



Siebel Database Upgrade Guide for DB2 UDB for z/OS

Version 8.0, Rev. B

May 2012

ORACLE®

Copyright © 2005, 2012 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 RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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 on 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. 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.

Contents

Chapter 1: What's New in This Release

Chapter 2: About Siebel Database Upgrade Topics

How To Use This Guide 15

How the Upgrade Topics are Organized 15

About the Applicability of Siebel Database Upgrade Topics 16

About Terms Used in Siebel Database Upgrade Topics 16

About File Paths and Commands in Upgrade Topics 17

Chapter 3: How the Siebel Database Upgrade Works

About Supported Upgrade Paths 19

About Using Oracle's Application Expert Services 20

About Unicode Support 20

About Siebel Upgrade Environments 21

About the z/OS Upgrade 23

About the Staging Database 24

About Siebel Additive Schema Upgrade Changes 24

About the Siebel Database Upgrade Process 26

About the Database Configuration Wizard 32

About the Siebel Upgrade Wizard and Driver Files 36

Job Flow of a Production Database Upgrade 39

 Key Members in a Mainframe-Centric Upgrade 43

 About the JCL Upgrade Jobs 43

About the Override File 44

About the Siebel Database Configuration Utilities 45

Chapter 4: How to Perform a Siebel Database Upgrade

Roadmap for Performing a Siebel Database Upgrade 47

 Roadmap for Upgrading from Siebel Financial Services 6.2.1 and Siebel 7.x 48

Roadmap for Upgrading to Siebel 8.0 Without a Development Environment 48

Process of Planning a Siebel Database Upgrade 48

Process of Upgrading a Siebel Development Environment 49

Process of Upgrading a Production Test Environment 57

Process of Tuning the Upgrade Performance 64

Process of Upgrading a Siebel Production Environment 66

Chapter 5: Planning A Siebel Database Upgrade

Planning Resources for Upgrading to 8.0 on z/OS 73

Planning Changes to the Physical Layout of the Schema 74

Testing Before a Production Upgrade 76

Considering Code Page Support 76

About Executing Jobs Using Siebel-Scheduled Mode or Vendor-Scheduled Mode 77

Staging and Target Database Planning 78

Obtaining Required Software and Hardware 78

Obtaining Required IBM Utilities 78

About Using the DSNTIAUL Utility 79

 About Applying the DSNTIAUL Patch 80

 About DSNTIAUL CCSID Conversion Errors 81

Obtaining Required Security Privileges 82

Planning Backup and Recovery Stages 83

About Creating a Schedule for the Upgrade 83

About Estimating Database Size 84

Upgrading Your DB2 Software 84

About Multilingual Deployments 85

About Database Clustered Indexes 85

Chapter 6: Basic Database Preparations for a Siebel Upgrade

Verifying Database Configuration 87

Creating Storage Groups 88

Preparing the Storage Layout of the Schema 88

 Modifying the Storage Control File 89

 About Extracting the Storage Control File 90

Extracting the Storage Control File	91
Validating the Extracted Storage Control File	95
Reviewing the Storage Control File	97
Reviewing EIM Table Partitioning and Data Distribution	99
Backing Up the Database	100
Granting a Siebel User Upgrade Authorization	100

Chapter 7: Preparing Siebel Application Data for Upgrade

Identifying and Resolving Duplicate Row IDs	103
Preparing Siebel eChannel Data for Upgrade	104
Preparing Forecasting Data for Upgrade	104
Preparing Products and Quotes for Upgrade	105
Preparing Financial Services Application Tables for Upgrade	106
Preparing Siebel Configurator Data for Upgrade	106
Setting the Value of the TYPE_CD Column	107
Converting LONG VARCHAR Columns to CLOB Columns	107

Chapter 8: Preparing a Development Environment for a Siebel Upgrade

Requirements for a Development Upgrade	109
About Moving Tables	109
Checking In Development Repository Projects	110
Preparing EIM Tables for Upgrade	111
Preparing for Intersection Table Maintenance and Deduplication	113
Determining Which Template File Was Used During an Extract or Merge	114

Chapter 9: Preparing a Production Environment for a Siebel Upgrade

Requirements for a Production Upgrade	117
About Moving the Customized Repository and Schema Definition Files	117
Renaming the Production Environment Repository	118
Preparing for a No-Development-Environment Siebel Upgrade	119

Chapter 10: Running the Database Configuration Wizard to Perform Upgrade Tasks

- Information Required By the Database Configuration Wizard 121
 - Information Required by the Database Configuration Wizard 122
- About Running the Database Configuration Wizard Under Windows 126
- About Running the Database Configuration Wizard Under UNIX 129
- Starting the Siebel Upgrade Wizard 132
 - Restarting the Siebel Upgrade Wizard After Pauses 134
 - Stopping the Siebel Upgrade Wizard 134
- Upgrading the Repository and Importing Seed Data 135

Chapter 11: Creating the Siebel Staging Database

- Process of Creating the Staging Database 137
- Required Tasks before Creating the Staging Database 137
- Creating the Staging Database Schema DDL Files 138
- Transferring the Staging DDL to the z/OS Host 139
- Preparing the z/OS Host Upgrade Environment and Creating the Staging Database 140
 - Requirements for Upgrade Procedures on the z/OS Host 140
 - Customizing the JCL UNIT Parameter Value 140
 - Creating and Allocating the z/OS Setup Data Sets 141
 - Preparing the Upgrade Environment and Building the Staging Database 142
- Removing Interface Tables and Triggers 145

Chapter 12: Generating the Siebel Upgrade Files

- About Generating the Upgrade Files 149
- Process of Generating the Upgrade Files 149
- Prerequisite Tasks for Generating the Upgrade Files 150
- Preparing the Additive Schema and JCL Files on the z/OS Host 151
 - Transferring the Additive Schema, PRET, and Data Migration Files to the z/OS Host 151
 - Preparing the z/OS Data Sets 152
 - Specifying the Scheduling Mode 153
 - Preparing the Additive Schema SQL and JCL Templates 153
- Applying the Additive Schema Changes to the Production Staging Database 155
- Preparing for Table Creation (PRET) on the Staging Database 156

Preparing and Executing the Non-Additive Schema and JCL Files on the z/OS Host	158
Transferring the Non-Additive Schema, Temp table, and Load and Unload files to the z/OS Host	158
Preparing the Non-Additive Schema SQL and JCL Templates and Executing the DDL	159
Applying the Non-Additive Schema Changes	160
Creating the Data Migration Indexes	161
Processing the Index Schema File	162
Transferring the SCINDEX.SQL File to the z/OS Host	163
Restructuring the Index DDL	163
Building JCL Templates for the Target Database	164

Chapter 13: Upgrading the Target Database

Process of Upgrading the Target Database	167
Creating and Loading Siebel Log Tables	168
Applying Additive Upgrade Changes to the Target Database	168
Performing the In-Place Target Database Upgrade	170
Preparing the Target Database for the Upgrade	171
Running the PRET Jobs For the Target Database	172
Applying Non-Additive Schema Upgrade Changes to the Target Database	172
Creating and Deploying Stored Procedures on the Target Database	176
Migrating Data On the Target Database	178
Creating 8.0 Schema Indexes	180
Migrating the Gen Primary 8.0 SQL to Update Data in Target Tables	181
Generating RUNSTATS Jobs	182
Restarting Upgrade Jobs That Fail	183

Chapter 14: Performing the Siebel Repository Merge

About Backing Up the New Customer Repository or Database Schema	185
About Reorganizing Tables Before the Repository Merge	186
Running the Repository Preparation Wizard on a Release 6.2.1 Repository	187
Performing a Siebel Repository Merge	189
Regenerating the Siebel Repository Definition Files	198
Creating a New SRF File	200

Chapter 15: Migrating Siebel 6.2.1 Customizations

About Migrating to the Siebel Web Client	203
--	-----

About Migrating 6.2.1 Scripts	209
About Migrating 6.2.1 Client-Side Interfaces	210
Identifying What Will Be Migrated	211
Exposing Hidden Properties of Applets and Views	212
Running the Siebel Web Client Migration Wizard	213
Reviewing Applet and View Migration	216
Running the Web Layout Wizard	217
Migrating Scripts Attached to Controls	218
Migrating Scripts Attached to Applets	220
Migrating Business Component, Business Service, and Application Scripts	223
Migrating Outbound COM Interfaces	224
Migrating Inbound COM Interfaces	225
Regenerating the Postmerge Utilities Log	228
Reviewing Customized Business Components	228

Chapter 16: Performing Post-Upgrade Tasks On the Target Database

Transferring the Development Environment Upgrade Output Files to the z/OS Host	231
Synchronizing the Schema	232
Migrating Custom Business Component Configurations	233
Performing Intersection Table Maintenance	234
Installing New License Keys After Upgrade	237
Deleting Redundant Upgrade Files	237

Chapter 17: Reviewing the Siebel Upgrade Log Files

About the Siebel Upgrade Log Files	239
Reviewing Siebel Upgrade Log Files for Errors	241
Manually Archiving Upgrade Log Files	242
Viewing the Siebel Job Log Status	242
Running SQL in Siebel Logs	243

Chapter 18: Postupgrade Tasks for a Siebel Development Environment

Reapplying Schema Customizations to the Siebel Database 245

Deleting Duplicate EIM Mappings 247

Regenerating the Database Template File 248

Chapter 19: Postupgrade Tasks for Siebel Database and File System

Updating the File System Directory 249

Updating File System Attachments 249

Chapter 20: Postupgrade Tasks for Siebel Applications

Setting Visibility Modes for Access Control 253

Removing Call Center Duplicate Logins After Upgrade 256

Upgrading File System Attachments for Siebel Financial Services Call Reports 256

Upgrading File System for Household Notes and Attachments 258

Configuring Products and Quotes 259

Upgrading Siebel Seeded Workflows 262

Upgrading Inbound Workflows 262

Chapter 21: Tuning the Siebel Production Upgrade Scripts

About Tuning the Upgrade Scripts 265

Optimizing Unload and Load Job Performance 266

Adding the Statistics Clause to Load Cards 267

Appendix A: Siebel Upgrade Planning Worksheet

Team Lead Summary 269

DB2 Connect Information 269

Siebel Development Environment Information 270

Siebel Production Environment Information 271

z/OS Host System Variables Information 272

Appendix B: Columns Denormalized During Upgrade to Siebel Release 8.0

Denormalized Columns for 6.2.1 Siebel Financial Services Applications 273

Denormalized Columns for 6.2.1 Siebel Business Applications 282

Denormalized Columns for 6.3 Siebel Industry Solutions	289
Denormalized Columns for 7.0.4 Siebel Business Applications	299
Denormalized Columns for 7.0.4 Siebel Financial Services Applications	301
Denormalized Columns for 7.0.4 Siebel Industry Solutions	305
Denormalized Columns for 7.5.2 Siebel Industry Applications	309

Appendix C: Siebel Upgrade Files

Siebel 8.0 z/OS Upgrade Files	311
Tables Amended During PRET Unload Processing	318
PRET Members Generated By Pretedit.txt	320
Tables Amended During PRET Processing	321

Index

1

What's New in This Release

What's New in Siebel Database Upgrade Guide for DB2 UDB for z/OS, Version 8.0, Rev. B

No new features have been added to this guide for this release. This guide has been updated to reflect only product name changes.

