Case Insensitivity Wizard uses default settings to configure these columns. If you want to modify the configuration options, then you can export the configuration strings to a text file, edit them, and run the wizard using the edited file as an input file.

- **Specify how queries are built for columns without indexes.** The Case Insensitivity Wizard defines CIAI columns and indexes only on columns that already have indexes defined. However, for columns without indexes that meet all other eligibility criteria, you can run the Case Insensitivity Wizard in a special mode to change the Default Insensitivity property from None to DB Case & Accent. In queries, the column values are then converted to uppercase before being compared. This allows searches to be both case and accent insensitive.

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

```
SELECT column_list FROM S_CONTACT  
WHERE UPPER(LAST_NAME) LIKE UPPER(Smi th)
```

Input File Format

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

The record format is as follows:

Table_Name, Column_Name, Method, Index_Strategy, Operation

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

S_CONTACT, EMAIL_ADDR, Database, Copy All, On

Only the first two items in the record are mandatory (*Table_Name* and *Column_Name*). For records where items are omitted, the Case Insensitivity Wizard inserts the default value for the items.

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

S_CONTACT, EMAIL_ADDR, Database, , On

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

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

Windows: *SIEBEL_TOOLS_ROOT*\objects

Eligibility Criteria

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

- The table and column must exist in the repository.
- The column must be active and belong to the specified table.
- The table type must be supported for CIAI configuration.

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

Table Name: Supported Table Types

Table 49 lists the table types that can be configured for CIAI queries.

Table 49. Supported Table Types

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

Column Name: Supported Column Functional Types

Table 50 lists the column functional types that can be configured for CIAI queries.

Table 50. Supported Functional Column Types

Column Functional Type	Can Be Configured for CIAI Queries?
Data (Private)	Yes
Data (Public)	Yes
Denormalized	Yes
Extension	Yes

Table 50. Supported Functional Column Types

Column Functional Type	Can Be Configured for CIAI Queries?
External	No
IFMGR: Exists	No
IFMGR: FKey	No
IFMGR: Status	No
IFMGR: ROW_ID	No
System	No

Column Name: Supported Physical Column Types

Table 51 lists the physical column types that can be configured for CIAI queries.

Table 51. Supported Physical Column Types

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

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

Method and Index Strategy

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

There are two methods:

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

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

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

- **Database.** The Database method defines a CIAI column for the base column and uses an *index strategy* to create indexes.

The index strategy in effect determines how indexes are defined for the CIAI column, as described in [Table 52](#).

Table 52. Index Strategies Used by the Database Method

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

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

For example:

- The S_CONTACT table has Base Column A with Index A that has two columns as keys, LAST_NAME and FST_NAME.
- You then select the column LAST_NAME for CIAI queries and specify the Copy All index strategy.
- The Case Insensitivity Wizard defines the column LAST_NAME_CI and a CIAI index for the new column.

- The wizard also defines Index B for Base Column A by copying Index A and specifying LAST_NAME_CI and FST_NAME as keys.
- You then select the column FST_NAME for CIAI queries.
- As part of configuring FST_NAME for CIAI queries, the wizard does the following in the repository:
 - Defines a new column FST_NAME_CI.
 - Deletes Index B on Base Column A and redefines it with the keys LAST_NAME_CI and FST_NAME_CI.

Operation: Inactivating CIAI Configuration

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

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

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

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

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

Case Insensitivity Wizard Defaults

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

Methods

The Case Insensitivity Wizard uses the following default settings for the methods:

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

Index Strategy

The Case Insensitivity Wizard uses the following defaults for index strategy:

- If the method is Force Case, then the wizard sets the index strategy to None.
- If the method is Database, and the base column does not have indexes defined on it, then the wizard sets the index strategy to None.

- If the method is Database, and the base column has indexes defined on it, then the wizard uses the Copy All index strategy.

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

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

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

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

Operation

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

Running the Case Insensitivity Wizard Multiple Times

You can run the Case Insensitivity Wizard multiple times. Typically, you would do this to tune the CIAI configuration of columns. If you have already run the Case Insensitivity Wizard once to configure columns, then you might run it again to perform additional tasks. [Table 53](#) lists operations that you can perform when running the Case Insensitivity Wizard again:

Table 53. Running the Case Insensitivity Wizard Multiple Times

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

Table 53. Running the Case Insensitivity Wizard Multiple Times

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

Column and Index Naming Convention

The Case Insensitivity Wizard uses the same naming conventions as the Siebel Enterprise Integration Manager Wizard when creating CIAI column names and index names. The naming conventions used by the wizard are fixed and cannot be overridden.

Column Names

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

Index Names

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

base_table_name_C#

In this table name, `#` is an integer incremented starting at 1 as needed to create a unique name.

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

Truncating Names

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

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

- Deletes underscores one at a time, beginning with the left-most
- Deletes vowels one at a time, beginning with the right-most

- Deletes characters (letters, numbers, and so on) one at a time, beginning with the right-most. The Case Insensitivity Wizard does not truncate prefixes or postfixes.