What's New in Siebel Database Upgrade Guide for DB2 UDB for z/OS, Version 8.0, Rev. A

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

Table 1. New Product Features in Siebel Database Upgrade Guide for DB2 UDB for z/OS, Version 8.0, Rev. A

Topic	Description
"About Siebel Additive Schema Upgrade Changes" on page 24	Amended topic. To ensure index creation during the additive upgrade process does not impact Siebel application usage, change the index definition to DEFINE YES, DEFER YES; this ensures the index is not populated while it is being created, so the associated table is not locked.
"Process of Upgrading a Siebel Development Environment" on page 49 "Process of Upgrading a Production Test Environment" on page 57 "Process of Upgrading a Siebel Production Environment" on page 66	Amended topics. The sequence in which the REORG and RUNSTATs utilities must be run during the development, production test, and production upgrade processes has been changed.
"Upgrade Custom Database Schema (upgphys)" on page 54	Amended topic. It is recommended that you issue an SQL DROP command to remove the Enterprise Integration Manager (EIM) tables from the database before synchronizing the Siebel logical and physical schemas.
"About Applying the DSNTIAUL Patch" on page 80	Amended topic. Added a section describing the DSNTIAUL patch, @@TIAUL USERMOD.
"Customizing the JCL UNIT Parameter Value" on page 140	New topic describing how to amend the UNIT=SYSDA parameter for the JCL generated for the Siebel upgrade.

Table 1. New Product Features in Siebel Database Upgrade Guide for DB2 UDB for z/OS, Version 8.0, Rev. A

Topic	Description
“Renaming the Production Environment Repository” on page 118	Amended topic. To prevent a naming conflict, rename your existing production repository just before you take your production database offline to run the in-place target database upgrade.
“Deleting Redundant Upgrade Files” on page 237	Amended topic. This topic now includes information on deleting stored procedures after the upgrade is completed.

Additional Changes

References to Siebel SupportWeb have been removed from the guide.

- The Siebel Bookshelf is available on Oracle Technology Network (<http://www.oracle.com/technetwork/indexes/documentation/index.html>) and Oracle Software Delivery Cloud. It might also be installed locally on your intranet or on a network location.
- *Siebel System Requirements and Supported Platforms* is located on Oracle Technology Network.
- Other Siebel CRM documentation (Release Notes, Maintenance Release Guides, Technical Notes, Alerts, Troubleshooting Steps, FAQs, Error Messages) is located on My Oracle Support.

What's New in Siebel Database Upgrade Guide for DB2 UDB for z/OS, Version 8.0

The title of this guide has changed from Upgrade Guide for DB2 UDB for z/OS to Siebel Database Upgrade Guide for DB2 UDB for z/OS. The subject matter of the book and the scope of the book have not changed.

A number of topics and chapters were removed from this guide because they duplicated information that is contained in the *Siebel Database Upgrade Guide*. These topics have been replaced with references to the relevant chapters of the *Siebel Database Upgrade Guide*, where appropriate.

Table 2 lists changes described in this version of the documentation to support release 8.0 of the software.

Table 2. New Product Features in Siebel Database Upgrade Guide for DB2 UDB for z/OS, Version 8.0

Topic	Description
<ul style="list-style-type: none"> ■ Application Upgrade Planning ■ Tables Modified or Seeded During Upgrade ■ Siebel Marketing Upgrade Reference 	These chapters were deleted because the information they contain is in the core <i>Siebel Database Upgrade Guide</i> .
<ul style="list-style-type: none"> ■ Chapter 5, “Planning A Siebel Database Upgrade.” ■ Chapter 6, “Basic Database Preparations for a Siebel Upgrade.” ■ Chapter 7, “Preparing Siebel Application Data for Upgrade.” ■ Chapter 14, “Performing the Siebel Repository Merge.” ■ Chapter 20, “Postupgrade Tasks for Siebel Applications.” 	Topics that duplicated information in the <i>Siebel Database Upgrade Guide</i> were deleted from these chapters.
Chapter 3, “How the Siebel Database Upgrade Works.”	Changed a number of topics to reflect changes in the z/OS upgrade process.
Chapter 4, “How to Perform a Siebel Database Upgrade.”	Changed steps in the roadmap to reflect the new z/OS upgrade process.
“About Using the DSNTIAUL Utility” on page 79	New topic. Describes the DSNTIAUL patch supplied with Siebel 8.0.
“About Estimating Database Size” on page 84	Revised this topic to include the number of tables and the space required for a sample Siebel 8.0 SIA database.
“About Multilingual Deployments” on page 85	This topic was amended. In Siebel 8.0, the upgrade process upgrades both the primary (base) language and all deployed languages for upgrades from Siebel 7.x. If you are upgrading from Release 6.2.1, however, only the primary language is upgraded; all other deployed languages must be upgraded manually.
“Preparing the Storage Layout of the Schema” on page 88	Revised this topic to reflect the Database Configuration Wizard screen changes.
“Reviewing EIM Table Partitioning and Data Distribution” on page 99	New topic. This topic was formerly in <i>Implementing Siebel Business Applications on DB2 for z/OS</i> .

Table 2. New Product Features in Siebel Database Upgrade Guide for DB2 UDB for z/OS, Version 8.0

Topic	Description
“Converting LONG VARCHAR Columns to CLOB Columns” on page 107	New topic. If you are upgrading to Siebel 8.0 from releases earlier than 7.8.x, the SCRIPT column of the S_SERVICE_SCRPT table must be converted from a LONG VARCHAR to a CLOB data type or the repository merge process fails.
Chapter 10, “Running the Database Configuration Wizard to Perform Upgrade Tasks.”	Revised a number of topics to describe the new upgrade options that run from the Database Configuration Wizard. Moved topics that did not relate to running the Database Configuration Wizard to other chapters.
Chapter 11, “Creating the Siebel Staging Database.”	New chapter. This chapter describes how to generate the staging database files and how to apply them on the z/OS host. Upgrades to previous releases of Siebel Business Applications did not use a staging database because in-place upgrades were not supported.
Chapter 12, “Generating the Siebel Upgrade Files.”	New chapter. This chapter describes how to generate the upgrade files for the in-place upgrade by running the Upgrade Wizard against the staging database.
Chapter 13, “Upgrading the Target Database.”	New chapter. This chapter describes how to perform the in-place upgrade of the target database.
Chapter 16, “Performing Post-Upgrade Tasks On the Target Database.”	New chapter. This chapter describes tasks you must perform after performing the in-place target database upgrade. Some of these tasks were previously described in the Upgrading a Database from the Mainframe chapter, which was deleted.
Upgrading a Development Environment Database from the Midtier	This chapter was deleted. Midtier upgrades are no longer supported.
Upgrading a Database from the Mainframe	This chapter was deleted. Some of the information was incorporated into new chapters in this guide.

2

About Siebel Database Upgrade Topics

This chapter provides general information about the upgrade topics. It includes the following topics:

- [How To Use This Guide on page 15](#)
- [How the Upgrade Topics are Organized on page 15](#)
- [About the Applicability of Siebel Database Upgrade Topics on page 16](#)
- [About Terms Used in Siebel Database Upgrade Topics on page 16](#)
- [About File Paths and Commands in Upgrade Topics on page 17](#)

How To Use This Guide

Upgrades: All upgrades.

To perform an upgrade to Oracle's Siebel 8.0 applications on the DB2 for z/OS platform, you will need to use this guide and the *Siebel Database Upgrade Guide*.

This guide describes the upgrade process on a z/OS platform and provides information on tasks that are specific to this platform. To complete your upgrade, however, you must also perform certain tasks outlined in the *Siebel Database Upgrade Guide* that are applicable to each RDBMS supported in the Siebel 8.0 release.

This guide directs you to the *Siebel Database Upgrade Guide* at the relevant points during the upgrade process.

Topics in the *Siebel Database Upgrade Guide* that apply when performing an upgrade of Siebel Business Applications on z/OS contain the following Platforms statement:

Platforms: MS Windows, UNIX, IBM z/OS.

How the Upgrade Topics are Organized

Use the roadmaps and process topics in [Chapter 4, "How to Perform a Siebel Database Upgrade,"](#) to guide you through the upgrade process. These topics provide a checklist of all the steps required to complete a particular type of upgrade, in the order in which you must perform them. Each step includes a link to a topic that explains how to complete the step.

The remaining chapters of the guide are organized according to the major phases of the upgrade. Each chapter includes the specific upgrade tasks you must perform for that portion of the upgrade as well as conceptual and process information relating to those tasks.

CAUTION: Topics in the chapters may not follow the order you perform them during the upgrade and, depending on your upgrade path, all topics may not apply. You must use the roadmap for your upgrade to determine the required and optional steps and their sequence. If you do not, you are likely to experience difficulties in completing your upgrade.

About the Applicability of Siebel Database Upgrade Topics

Upgrades: All upgrades.

Environments: All environments.

The applicability of topics in the book is listed at the beginning of each topic. [Table 3](#) lists the applicability categories and their meaning.

For each topic, only the relevant categories are listed.

Table 3. Topic Applicability Categories

Applicability Category	Meaning
Upgrades	<p>Lists the upgrades to which the topic applies.</p> <p>For example, “Upgrades: 7.0.x, 7.5.x” means the topic applies to upgrades from 7.0.x or 7.5.x. The topic does <i>not</i> apply to upgrades from 7.7.x or later.</p>
Environments	<p>Lists the Siebel environments to which the topic applies.</p> <p>For example, “Environments: Development environment only” means the topic applies only to a development environment upgrade.</p> <p>For more information on Siebel environments, see “About Siebel Upgrade Environments” on page 21.</p>

About Terms Used in Siebel Database Upgrade Topics

Upgrades: All upgrades.

Environments: All environments.

This guide follows several naming conventions:

- *DB2* refers to IBM DB2 UDB for z/OS.
- Current release means the currently shipping release of the Siebel Business Applications provided by Oracle.
- Release 7.x refers collectively to all versions of Siebel 7, for example Release 7.0.4 or 7.5.3. See the Upgrade section in *Siebel System Requirements and Supported Platforms* on Oracle Technology Network for specific information on which release numbers are meant by Release 7.x.

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

About File Paths and Commands in Upgrade Topics

Upgrades: All upgrades.

Environments: All environments.

Environment variables and path placeholders for both Windows and UNIX paths are used throughout this guide. You must 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 topics:

- *SIEBEL_ROOT* is the absolute path to the Siebel Server installation directory. When you install a Siebel Server, the installation program queries for the Siebel 8.0 installation path and installs the Siebel Server in a subdirectory of this path called *si ebsrvr*. For example, if you specified C: \sba80 as the installation directory (this is the default), then *SIEBEL_ROOT* is C: \sba80\si ebsrvr.
- *DBSRVR_ROOT* is the absolute path to the Siebel Database Configuration Utilities files installation directory on the Siebel Server. When you install the Siebel Database Configuration Utilities, the installation program queries for the Siebel Server installation directory and then installs the Siebel Database Configuration Utilities files at the same level in a subdirectory called *dbsrvr*. For example, if *SIEBEL_ROOT* is C: \sba80\si ebsrvr, then *DBSRVR_ROOT* is C: \sba80\dbsrvr.

UNIX Paths

The following environment variables and path conventions specify file system locations in the topics in this guide:

- `SI EBEL_ROOT` is an environment variable that defines the absolute path of the Siebel Server installation directory. When you install a Siebel Server, the installation program queries for the installation directory and 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, then `$SI EBEL_ROOT` is `/usr/si ebel /si ebsrvr`.

The definition of `SI EBEL_ROOT` and other environment variables required for doing an upgrade are located in `/si ebsrvr/si ebenv. sh`. The Siebel Server installation script sets environment variable definitions in this shell script. Do not edit or delete this file.

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

- `DBSRVR_ROOT` is a path convention used in this 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 Configuration Utilities files on the Siebel Server. When you install the Siebel Database Configuration Utilities, the installation program queries for the Siebel Server installation directory and installs the Siebel Database Configuration Utilities files at the same level in a subdirectory called `dbsrvr`. For example, if `$SI EBEL_ROOT` is `usr/si ebel /si ebsrvr`, then `DBSRVR_ROOT` is `/usr/si ebel /dbsrvr`.

- Run UNIX scripts in a C or Korn shell.

Executing Commands

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

- Windows: Open a Command Prompt window and use the `cd` command to make the specified directory the current directory. Enter the command.

Do not use the Windows File Explorer to navigate to the directory and do not run the command by entering it in the Run window in the Start Menu.

- UNIX: In a shell window, make the specified directory the current directory, source the `siebenv` script, then enter the command.

Use lowercase for all filenames, directory names, path names, parameters, flags, and command-line commands, unless you are instructed otherwise.

3

How the Siebel Database Upgrade Works

This chapter provides an overview of the upgrade process and upgrade environments, and describes the utility used to perform the upgrade. Also review the topics that are relevant to the z/OS upgrade in the chapter of the *Siebel Database Upgrade Guide* that describes how the Siebel database upgrade works. This chapter includes the following topics:

- [About Supported Upgrade Paths on page 19](#)
- [About Using Oracle's Application Expert Services on page 20](#)
- [About Unicode Support on page 20](#)
- [About Siebel Upgrade Environments on page 21](#)
- [About the z/OS Upgrade on page 23](#)
- [About the Staging Database on page 24](#)
- [About Siebel Additive Schema Upgrade Changes on page 24](#)
- [About the Siebel Database Upgrade Process on page 26](#)
- [About the Database Configuration Wizard on page 32](#)
- [About the Siebel Upgrade Wizard and Driver Files on page 36](#)
- [Job Flow of a Production Database Upgrade on page 39](#)
- [About the Override File on page 44](#)
- [About the Siebel Database Configuration Utilities on page 45](#)

About Supported Upgrade Paths

Upgrades: All upgrades.

Environments: All environments.

Supported upgrade paths are described in *Siebel System Requirements and Supported Platforms* on Oracle Technology Network. This guide describes how to upgrade the following installations:

- A Siebel Financial Services 6.2.1 installation to Release 8.0
- A Siebel 7.x installation to Release 8.0

This guide does not cover the following specific upgrade paths or infrastructure changes. For help with these tasks, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services:

- Changing operating system type during an upgrade, for example changing from Windows to UNIX.
- Changing database platform type during an upgrade.

- Migrating from Siebel Industry Solutions or Siebel Financial Services applications to Siebel Business Applications.

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

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

About Using Oracle's Application Expert Services

The Siebel 8.0 upgrade process on DB2 for z/OS is designed to run in all standard Siebel implementations. It is built on the assumption that data exists in all Siebel tables and that all this data needs to be migrated.

In reality, your implementation has probably been customized to suit your business so, for example, you might not use all of the Siebel tables shipped or they might contain varying amounts of data. To accommodate this fact, the upgrade process is customizable, for example, you can eliminate upgrade jobs that run on empty tables, or you can choose to run unload jobs simultaneously.

Global Customer Support provides support for all standard z/OS upgrades but it does not support customized upgrades. If you require help with a standard upgrade, 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 are listed on My Oracle Support.

Customizing the 8.0 upgrade scripts is a complex process and, for this reason, if you want to customize the upgrade scripts, you must contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

This guide describes a number of upgrade customization tasks that you can perform where the assistance of Oracle's Application Expert Services is *required*.

CAUTION: You must contact Oracle's Application Expert Services before performing tasks where such help is noted as a *requirement*. If you do not, you may invalidate your support agreement.

This guide also describes tasks where enlisting the help of Oracle's Application Expert Services is *recommended*. Failure to contact Oracle's Application Expert Services for help with these tasks does not have implications for continuing support.

Where a task requires the help of Oracle's Application Expert Services, this is indicated in the relevant topic.

About Unicode Support

Previous releases of Siebel Business Applications supported ASCII- and EBCDIC-based coded character set IDs (CCSIDs) on DB2 for z/OS. In Siebel 8.0, Unicode is also supported on DB2 for z/OS V8.

You can migrate your Siebel database from an EBCDIC or ASCII code page to a Unicode encoding system but this process is not part of the standard Siebel 8.0 upgrade procedure. A separate migration procedure for migrating to Unicode is performed on the DB2 host after you have upgraded to Siebel 8.0. For information on migrating to Unicode, see *Implementing Siebel Business Applications on DB2 for z/OS*.

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

About Siebel Upgrade Environments

Upgrades: All upgrades.

Environments: All environments.

This guide describes how to upgrade three environments:

- Development environment
- Production test environment
- Production environment

Development Environment

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

A development environment contains the following elements:

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

There are two scenarios for development environment upgrades:

- **Development environment database with DB2 UDB.** If your development environment database is DB2 UDB, you must use the *Siebel Database Upgrade Guide* for your development environment upgrade. Then use this guide for your production environment upgrade.
- **Development environment database with DB2 UDB for z/OS.** During DB2 for z/OS upgrades, the Siebel Upgrade Wizard generates the schema and output files. Your database administrator must review these files and apply them on the z/OS host.

Production Test Environment

The production test environment is where you test the upgraded release to validate its function and performance before deploying it to users. This is also where you tune the upgrade process to minimize the time required to perform your production upgrade.

By tuning the production upgrade scripts in a test environment, you 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.

This environment includes the following elements:

- 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 Configuration Utilities installed on the same machine as the Siebel Server
- All the Siebel applications currently installed in your production environment
- A copy of the Siebel database installed in your production environment

You perform the following processes in the production test environment:

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

Tuning the upgrade scripts can significantly reduce the time required to complete the production upgrade.

Production Environment

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

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

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

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

Mapping Your Environments

You may have more or fewer environments than those described above. [Table 4](#) gives recommendations for mapping your environments to the ones described in this guide.

Table 4. Mapping Your Environments to Siebel 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 technical support or training. Developers are usually installed as 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 may be multiple upstream environments in addition to the production test environment. For example, these could include environments used by a product management group, Technical Support, and Quality Assurance. 	Production test environment upgrade.
<p>The environment is used for live business transactions by both local and remote users.</p>	Production environment upgrade.

About the z/OS Upgrade

Historically, Siebel production environment upgrades were primarily performed on the z/OS host (mainframe-centric) but you could choose to upgrade your development environment using either a mainframe-centric or midtier-centric process. With Siebel 8.0, only mainframe-centric development environment upgrades are supported for DB2 for z/OS. In a mainframe-centric upgrade, all upgrade DDL (CREATE, DROP, ALTER, and GRANT) and DML (INSERT, UPDATE, DELETE, and SELECT) are executed on the mainframe host.

Siebel upgrades on DB2 for z/OS are initiated from a midtier platform (Windows or UNIX) using the Siebel Upgrade Wizard, which automatically generates the DDL and DML files necessary for the upgrade. The Siebel Upgrade Wizard also automatically populates and updates data on the target database from the midtier.

You must manually transfer the DDL and the data migration DML files generated on the midtier platform by the Upgrade Wizard to the DB2 host where they are unpacked and applied; this process is guided on the mainframe using REXX execs and ISPF panels. This manual portion of the upgrade, which runs directly on the z/OS host, allows you to tailor the upgrade process to suit your hardware environment and to minimize downtime.

Each ISPF Upgrade panel has a bookmark label and message, indicating the last step completed or currently in progress. These bookmarks guide you through the upgrade process; you cannot re-run jobs that have completed successfully or run jobs out of sequence.

About the Staging Database

Upgrades: All upgrades.

Environments: All environments.

Previously, for recovery purposes, in-place upgrades were not supported for Siebel mainframe-centric upgrades. Instead, a target database was built for the new release and data was unloaded from the existing source database and migrated in the appropriate format to the upgraded database. The source database was not upgraded, although minor modifications were made to it during the upgrade process.

In Siebel 8.0, the production database is upgraded in-place. However, to accommodate z/OS customers high-system-availability requirements, a staging database is first created which is used to generate all the z/OS upgrade components (for example, the JCL and SQL upgrade files). The staging database allows you to generate all the midtier upgrade files and to build the JCL in advance of the actual in-place upgrade; this removes these steps from the critical upgrade path and minimizes system down-time.

The manual parts of the upgrade are run against the staging database. The automatic parts of the upgrade are run directly against the development or production database being upgraded (the target database).

The staging database is created from the existing development or production database that is to be upgraded (the target database). The target database DDL, and the storage control file specifying the physical database layout, are extracted and then executed in a separate DB2 subsystem to create the staging database. The staging database therefore contains the same schema and tables as the database to be upgraded, but it does not contain data.

About Siebel Additive Schema Upgrade Changes

Upgrades: All upgrades.

Environments: Production test, production.

In Siebel 8.0, you can apply additive and non-additive schema changes separately to the production database in order to reduce the downtime for the in-place upgrade.

Non-additive schema changes impact the running application data model and require shutting down the production database. Additive schema changes are non-disruptive; these changes do not impact Siebel application usage. Therefore, you can apply additive schema changes to the live production database before you perform the in-place upgrade. This reduces the number of steps that must be performed when the database is offline and this reduces database downtime.

This is an optional process; if you do not apply the additive schema changes before you perform the production database in-place upgrade, they are applied in one step with the non-additive upgrade changes.

Types of Changes

The additive schema files generated by the Upgrade Wizard make the following types of schema changes to support the new release. These changes do not adversely affect data integrity or database normalization:

- Creating new tables.
- Adding columns to an existing table. The column must either be specified as null, or if the column is not null, it must have a specified default value.
- Creating non-unique indexes on new tables.
- Creating a unique index on an existing table
- Altering a unique index on an existing table
- 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.
- Changing a not-null column to null
- Changing a data type from char to varchar

About Index Creation During the Additive Upgrade Process

Although additive schema changes are generally non-disruptive, index creation during the additive upgrade process can impact Siebel application usage if both of the following conditions exist:

- Indexes are specified with the DEFINE parameter set to YES
- The DEFER YES parameter is *not* specified for the index

If data exists in a table for which an index is being created, DB2 changes the DEFINE parameter value from NO to YES, and issues a warning message. If the DEFER YES parameter is not also specified for the index, the index is populated while it is being created, and locks are placed on the associated table until the process is completed, preventing updates being made to the table.

In these cases, change the index definition to DEFINE YES, DEFER YES; this ensures the index is not populated while it is being created, so the associated table is not locked. You can run the IBM DB2 REBUILD INDEX utility (DSNUTILB) to populate the index at a later time when performing regularly scheduled maintenance.

Implementation of Additive Changes

The additive and non-additive schema upgrade files are generated separately by the Upgrade Wizard in the output directory you specify when you run the Database Configuration Wizard. Additive schema file names include the *.additive* identifier. Review these files to determine the schema changes that are applied by the additive schema process and edit them as necessary, for example, verify that they do not make changes that conflict with customizations.