Column and Index Name Uniqueness

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

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

Error Reporting

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

You then have two options:

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

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

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

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

Index

- A**
 - about database environments** 24
 - access control**
 - about 75
 - address data**
 - upgrade planning 75
 - aggregate categories (new), reviewing** 259
 - AIX memory allocation segment space**
 - securing for Siebel Database 86
 - AIX memory allocation segment space, securing** 86
 - applet functionality, reviewing** 251
 - audience for guide** 32
- B**
 - binary sort order**
 - Microsoft SQL Server, verifying database was created using 103
- C**
 - chart views, assigning category and type** 260
 - clustered indexes**
 - rebuilding 105
 - configuration parameters, about and table** 104
 - copied objects, automatic upgrade of**
 - upgrade ancestor, choosing 58
 - Customer Repository, preparing for merge**
 - inheriting upgrade behavior, about 57
 - New Customer Repository, preparing to import 149, 178, 237
 - upgrade inheritance, configuration steps 168
- D**
 - database server configuration, verifying**
 - Oracle database server configuration 99
 - database sort order, verifying**
 - about 36
 - IBM DB2, verifying database created using identity sort 89
 - Microsoft SQL Server, verifying sort order on 103
 - database, preparing for upgrade**
 - AIX memory allocation segment space, securing 86
 - binary sort order, verifying database created using 103
 - database sort order, verifying 36
 - Microsoft SQL Server, verifying sort order on 103
 - mobile and dedicated users, preparing for upgrade 111
 - New Customer Repository, preparing to import 149, 178, 237
 - RDBMS software, upgrading 35
 - dedicated users, preparing for upgrade** 111
 - dock objects and visibility rules**
 - preserving 85
- E**
 - Encryption Upgrade Utility**
 - RC4 and AES encryption, about upgrading to 73
 - encryption, upgrading to RC4 or AES**
 - about 73
 - explorer views, assigning category and type** 261
- F**
 - file paths, commands**
 - about 22
 - executing 23
 - path navigation 23
 - UNIX 23
 - Windows 22
- G**
 - Generate Reporting Relationships button, exposing** 268
 - global deployment, postupgrade tasks**
 - global time zone, about setting up environment to support 269
 - global time zone, enabling after an upgrade 269
 - Unicode, migrating to 71
 - grid-based applets, reviewing** 252
 - guide**
 - audience for 32
 - naming and typographical conventions 21
 - guidelines**
 - Siebel Database upgrade 32

H**Hierarchical Merge Report** 216**I****IBM DB2, preparing database for upgrade**

32-KB temporary tablespace, creating 92

16-KB temporary tablespace, creating 92

database configuration parameters, about and table 91

DB2 Application Development Client, verifying installation of 95

DB2 permission, verifying on AIX and Oracle Solaris 92

identity sort order, verifying database created using 89

instance owner permission, verifying for AIX and Oracle Solaris 92

RDBMS software, upgrading 35

sort order, verifying on 89

verifying client for Siebel upgrade 89

ICL

about 333

reviewing UI objects affected by 263

identity sort order, verifying database created using 89**Incorporate Custom Layouts, about** 333**incremental merge** 187**incremental repository merge** 187

preparing to restart 201

report for Usage Pattern Tracking 212

restarting 202

indexes

clustered indexes, rebuilding 105

inheriting upgrade behavior, about 57**item identifier, Web template** 265**L****License Key Activation utility** 267**license keys, activating or deactivating log files** 267

archiving manually 165

manually archiving 165

reviewing upgrade log files 161

SiebSrvr log files, about saving 157

using Logparse utility 159

M**Microsoft SQL Server**

clustered indexes, rebuilding 105

configuration parameters, about and table 104

database sort order, verifying 103

Siebel Software Configuration utility, changing language utility runs 132

temporary database space, about 104

mobile users, preparing for upgrade 111**multilingual deployment, installing additional languages** 69**multilingual seed data**

Siebel Database Server for DB2, installing 69

multilingual upgrade, preparing for 86**multi-value group shuttle applets, reviewing** 255**N****naming conventions** 21**O****objects**

descendents or copies, viewing 168

obsolete Siebel UI fields, eliminating 263**Oracle database**

server configuration, before upgrading tasks 99

sort order, verifying on 99

upgrading, steps before upgrading 99

Oracle GoldenGate

about Siebel high availability upgrade files 291

example of implementation 298

extracting files 291

implementation 290

P**performance tuning, process of** 320**planning**

guidelines 32

mobile devices 79

Order Management 81

Resonate Central Dispatch 80

resources 21

Workflow Designer 76

postmerge utilities

about 59

postupgrade tasks

dock object and rule definitions, verifying are correct 247

Generate Reporting Relationships button, exposing 268

inbound workflows, updating 273

log files, manually archiving 165

previously unused tables now used in 7.x, examples of 245

reporting relationships, generating (procedure) 268

- reporting relationships, generating,
 - about 268
- symbolic string model, upgrading to 243
- tables not used in 7.x, examples of 245

preupgrade tasks

- AIX memory allocation segment space,
 - securing 86
- binary sort order, verifying database created
 - using 103
- database sort order, verifying 36
- Microsoft SQL Server, verifying sort order
 - on 103
- mobile and dedicated users, preparing for
 - upgrade 111
- New Customer Repository, preparing to
 - import 149, 178, 237
- RDBMS software, upgrading 35

production environment upgrade

- global time zone, about operating in 269
- New Customer Repository, preparing to
 - import 149, 178
- preparing production upgrade on UNIX 141,
 - 196