Additive schema changes must first be applied in a production test environment which has the same data as the production environment. After generating and applying the additive schema files to the production test database, make sure users in the test environment can enter, query and delete records to check that applying additive schema changes to the database does not affect the way in which the existing version of the Siebel application works.

About the Siebel Database Upgrade Process

Upgrades: All upgrades.

Environments: All environments.

Upgrading to a new release involves two aspects:

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

Environment Upgrade Order

If you have a development environment, you must upgrade it first. During this phase, your customizations are merged with the new Siebel release. A merged repository file and database schema file are created and become inputs to the production test environment upgrade and production upgrade.

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

Flow of the Upgrade Within an Environment

The basic flow of the upgrade process is shown in Figure 1.

NOTE: In production test environment and production environment upgrades, the upgrep + upgphys steps are run together and there are several additional deployment steps.

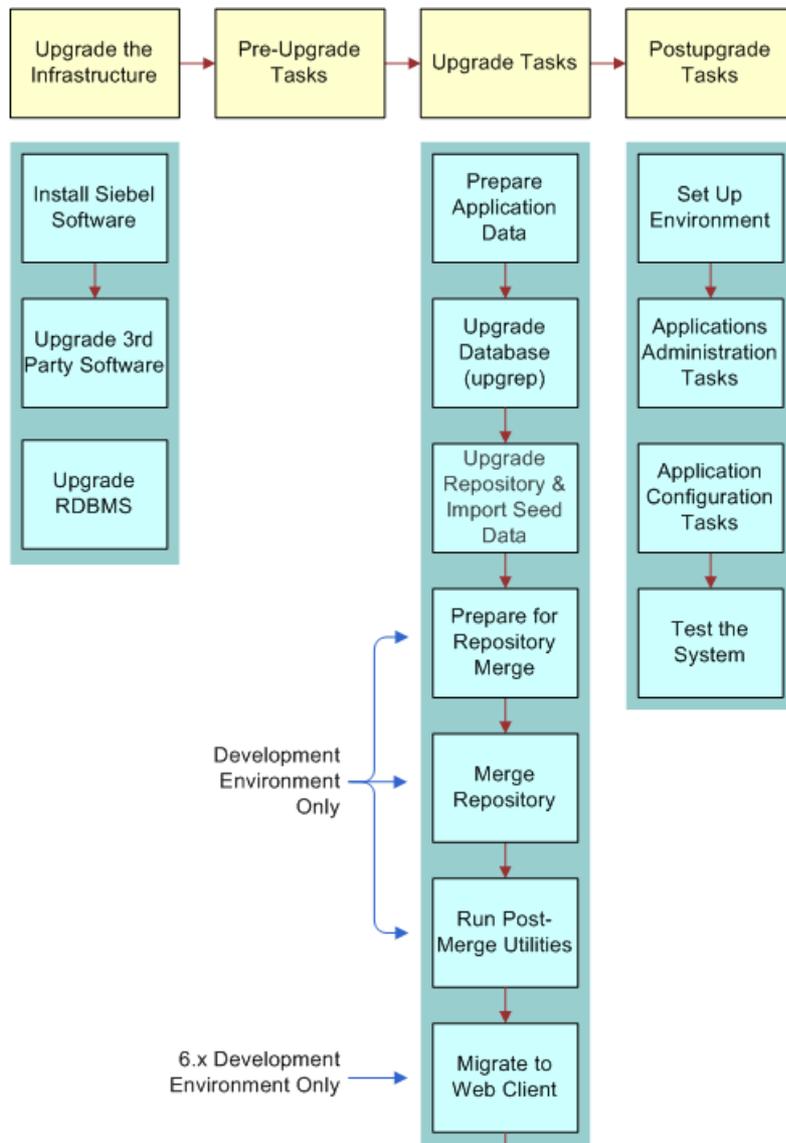


Figure 1. Flow of the Upgrade Process

Upgrade the Infrastructure

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

Perform Preupgrade Tasks

This phase prepares the Siebel database for upgrade and includes such tasks as closing database connections, clearing pending workflow tasks, disabling customized triggers and editing and validating the storage control file.

Perform Upgrade Tasks (Development Environment)

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

- **Prepare application data.** These tasks prepare test data for migration.
- **Upgrade database (upgrep).** Run the Database Configuration Wizard in upgrep mode, selecting the zSeries Staging of Files for Upgrade option. This generates the files required to build the staging database and generate the upgrade files required to perform the in-place upgrade of the database.
- **Perform the in-place upgrade of the database.** During this phase, a basic upgrade of the Siebel database schema is performed and repositories are loaded to prepare for the repository merge.

The upgrep mode makes the following changes:

- Drops interface tables and database triggers
 - Populates columns that must change from NULL to NOT NULL
 - Creates new tables. Merges existing tables.
 - Prepares for index creation. Verifies that there are no unique key violations.
 - Creates indexes
 - Imports seed data
 - Imports the Prior v.x.x Siebel 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
- **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 Wizard in upgphys mode. It further upgrades the Siebel database with changes resulting from the repository merge and completes the database upgrade.

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

Specifically, this mode performs the following tasks:

- Synchronizes the Siebel database schema to the logical schema definition in the merged repository.

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

- Deduplicates intersection tables
- Exports repository object definitions to a file, custrep.dat, and exports the logical schema definition to a file, schema.ddl

These two files are used as input to the production upgrades.

- Renames the New Customer Repository to Siebel Repository
- Updates the schema version in S_APP_VER

Perform Upgrade Tasks (Production Test Environment)

This phase upgrades a production test environment Siebel database to the new release allowing you to test how customizations work with the new release and to tune the upgrade scripts.

CAUTION: You are required to contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services for help with tuning your upgrade scripts. If you do not, you may invalidate your support agreement.

This phase includes the following tasks:

- **Prepare application data.** These tasks are about preparing application data for migration.
- **Prepare for Production Upgrade.** Run the Database Configuration Wizard in Prepare for Production Upgrade mode. This mode does the following in the production test environment.

NOTE: You must define an ODBC connection to the development environment database before performing this upgrade step.

- Examines the upgraded development environment database and generates SQL (dedup.jcl and dedup_prod.jcl files) that removes duplicate records from the intersection tables and sets up support for database aggregation.

You only have to run the utility in Prepare for Production Upgrade mode once—either in a production test environment or in the live production environment—to generate the dedup.jcl and dedup_prod.jcl files.

- Compares the repository schema and the physical database schema. Generates a file, SCINDEX.SQL, which is used to remove obsolete indexes. SCINDEX.SQL lists indexes present in the physical schema that are not present in the repository schema.
- **Upgrade database (upgrep + upgphys)**. Run the Database Configuration Wizard in upgrep + upgphys mode, selecting the zSeries Staging of Files for Upgrade option. This generates the files required to build the staging database and generate the upgrade files required to perform the in-place upgrade of the database. The additive and non-additive schema upgrade files are generated separately.
- **(Optional) Apply the additive upgrade files to upgrade the database.** This is a non-disruptive upgrade process.
- **Perform the in-place upgrade of the database.** The following changes are made during this phase of the upgrade process:
 - Drops interface tables and database triggers
 - Populates columns that must change from NULL to NOT NULL
 - Uses the custrep.dat and schema.ddl files from the development environment upgrade to create new tables and merge existing tables.
 - Prepares for index creation. Verifies that there are no unique key violations.
 - Creates indexes
 - Imports seed data
 - Updates primary children foreign key references
 - Performs miscellaneous file actions
 - Makes several administrative changes to table data, including updating the schema version in S_APP_VER.
- **Upgrade the repository and import seed data (upgrep + upgphys)**. Run the Database Configuration Wizard in upgrep + upgphys mode again, selecting the zSeries Seed/Repository Upgrade option to complete upgrade processing.
- **Tune upgrade scripts (optional)**. You can improve the performance of the production environment upgrade by tuning the production upgrade scripts in the test environment.

Run several production upgrades against the test database. This allows you to understand the upgrade process before performing the production upgrade, to conduct performance testing, and to fine-tune the upgrade scripts. After carrying out thorough performance testing, you can perform the live production upgrade using the tuned upgrade files.

Perform Upgrade Tasks (Production Environment)

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

- **Prepare Application Data.** These tasks involve preparing application data in the production database for migration.
- **Upgrade database (upgrep + upgphys).** Run the Database Configuration Wizard in upgrep + upgphys mode, selecting the zSeries Staging of Files for Upgrade option. This generates the files required to build the staging database and generate the upgrade files required to perform the in-place upgrade of the database. The additive and non-additive schema upgrade files are generated separately.

NOTE: If you have tuned the upgrade scripts during a production test upgrade and want to use them during your production upgrade, you do not have to perform this step. Instead, change the production test environment values in the upgrade files to production environment values and then apply these files on the Z/OS host to upgrade the target database. For further information on this task, see [“Process of Upgrading a Siebel Production Environment”](#) on page 66.

- **(Optional) Apply the additive upgrade files to upgrade the database.** This is a non-disruptive upgrade process.
- **Perform the in-place upgrade of the database.** The same changes are made during the in-place upgrade of the production environment as were made during the in-place upgrade of the production test environment.
- **Upgrade the repository and import seed data (upgrep + upgphys).** Run the Database Configuration Wizard in upgrep + upgphys mode again, selecting the zSeries Seed/Repository Upgrade option to complete upgrade processing.

NOTE: You do not have to run the utility in Prepare for Production mode before starting your production environment upgrade. You ran it as part of the production test environment upgrade. The required upgrade SQL commands have already been generated.

Perform Postupgrade Tasks

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

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

Related Topics

[“About the Database Configuration Wizard” on page 32](#)

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

[“About the Siebel Database Configuration Utilities” on page 45](#)

About the Database Configuration Wizard

Upgrades: All upgrades.

Environments: All environments.

The Database Configuration Wizard is part of the Siebel Configuration Wizard. It interactively gathers the information required to perform the following operations:

- Install the Siebel database. This wizard sets up the Siebel database in the RDBMS as part of a first-time installation of Siebel applications. It is also used to add a language to the Siebel Database Server installation.
- Upgrade the Siebel database. This wizard upgrades the Siebel database to a new release in a development or production environment.
- Import or export a Siebel repository. This wizard moves entire repositories between database environments with the same schema definition.
- 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.
 - Convert existing Lists of Values (LOVs) to Multilingual Lists of Values (MLOVs).
 - Configure the database by extracting storage control files from the DB2 catalog and validating the extracted files.
 - Migrate the database from an EBCDIC or ASCII encoding format to a Unicode encoding format.

About Running the Upgrade Database Option

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

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

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

To upgrade a development environment, production test environment, or production environment, you must run the Database Configuration Wizard (and Siebel Upgrade Wizard) several times, as shown in [Table 5](#).

Table 5. Database Configuration Wizard Modes

Upgrade Step	Select This Environment Type	Select This Upgrade Option(s)
Development environment upgrep	Development	upgrep: zSeries Staging of Files for Upgrade
Development environment upgrep	Development	upgrep: zSeries Seed/Repository Upgrade
Development environment upgphys	Development	upgphys
Production test environment prepare for production	Production	Prepare for Production Upgrade
Production test environment upgrep + upgphys	Production	upgrep + upgphys: zSeries Staging of Files for Upgrade
Production test environment upgrep + upgphys	Production	upgrep + upgphys: zSeries Seed/Repository Upgrade
Production environment upgrep + upgphys	Production	upgrep + upgphys: zSeries Staging of Files for Upgrade
Production environment upgrep + upgphys	Production	upgrep + upgphys: zSeries Seed/Repository Upgrade

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

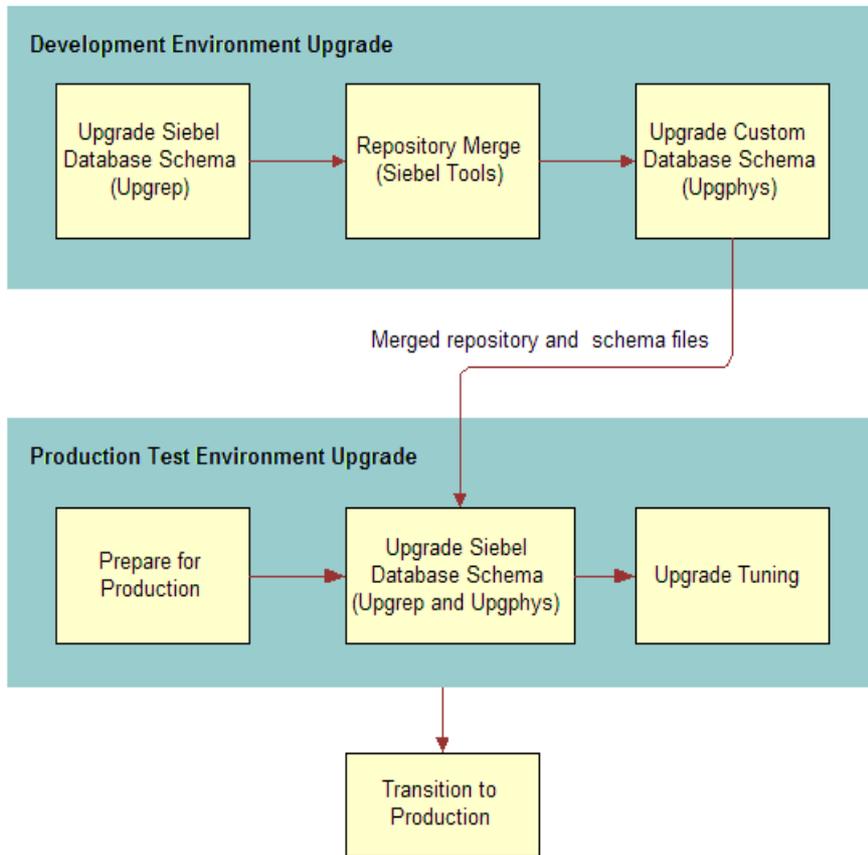


Figure 2. Upgrade Flow

How the Upgrade Configuration File and SQL Files Are Created

When you run the Database Configuration Wizard, it does the following:

- Collects configuration information.
- Creates a master upgrade configuration file (UCF). This file maps the information you entered in the Database Configuration Wizard to environment variables. When the Siebel Upgrade Wizard is performing the steps in a driver file, it uses these variables to generate the command contained in each step.
- Prompts you to start the Siebel Upgrade Wizard. The wizard forwards the information in the UCF file to an SQL generator that creates or populates SQL files on the midtier with the required commands to perform the upgrade. You then transfer these files to the z/OS host and apply them.

In some cases, you will need to modify the generated SQL files as required by Siebel Release Notes, Technical Notes, or Siebel Alerts before you transfer the files to the z/OS host.

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_MasterFileType.ucf

where:

- *UPGRADEOPTION* = the upgrade process you are performing. This can be one of the following:
 - Siebel database schema upgrade = upgrep
 - Custom database schema upgrade = upgphys
 - Siebel Database Upgrade (production) = upgrep + upgphys
 - Prepare for production upgrade = prepare_for_production_upgrade
- *ENVIRONMENT* = the environment that you are upgrading. This can be one of the following:
 - Development environment upgrades = dev
 - Production environment upgrades = prod
- *VERSION* = the version from which you are upgrading. The following numbers are used for the Siebel Release version of the file name:
 - 621
 - 704
 - 752
 - 752_200
 - 753
 - 77
 - 782
- *MasterFileType* = the type of master file. This can be one of the following:
 - Configuration files generated for the manual phase of the upgrep process (zSeries Staging of Files for Upgrade process) = mf_m
 - Configuration files generated for the automatic phase of the upgrep process (zSeries Seed/Repository Upgrade process) and for the upgphys process = mf

The automatic phases of the upgrade process are run after the manual phases are completed.

Example

If you are upgrading from Siebel 7.0.4, the UCF files generated from the development environment upgrade are as follows:

```
master_upgprep_dev_704_mf_m.ucf
master_upgprep_dev_704_mf.ucf
master_upgphys_dev_704_mf.ucf
```

The UCF file generated from the Prepare for Production Upgrade mode is as follows:

```
master_prepare_for_production_upgrade.ucf
```

About the Siebel Upgrade Wizard and Driver Files

Upgrades: All upgrades.

Environments: All environments.

The Upgrade Wizard generates the files required to perform the upgrade of the Siebel database on the z/OS host and also makes changes to the Siebel database directly. After the Siebel Upgrade Wizard starts, it executes this process:

- Reads the upgrade configuration file (UCF) generated by the Database Configuration Wizard
- Calls a driver
- Passes the information in the UCF to the driver, which then passes UCF file information to, for instance:
 - the `ddlmp` utility, which executes `ddl`-type SQL commands
 - the `dataimp` utility, which executes data-related SQL commands

The Siebel Upgrade Wizard also does the following:

- Upgrades Siebel seed data
- Loads the Prior Customer Repository for development environment upgrades

If `ddlmp` or `dataimp` encounter errors, they exit. When the errors have been corrected, you can launch the Siebel Upgrade Wizard, and the upgrade resumes from where it stopped.

The Siebel Upgrade Wizard pauses on four occasions during the upgrade file generation process. At each pause, you must transfer the files that the Upgrade Wizard has generated to the z/OS host and execute them before you resume the upgrade. Message text at each pause informs you of the tasks you must perform.

The files generated by the Upgrade Wizard are output by default to the `DBSRVR_ROOT\DB2390\dboutput\upgrade` directory (Windows) or `DBSRVR_ROOT/DB2390/dboutput/upgrade` directory (UNIX) or to the DDL Output Directory you specified when you ran the Database Configuration Wizard.

Driver Files

The Siebel Upgrade Wizard performs the upgrade by executing the commands and SQL scripts 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 Siebel Upgrade Wizard.

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

- driver_upgrep_dev_704_mf_m.ucf
- driver_upgrep_dev_704_mf.ucf
- driver_upgphys_dev_704_mf.ucf

Here is an excerpt from a driver file that controls a development environment upgrep from Release 7.0.4 SIA to Release 8.0 SIA. The excerpt contains two 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 \$UserName /p \$Password /q \$DatabaseOwner /separator / /a /c rem /l \$SiebelLogDir/del appkey.log \$DbsrvrRoot/\$DatabasePlatform/del appkey.sql /v

Number of 10 Second Wait Loops = 2000

Return Code = 0

Title = Delete old license key

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\DB2390\UPGRADE\VERSION*

UNIX: *DBSRVR_ROOT/DB2390/UPGRADE/VERSION*

where:

VERSION = the version from which you are upgrading, for example v7.7

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

driver_upgrep_dev_77_mf_m.ucf

driver_upgrep_dev_77_mf.ucf

Related Topics

[“About the Database Configuration Wizard” on page 32](#)

[“About the Siebel Database Configuration Utilities” on page 45](#)

Job Flow of a Production Database Upgrade

Upgrades: All upgrades.

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

This topic describes the major steps in a production database upgrade using the Database Configuration Wizard and Upgrade Wizard. This topic does not describe all the pre- and post-upgrade tasks you must complete for a production environment. See [“Process of Upgrading a Siebel Production Environment” on page 66](#) for a description of all the steps in upgrading a production environment.

The production environment upgrade job flow differs from a development environment upgrade job flow primarily in that a repository merge is not required. The repository (custrep.dat) and the logical schema (schema.ddl) are exported from the upgraded development environment and used in the production upgrade.

A production *database* upgrade job flow, as illustrated in Figure 3, is largely the same as the job flow of the *upgrep* stage of a development database upgrade.

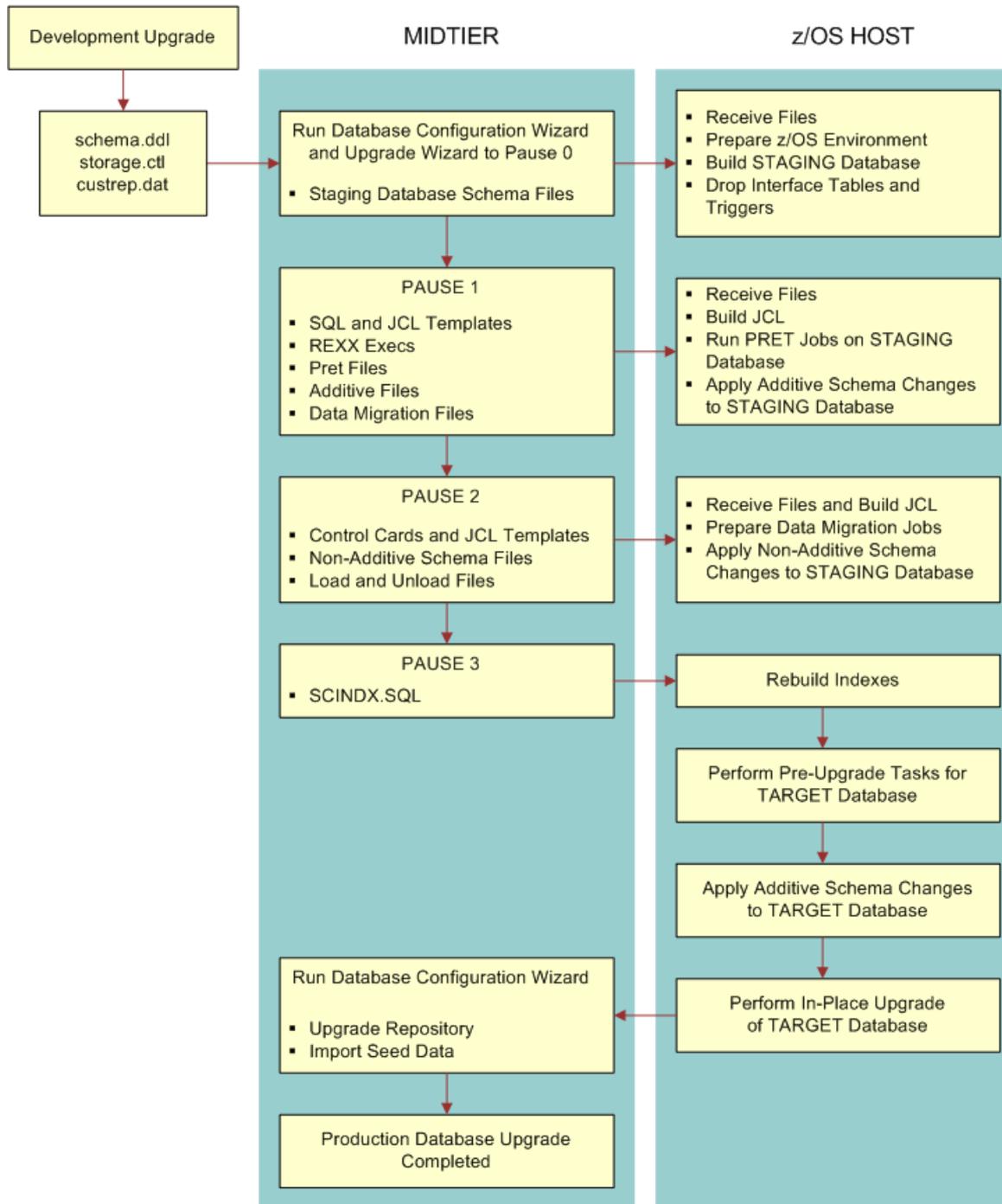


Figure 3. Flow of a Production Database Upgrade

The main steps in the production upgrade illustrated in [Figure 3](#) are as follows:

- 1 Inputs to the Production Upgrade.** The files required for the production upgrade are:
 - a** The storage control file (storage.ctl). This file specifies the storage layout of your upgraded database. Create and validate the storage.ctl before running a production upgrade.
 - b** The custrep.dat file (new customized repository) and the schema.ddl file (modified schema definition file). These files are generated during the development upgrade and must be copied to the appropriate upgrade directory before running a production upgrade.
- 2 Prepare for Production.** Run the Database Configuration Wizard on the midtier in Prepare for Production Upgrade mode to generate dedup files and the SCINDEX.SQL file; the dedup files remove duplicate records from the intersection tables.
- 3 Run upgrep + Upgphys.** Run the Database Configuration Wizard on the midtier:
 - a** Enter production environment information to create the production upgrade configuration file. Specify the following values:
 - Upgrade Option: Upgrade Siebel Database Schema (upgrep + upgphys)
 - Upgrade Process: zSeries Staging of Files for Upgrade
 - b** Launch the Upgrade Wizard to generate the DDL files that will be used to build the staging database.

The Wizard generates the staging database schema files on the midtier and then stops (Pause #0).
- 4 Create the Staging Database.** Transfer the staging DDL files from the midtier to the z/OS host and apply them. Create the staging database and prepare the z/OS upgrade environment.
- 5 Generate Additive Schema Upgrade Files.** Restart the Database Configuration Wizard on the midtier. The wizard generates files and then stops at Pause #1.
- 6 Apply Pause #1 Files.** The files generated by the Upgrade Wizard on the midtier up to pause #1 consist of JCL and SQL templates and REXX execs that are run on the z/OS host against the staging database. These jobs are used to perform pre-upgrade tasks for the staging database. Do the following:
 - a** FTP the files generated by the Upgrade Wizard from the midtier to the mainframe.
 - b** Receive the files, unpack the staging data sets into PDS format, and create JCL members to run the additive schema and PRET jobs against the staging database.
 - c** Apply additive schema changes to the staging database.
 - d** Run PRET jobs against the staging database.
- 7 Generate Non-Additive Schema Upgrade Files.** Restart the Database Configuration Wizard on the midtier. The wizard generates files and then stops at Pause #2.
- 8 Apply Pause # 2 Files.** The files generated by the Upgrade Wizard on the midtier up to pause #2 include non-additive schema files, load control cards and JCL templates, and unload SQL and JCL templates. Use the files generated up to this point to perform the in-place upgrade of the staging database. Do the following:
 - a** FTP the files that have been generated from the midtier to the host.

- b** Modify the load and unload data sets.
 - c** Run the unpack jobs to populate PDS members.
 - d** Customize the UNLOAD and LOAD jobs for target tables with CLOB columns.
 - e** Add jobcards and Siebel logging to the Load, Unload, and data migration jobs.
 - f** Apply the non-additive schema changes to the staging database.
 - g** Create temporary tables and indexes for the data migration scripts.
 - h** Generate index rebuilds.
- 9 Generate the SCINDEX Upgrade File.** Restart the Database Configuration Wizard on the midtier. The wizard generates files and then stops at Pause #3.
- 10 Apply Pause # 3 Files.** The SCINDEX file generated at this point is used to rebuild indexes. Do the following:
- a** FTP the SCINDEX.SQL file from the midtier to the host.
 - b** Apply the SCINDEX.SQL file to rebuild indexes.

This completes the file generation process

11 Perform Pre-Upgrade Tasks for the Target Database as follows:

- a** On the z/OS host, build the JCL templates that will be used to perform the target database in-place upgrade by applying target database information to the JCL templates that were used to run the staging database in-place upgrade processes.
- b** Create and load Siebel log tables.
- c** (Optional) Apply additive schema changes.

12 Perform the In-Place Upgrade of the Target Database as follows:

- a** Drop interface tables, triggers, and stored procedures.
- b** Run PRET jobs to prepare the target database for table creation during the upgrade.
- c** Apply non-additive schema changes to the target database.
- d** Create, bind, and test the stored procedures.
- e** Run the Data Migration jobs to migrate pre-8.0 Siebel data to version 8.0 data.
There are optional data migration scripts for eChannel, Household, and Product Configurator data and for Siebel Financial Services (FINS) applications.
- f** Drop old indexes, create new 8.0 indexes, and run index rebuild jobs.

13 Upgrade the Repository and Import Seed Data as follows:

- a** Run the Database Configuration Wizard on the midtier.
- b** Enter production environment information to create the production upgrade configuration file. Specify the following values:
 - Upgrade Option: Upgrade Siebel Database Schema (upgrep + upgphys)
 - Upgrade Process: zSeries Seed / Repository Upgrade

- c Launch the Upgrade Wizard. The Upgrade Wizard automatically runs the remaining DML upgrade jobs from the midtier to perform a number of tasks, including the following:
 - Deletes the old license key.
 - Verifies the repository name and imports the New Customer Repository from the upgraded development environment.
 - Upgrades seed data to the new version.

The production database upgrade is now completed.

Key Members in a Mainframe-Centric Upgrade

Members in the *DSNHLQ.SIEBEL.EXEC* contain useful information that you can refer to when performing your development and production upgrade. These key members are listed in [Table 6](#).

Table 6. Key Members in the *DSNHLQ.SIEBEL.EXEC*

Member Name	Describes
@JOBPRFX	Shows details of the job prefixes chosen for the upgrade.
@LASTJOB	Lists the job name of the last job for each step of the upgrade, for example: <ul style="list-style-type: none"> ■ PRESCHM: S003562Z ■ PRESCHF: S006463Z
@STGTAR	Lists details of the staging and target environments.
@TBOSTG	Provides staging tableowner details.
@TBOTAR	Provides target tableowner details.
@UPGPATH	Lists the upgrade path and the upgrade status, that is, the last step executed, for example: PAUSE=4 STEP=98- TARGET UPGISS DATA MIGRATION COMPLETE - UPGISS
@UPGTYPE	Shows the type of upgrade being run, for example, SIA 7.5.3

About the JCL Upgrade Jobs

This topic provides information about the JCL jobs you run to perform the upgrade on the z/OS host.

About Job Cards

The *DSNHLQ.SIEBEL.EXEC* data set contains base job cards for all the upgrade jobs. The upgrade process uses these job cards as a base for all jobs of a particular type and then generates an individual job card for each job based on the information in the base card and in the @JOBPREFIX member of the *DSNHLQ.SIEBEL.EXEC* data set.

For example, the base job card for PRESCHM jobs is JCPRES. An individual job card is created for each PRESCHM job based on information in both of the following:

- *DSNHLQ.SIEBEL.EXEC(@JOBPREFIX)* – job prefix
- *DSNHLQ.SIEBEL.EXEC(JCPRES)* – base job card

Once job cards have been generated, changing the information in the @JOBPREFIX member does not cause job card details to change.

Job Dependencies

Some JCL jobs, for example, some Data Migration jobs, have dependencies on other jobs. These jobs are submitted in a specific order, with subsequent jobs only being submitted when the current job has completed successfully. The @DEPFLOW member in the relevant JCL data set shows the order in which the jobs are to be submitted, for example, see *DSNHLQ.SIEBEL.PRESCHM.JCL(@DEPFLOW)* for information about the submission sequence for PRESCHM jobs.

Load and Unload Jobs

Load and Unload job procedures are located in the *DSNHLQ.SIEBEL.PROC* data set.

About the Override File

Upgrades: Releases 6.x, 7.0.x, 7.5.x.

Environments: Development environment only.

The override.inp file allows you to override your existing database storage layout when you upgrade to Release 8.0.

Since Release 7.7, the Siebel database schema structure has the following characteristics:

- One table in each table space
- One table space in each database

This layout is referred to as the *1:1:1* model. See [“Planning Changes to the Physical Layout of the Schema” on page 74](#) for further information.

If you are upgrading from a pre 7.7 release of Siebel applications, you may want to switch to the *1:1:1* model from your existing database schema structure because it is more efficient. There are three options available when upgrading to the current release:

- Preserve the existing model of multiple tables in each table space and only build new tables with the *1:1:1* model.

To use this option, extract a storage control file and merge it with an existing template. All existing definitions are preserved and new tables are built according to the definitions supplied in the template storage control file.

- Preserve the existing model for most tables but indicate that specific tables are to use the *1:1:1* model by adding the table names to the `override.inp` file. By default, the `override.inp` file has two table entries:
 - `S_DOCK_TXN_LOG`
 - `S_SERVICE_SCRIPT`

The existing definition for these two tables is *not* preserved during the upgrade. Therefore, these two tables can be built using the new model as specified in the template.

- Move all tables from the current model to the *1:1:1* model.

To use this option, do *not* use the extract and merge process to create the storage control file. Instead, use the preconfigured storage control file. In the preconfigured storage control file, all tables are created using the *1:1:1* model. Therefore, each table is created in its own database or table space.

The use of the preconfigured storage control file allows you to use the *1:1:1* model for all tables other than obsolete tables and customer extended tables. The preconfigured storage control file does not apply to obsolete or customer extended tables so there are no entries for them. You cannot put these into the `override.inp` file since there is nothing to override and the tables are not in the storage control file. In this case, the existing schema is preserved.

The table spaces are not cleaned up unless they are empty. Some of the obsolete and customer extended tables are still in those table spaces so the upgrade process cannot just do a general cleanup.

About the Siebel Database Configuration Utilities

Upgrades: All upgrades.

Environments: All environments.

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

The Siebel Database Configuration Utilities are a set of files and scripts that provide inputs to the upgrade executables.

For best performance, install the Siebel Database Configuration Utilities files on the Siebel Server that you will use to perform the upgrade. The Siebel Database Configuration Utilities files will be installed at the same directory level as the Siebel Server in a directory called `dbsrvr`. For example, if the Siebel Server is installed in `C:\sba80\si_ebsrvr` (Windows), then the Siebel Database Configuration Utilities will be installed in `C:\sba80\dbsrvr`. For further information, see *Implementing Siebel Business Applications on DB2 for z/OS*.