R

RDBMS software, preparing for upgrade 35

reporting relationships

- about generating 268
- Generate Reporting Relationships button,
 - exposing 268
- generating (procedure) 268

repositories, upgrading

- about 51
- EIM temporary columns, generating 174
- Incorporate Custom Layouts (ICL) 333
- postmerge utilities, about 59
- renaming 129
- upgrade behavior, inheriting 57

repository merge results

- determining success 171, 213
- reviewing conflicts 174, 218
- reviewing deleted objects 241
- reviewing log files 215
- reviewing obsolete objects 242

rich text controls, revising 258

RTCs, revising 258

S

seed data

- installing multilingual seed data 69
- preparing customized for upgrade 108

shuttle applets, reviewing 255

Siebel Additive Schema Changes mode,

- about 46

Siebel Business application, upgrading

- multilingual deployment, installing additional
 - languages 69

Siebel Case Insensitivity Wizard

- about 50
- functional overview 150
- running 149

Siebel custom indexes

- preparing for upgrade 84

Siebel data model

- access control, about 75

Siebel Database

- verifying connectivity 83

Siebel Database Configuration Utilities, about 37

Siebel Database Configuration Wizard

- preparing to run 132, 191
- running on UNIX 141, 148, 196, 200
- running on Windows 136, 195
- starting 145
- stopping on Windows 147, 199

Siebel Database Schema, upgrading

- restarting upgrade 145

Siebel Database Server software

- multilingual deployment, installing additional
 - languages 69

Siebel Database upgrade

- about process 37
- guidelines 32
- performance tuning 320
- roadmap 301
- roadmap, configuring 304
- roadmap, evaluating 303
- roadmap, implementing 308
- roadmap, planning 303
- roadmap, testing 307

Siebel development environment, process of upgrading 308

Siebel Financial Services

- household data integrity, verifying 109

Siebel interface table data

- exporting 85

Siebel Personalization

- upgrade planning 81

Siebel production environment, process of upgrading 324

Siebel production test environment, process of upgrading 315

Siebel Repository

- archiving unneeded repositories 85
- development environment repositories,
 - preparing, renaming 129
- dock objects and visibility rules,

- recording 85
 - merge, about 51
 - merge, performing 169
 - merge, preparing for 167
 - merge, recovering from failed 167
 - renaming 129
 - Siebel Repository, incremental merge** 187
 - custom database schema, upgrading 221
 - process of performing 190
 - requirements for 187
 - Siebel development environment, process of upgrading 204
 - Siebel production, production test environments, performing upgrade 240
 - Siebel repository definition files, process of regenerating 237
 - Siebel Software Configuration utility**
 - language, changing 132
 - Siebel tables and view**
 - preparing for upgrade 83
 - Siebel UI**
 - reviewing required fields 264
 - Siebel upgrade environments**
 - about 37
 - development upgrade flow example 51
 - Siebel Upgrade Wizard, about** 43
 - Siebel Web templates**
 - migration 236
 - upgrade planning 73
 - Siebel Workflow**
 - inbound workflow, updating 273
 - SiebSrvr log files, about saving** 157
 - sort order**
 - binary sort order, verified database created using 103
 - Microsoft SQL Server, verifying on 103
 - string translation**
 - upgrade planning 80
 - supported upgrade paths, about** 24
 - swt_mig.log file** 236
- T**
- tables, preparing for upgrade**
 - DB2 long columns, identifying for truncation 35, 36, 84, 85, 96, 99,
 - 111, 149, 178, 237
 - Tools repository, renaming** 129
 - typographical conventions** 21
- U**
- UI navigation, reviewing** 254
 - Unicode**
 - migrating to, about and procedure 71
 - migration, planning considerations 72
 - upgrade environments**
 - about 24
 - about, development 25
 - about, mapping 25
 - about, postupgrade 28
 - about, production 27
 - about, production test 26
 - upgrade inheritance**
 - about 57
 - configuration steps 168
 - upgrade log files, about** 48
 - upgrade paths**
 - about supported 24
 - upgrade process**
 - Siebel production environment 324
 - Siebel production test environment 315
 - upgrade, planning for**
 - resources 29
 - upgrading ancestor, choosing an** 58
 - upgrading inheritance**
 - descendent or copies of an object, viewing 168
 - parent to descendents, propagating changes from 59
 - Usage Pattern Tracking (UPT), and incremental repository merge** 212
- V**
- view tabs, verifying display** 262
 - visibility filters, reviewing** 259
- W**
- Web template item identifier** 265
 - workflows**
 - inbound, updating 273
 - preparing processes for upgrade 107