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

4

How to Perform a Siebel Database Upgrade

This chapter provides a roadmap for performing each type of upgrade to Siebel 8.0. Each roadmap lists the processes and tasks you must follow to complete the upgrade. Print the relevant roadmap and use it to guide you in carrying out the upgrade. This chapter includes the following topics:

- [Roadmap for Performing a Siebel Database Upgrade on page 47](#)
- [Process of Planning a Siebel Database Upgrade on page 48](#)
- [Process of Upgrading a Siebel Development Environment on page 49](#)
- [Process of Upgrading a Production Test Environment on page 57](#)
- [Process of Tuning the Upgrade Performance on page 64](#)
- [Process of Upgrading a Siebel Production Environment on page 66](#)

Related Topics

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

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

Roadmap for Performing a Siebel Database Upgrade

Upgrades: All upgrades.

Environments: All environments.

Use one of the roadmaps listed below to guide you through the steps for upgrading your Siebel database:

- ["Roadmap for Upgrading from Siebel Financial Services 6.2.1 and Siebel 7.x" on page 48](#)
- ["Roadmap for Upgrading to Siebel 8.0 Without a Development Environment" on page 48](#)

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

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

Roadmap for Upgrading from Siebel Financial Services 6.2.1 and Siebel 7.x

If you are upgrading from Release 6.2.1 of Siebel Financial Services or Release 7.x, complete the processes in this roadmap in the order shown:

- 1 [“Process of Planning a Siebel Database Upgrade” on page 48](#)
- 2 [“Process of Upgrading a Siebel Development Environment” on page 49](#)
- 3 [“Process of Upgrading a Production Test Environment” on page 57](#)
- 4 [“Process of Tuning the Upgrade Performance” on page 64](#)
- 5 [“Process of Upgrading a Siebel Production Environment” on page 66](#)

For a description of the differences between production and development upgrades, see [“About Siebel Upgrade Environments” on page 21](#).

Roadmap for Upgrading to Siebel 8.0 Without a Development Environment

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

Upgrading without a development environment means the following are true:

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

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

- 1 [“Process of Planning a Siebel Database Upgrade” on page 48](#).
- 2 [“Preparing for a No-Development-Environment Siebel Upgrade” on page 119](#).
- 3 Perform a production test upgrade. See [“Process of Upgrading a Production Test Environment” on page 57](#).
- 4 [“Process of Tuning the Upgrade Performance” on page 64](#).
- 5 Upgrade your production environment. See [“Process of Upgrading a Siebel Production Environment” on page 66](#).

Process of Planning a Siebel Database Upgrade

Upgrades: All upgrades.

Environments: All environments.

This process is part of an upgrade roadmap. See [“Roadmap for Upgrading from Siebel Financial Services 6.2.1 and Siebel 7.x” on page 48](#) and [“Roadmap for Upgrading to Siebel 8.0 Without a Development Environment” on page 48](#).

To plan the upgrade, read the following:

- 1 [Chapter 3, “How the Siebel Database Upgrade Works”](#) and relevant topics in the chapter of the *Siebel Database Upgrade Guide* that describes how the Siebel database upgrade works.
- 2 [Chapter 5, “Planning A Siebel Database Upgrade”](#) and relevant topics in the chapter of the *Siebel Database Upgrade Guide* that describes Siebel database and UI upgrade planning.
- 3 The chapter of the *Siebel Database Upgrade Guide* that describes application planning for a Siebel upgrade.
- 4 The appendix in the *Siebel Database Upgrade Guide* that contains reference information on Siebel Marketing upgrades.
- 5 The appendix in the *Siebel Database Upgrade Guide* that lists tables modified or seeded during a Siebel upgrade.
- 6 Review Siebel upgrade-related publications on Oracle Technology Network or My Oracle Support.

Process of Upgrading a Siebel Development Environment

Upgrades: All upgrades.

Environments: Development environment only.

This process is part of a roadmap. See [“Roadmap for Performing a Siebel Database Upgrade” on page 47](#).

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

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

Steps that apply only to upgrades from Release 6.2.1 are marked as such.

Review Siebel Technical Notes and Alerts on My Oracle Support

- 1 Check My Oracle Support for Siebel Alerts and Technical Notes regarding upgrade.
- 2 Review the chapter in *Siebel Database Upgrade Guide* that describes Siebel database and UI upgrade planning. This chapter lists important publications and resources for performing an upgrade.

Upgrade Third-Party Software

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

Upgrade the Servers

Verify you have identified all the maintenance releases, Fix Packs, and quick-fix patches required for the upgrade. These requirements are documented in *Siebel Maintenance Release Guide* 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 and *Implementing Siebel Business Applications on DB2 for z/OS*:

- 4 Install the Siebel 8.0 Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE).

The upgraded Siebel Servers will not work correctly with the RDBMS server until after you have upgraded the Siebel database to the new release.
- 5 Install the Siebel Database Configuration Utilities files on the Siebel Server you will use to perform the upgrade.
- 6 Install language packs for your currently deployed languages and any new languages.
- 7 If you have customized the configuration of Enterprise components, such as Siebel Servers, you must migrate the customizations to the upgraded environment. See *Going Live with Siebel Business Applications*.

Upgrade the RDBMS

- 8 If required, upgrade your version of DB2 for z/OS. Refer to the IBM documentation to perform the upgrade. For information on supported RDBMS systems and versions, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

Identify and Resolve Duplicate Row IDs

- 9 **6.2.1 upgrades only:** ["Identifying and Resolving Duplicate Row IDs" on page 103.](#)

After you install the Siebel Database Configuration Utilities software, but before you upgrade to this release, you must identify and resolve any duplicate row IDs in your Siebel database.

Preupgrade Tasks for the Siebel Database

- 10 Review guidelines for configuring DB2 for z/OS. See *Implementing Siebel Business Applications on DB2 for z/OS*.

- 11 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 12 Stop the Siebel Servers.
- 13 Prepare the storage control file to use in the upgrade. See [“Preparing the Storage Layout of the Schema” on page 88](#).
- 14 Perform relevant tasks in:
 - [Chapter 6, “Basic Database Preparations for a Siebel Upgrade”](#)
 - The chapter in the *Siebel Database Upgrade Guide* that describes basic database preparations for a Siebel upgrade
- 15 Migrating Siebel Repository Objects to the Standard UI. For information on this topic, see the chapter in the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge.

Preupgrade Tasks for a Development Environment Upgrade

- 16 Perform the tasks in [Chapter 8, “Preparing a Development Environment for a Siebel Upgrade.”](#)
- 17 Rename the Siebel Repository. For information on this step, see the chapter in the *Siebel Database Upgrade Guide* that describes upgrading the Siebel database.

Preupgrade Tasks for Application Data

- 18 Perform the tasks described in the following documents:
 - 477519.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 521.
 - 477254.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 481.
- 19 Perform the relevant tasks in:
 - [Chapter 7, “Preparing Siebel Application Data for Upgrade”](#)
 - The chapter of the *Siebel Database Upgrade Guide* that describes how to prepare Siebel application data for upgrade.

Some of these tasks are optional, depending on the currently installed Siebel products and your upgrade path. Review and perform these tasks as necessary.

Preparing Developers for the Upgrade

- 20 Back up the development database.
- 21 Verify that all developers have checked in their projects and that all projects are unlocked.
- 22 Verify that all developers have disconnected from the database. The only open connection should be the account that is performing the upgrade.

- 23 Install the new Siebel Tools and license keys on development workstations. Keep at least one copy of the previous version of Siebel Tools. You will need it to perform repository operations before the repository merge.
- 24 Perform all remaining tasks with the new Siebel Tools, unless stated otherwise.

Upgrade Siebel Database Schema (upgrep)

- 25 (Optional) Changing the Siebel Database Configuration Wizard Language. For information on this topic, see the chapter of the *Siebel Database Upgrade Guide* that describes how to upgrade the Siebel database.
- 26 Create the staging database by performing the following tasks:
 - a "Required Tasks before Creating the Staging Database" on page 137
 - b "Creating the Staging Database Schema DDL Files" on page 138.
 - c "Transferring the Staging DDL to the z/OS Host" on page 139.
 - d "Preparing the z/OS Host Upgrade Environment and Creating the Staging Database" on page 140.
 - e "Removing Interface Tables and Triggers" on page 145.
- 27 Generate the Upgrade Files by performing the following tasks:

NOTE: Edit the generated files as required by Siebel Technical Notes, Alerts, Release Notes or other publications before transferring them to the z/OS host.

 - a "Prerequisite Tasks for Generating the Upgrade Files" on page 150
 - b "Restarting the Siebel Upgrade Wizard After Pauses" on page 134
 - c "Preparing the Additive Schema and JCL Files on the z/OS Host" on page 151.
 - d "Restarting the Siebel Upgrade Wizard After Pauses" on page 134
 - e "Preparing and Executing the Non-Additive Schema and JCL Files on the z/OS Host" on page 158.
 - f "Restarting the Siebel Upgrade Wizard After Pauses" on page 134
 - g "Processing the Index Schema File" on page 162.
 - h "Building JCL Templates for the Target Database" on page 164.
- 28 Upgrade the target database by performing the following tasks:
 - a "Creating and Loading Siebel Log Tables" on page 168
 - b "Process of Upgrading the Target Database" on page 167
 - c "Performing the In-Place Target Database Upgrade" on page 170
- 29 Upgrade the repository and import seed data. See "Upgrading the Repository and Importing Seed Data" on page 135.
- 30 "About the Siebel Upgrade Log Files" on page 239.
- 31 If the upgrade contains unacceptable errors, do the following:

- a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Configuration Wizard.
- 32 [“Installing New License Keys After Upgrade” on page 237.](#)
- 33 **6.2.1 upgrades only:** If you have multilingual deployments, perform the steps in 477094.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 447. This document describes how to import language-specific repository strings and seed data into the upgrade repositories.
- 34 Back up the upgraded database repository.

Prepare for Repository Merge

- 35 Perform relevant tasks in the chapter of the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge.
- NOTE:** Review the topic that describes how to configure Siebel Repository objects to inherit upgrade behavior.
- 36 [“About Backing Up the New Customer Repository or Database Schema” on page 185.](#)
- 37 **6.2.1 upgrades only:** [“Running the Repository Preparation Wizard on a Release 6.2.1 Repository” on page 187.](#)
- 38 Execute the REORG utility on tables that receive a large number of inserts during the repository import process. For further information about reorganizing tables, see [“About Reorganizing Tables Before the Repository Merge” on page 186.](#)
- 39 Run database statistics on the Siebel database. For further information on running database statistics, see [“Generating RUNSTATS Jobs” on page 182.](#) Running statistics on the Siebel database improves merge performance.

If upgrading from Release 7. 7 or Release 7.8.x, run statistics specifically on the S_SYM_STR and SYM_STR_INT tables. If you are upgrading from a pre-7.7 release, the S_SYM_STR and SYM_STR_INT tables are not populated until the merge is completed so you do not need to run statistics on them at this point.

Perform Repository Merge

CAUTION: The Repository merge process cannot be stopped and restarted so make sure you have backed up the database schema or the New Customer Repository before starting the merge.

- 40 [“Performing a Siebel Repository Merge” on page 189.](#)
- 41 Review the Siebel Repository merge logs. For information on this task, see the chapter of the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge.
- 42 If the repository merge contains unacceptable errors, do the following:
- a Restore the backup of the database or New Customer Repository.
 - b Correct the errors.

c Rerun the repository merge.

43 Review Siebel Repository Object Property Conflicts.

For information on this task, see the chapter of the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge.

44 Run the Siebel Postmerge Utilities.

For information on this task, see the chapter of the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge.

45 Generate Siebel EIM Temporary Columns.

For information on this task, see the chapter of the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge.

46 Generate database statistics for new tables and new indexes using the RUNSTATS utility. Your DBA can check the system catalog to determine the database objects that do not have statistics.

For further information on generating database statistics, see [“Generating RUNSTATS Jobs” on page 182](#).

47 Back up the Siebel database.

Migrate 6.2.1 Customizations

6.2.1 upgrades only: The steps in [“Migrate 6.2.1 Customizations”](#) apply only to upgrades from Release 6.2.1.

48 [“Identifying What Will Be Migrated” on page 211.](#)

49 [“Exposing Hidden Properties of Applets and Views” on page 212.](#)

50 [“Running the Siebel Web Client Migration Wizard” on page 213.](#)

51 [“Reviewing Applet and View Migration” on page 216.](#)

52 [“Running the Web Layout Wizard” on page 217.](#)

53 [“Migrating Scripts Attached to Controls” on page 218.](#)

54 [“Migrating Scripts Attached to Applets” on page 220.](#)

55 [“Migrating Business Component, Business Service, and Application Scripts” on page 223.](#)

56 [“Migrating Outbound COM Interfaces” on page 224.](#)

57 [“Migrating Inbound COM Interfaces” on page 225.](#)

58 [“Regenerating the Postmerge Utilities Log” on page 228.](#)

59 [“Reviewing Customized Business Components” on page 228.](#)

Upgrade Custom Database Schema (upgphys)

60 Run the Database Configuration Wizard to extract the storage control file from the target database as follows:

- Specify the following values when prompted to do so:
 - **Extract Options:** Extract from Catalog
 - **Storage Control File:** Specify the name of the storage control file to be extracted, for example, storage_postupg.ctl
- Make sure you specify values for the target database when prompted for the names of the schema qualifier, ODBC data source, and database user name and password.

The procedure to extract a storage control file using the Extract from catalog option is described in *Implementing Siebel Business Applications on DB2 for z/OS*. The extracted target database storage control file is used as input to the upgphys upgrade process.

- 61 Remove EIM tables using the procedure described in [“Removing Interface Tables and Triggers” on page 145](#). It is recommended that you remove EIM tables before synchronizing the Siebel logical and physical schema ([Step 64 on page 55](#)) to ensure that the synchronization process does not fail when processing EIM tables.
- 62 Run the Database Configuration Wizard to complete the development environment upgrade:
 - Choose the following settings:
 - **Upgrade Options:** Upgrade Custom Database Schema (upgphys)
 - **Environment Type:** Development
 - **Storage Control File:** Storage control file that you generated in [Step 60](#).
 - Make sure you specify values for the target database when prompted for the names of the schema qualifier, ODBC data source, and database user name and password.

Launch the Siebel Upgrade Wizard. SQL commands are executed on the development environment database and a number of output files are generated.

- 63 [“Transferring the Development Environment Upgrade Output Files to the z/OS Host” on page 231](#).
- 64 [“Synchronizing the Schema” on page 232](#).
- 65 [“Migrating Custom Business Component Configurations” on page 233](#).
- 66 [“About the Siebel Upgrade Log Files” on page 239](#).
- 67 If the upgrade contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Configuration Wizard.
- 68 [“Manually Archiving Upgrade Log Files” on page 242](#).
- 69 Back up the upgraded database.
- 70 [“Deleting Redundant Upgrade Files” on page 237](#).

Initialize and Extract Developers’ Local Databases

- 71 Install the new release of Siebel Tools on developers’ machines.

- 72** Initialize and extract the local database on development machines. For more information about this task, see *Developing and Deploying Siebel Business Applications*.

Review the User Interface

- 73** (Optional). Compile an SRF file to help review the UI. See [“Creating a New SRF File” on page 200](#).
- 74** If you customized style sheet or web template files in the previous release, you have to implement those customizations in the new release again, if desired.
- Carefully review the UI in the new release before implementing customizations to these files.
- 75** Perform the relevant tasks in the chapter of the *Siebel Database Upgrade Guide* that describes reviewing the Siebel user interface.
- 76** 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. Convert remaining flow based applets as desired. For further information on editing applet layout, see *Configuring Siebel Business Applications*.

Postmerge Development Tasks

- 77** Perform the relevant tasks in the chapter of the *Siebel Database Upgrade Guide* that describes the Siebel postmerge development tasks.
- 78** Resolve any business component and join conflicts.

Postupgrade Tasks for Database and File System

- 79** Perform the applicable tasks in:
- [Chapter 19, “Postupgrade Tasks for Siebel Database and File System”](#)
 - The chapter of the *Siebel Database Upgrade Guide* that describes postupgrade tasks for the Siebel database and file system.
- 80** Perform the tasks in [Chapter 18, “Postupgrade Tasks for a Siebel Development Environment.”](#)
- 81** 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 and *Implementing Siebel Business Applications on DB2 for z/OS* for recommended parameter settings.
- 82** If you exported data from interface tables before the upgrade, review the database and import the data as desired.
- 83** Upgrading to Siebel RC2 or AES Encryption. See the chapter of the *Siebel Database Upgrade Guide* that describes Siebel database and UI upgrade planning.

The development environment is now upgraded. The remaining sections deal with configuration and validation tasks.

Postupgrade Tasks for Applications Configuration

- 84 If applicable, review the results of the Person and Organization merge. Make configuration changes as required. For further information, see the topic on the Party Model in the chapter of the *Siebel Database Upgrade Guide* that describes Siebel database and UI upgrade planning.
- 85 Perform the applicable tasks in:
 - Chapter 20, “Postupgrade Tasks for Siebel Applications”
 - The chapter of the *Siebel Database Upgrade Guide* that describes the postupgrade tasks for Siebel applications.
- 86 Verify the function of interfaces in integrated applications.
- 87 Activate and deploy workflows. To perform these tasks, see *Siebel Business Process Framework: Workflow Guide*.
- 88 If you have set up integrations for transferring data to or from third-party applications using Siebel EAI, verify the integrations are configured correctly. For information on using EAI, see *Overview: Siebel Enterprise Application Integration*.
- 89 If you have used EIM to set up batch processing jobs, verify EIM is configured correctly. For information on using EIM, see *Siebel Enterprise Integration Manager Administration Guide*.

Perform System Tests

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

Prepare for Transition to Production Test Environment

- 91 Create a new SRF file. For information on creating an SRF file, see *Developing and Deploying Siebel Business Applications*.
- 92 If you revised the repository after running `upgphys`, you must regenerate the repository definition files. See the chapter of the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge for information on regenerating Siebel repository definition files.

Process of Upgrading a Production Test Environment

Upgrades: All upgrades.

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

This process is part of a roadmap. See [“Roadmap for Performing a Siebel Database Upgrade” on page 47](#).

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

NOTE: The production test environment must replicate the production environment exactly.

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

Review Siebel Technical Notes and Alerts on My Oracle Support

- 1 Check My Oracle Support for Siebel Alerts and Technical Notes regarding upgrade.
- 2 Review the chapter in *Siebel Database Upgrade Guide* that describes Siebel database and UI upgrade planning. This chapter lists important publications and resources for performing an upgrade.

Upgrade Third-Party Software

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

Upgrade the Servers

Verify you have identified all the maintenance releases, Fix Packs, and quick-fix patches required for the upgrade. These requirements are documented in *Siebel Maintenance Release Guide* 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 and *Implementing Siebel Business Applications on DB2 for z/OS*. Do the following:

- 4 Install the Siebel 8.0 Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE).

The upgraded Siebel Servers will not work correctly with the RDBMS server until after you have upgraded the Siebel database to the new release.

- 5 Install the Siebel Database Configuration Utilities files on the Siebel Server you will use to perform the upgrade.
- 6 Install language packs for your currently deployed languages and any new languages.

- 7 If you have customized the configuration of Enterprise components, such as Siebel Servers, you must migrate the customizations to the upgraded environment. See *Going Live with Siebel Business Applications*.

Upgrade the RDBMS

- 8 If required, upgrade your version of DB2 for z/OS. Refer to the IBM documentation to perform the upgrade. For information on supported RDBMS systems and versions, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

Identify and Resolve Duplicate Row IDs

- 9 **6.2.1 upgrades only:** [“Identifying and Resolving Duplicate Row IDs” on page 103.](#)

After you install the Siebel Database Configuration Utilities software, but before you upgrade to this release, you must identify and resolve any duplicate row IDs in your Siebel database.

Preupgrade Tasks for the Siebel Database

- 10 Review guidelines for configuring DB2 for z/OS. See *Implementing Siebel Business Applications on DB2 for z/OS*.
- 11 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 12 Prepare the storage control file to use in the upgrade. See [“Preparing the Storage Layout of the Schema” on page 88.](#)
- 13 Perform the relevant tasks in:
 - [Chapter 6, “Basic Database Preparations for a Siebel Upgrade”](#)
 - The chapter of the *Siebel Database Upgrade Guide* that describes the basic database preparations for a Siebel upgrade.
- 14 Before starting the production test upgrade, ensure that:
 - Siebel database transactional tables with clustering indexes are organized in cluster sequence.
 - Generate database statistics for your Siebel database transactional tables, if required.

Preupgrade Tasks for Application Data

- 15 Perform the tasks in 477519.1 (Article ID) on My Oracle Support. This document was formerly published as Siebel Technical Note 521.
- 16 Perform the relevant tasks in the following:
 - [Chapter 7, “Preparing Siebel Application Data for Upgrade”](#)

- The chapter of the *Siebel Database Upgrade Guide* that describes how to prepare Siebel application data for upgrade.

Some of these tasks are optional, depending on the currently installed Siebel products and your upgrade path. Review and perform these tasks as necessary.

Preupgrade Tasks for a Production Test Environment Upgrade

17 For instructions on how to set up the Siebel database and Siebel Servers in the production test environment, refer to 477772.1 (Article ID) on My Oracle Support. This document was formerly published as Siebel Technical Note 586.

18 Stop the Siebel Servers.

19 Close all database connections. The only database connection should be the account performing the upgrade.

Disconnect the Siebel Server from the development environment database and connect it to the production test environment database.

20 Copy application files to the environment:

- Custom SRF file.
- Reports files.
- Custom Web templates and style sheets.

See the topic on Copying UI Files to a New Siebel Environment in the chapter of the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge.

21 [“About Moving the Customized Repository and Schema Definition Files” on page 117.](#)

If you revised repository objects or schema definitions after performing your development environment upgrade, regenerate the schema.ddl and custrep.dat files before transferring them to the production test environment. See [“Regenerating the Siebel Repository Definition Files” on page 198.](#)

22 Preparing for a No-Development-Environment Siebel Upgrade.

If you do not have a development environment, see [“Preparing for a No-Development-Environment Siebel Upgrade” on page 119](#) for information on this task.

23 Verify the production test database is either a copy of the current production database or has the same topology and a similar amount of data. This is important for effective performance testing of the upgrade scripts.

24 Back up the production test environment database. To do upgrade tuning, you will restore this database and perform test-upgrades on it.

Prepare for Production

25 (Optional) Change the Siebel Database Configuration Wizard language. For information on this topic, see the chapter of the *Siebel Database Upgrade Guide* that describes how to upgrade the Siebel database.

- 26** Run the Database Configuration Wizard as described in [“About Running the Database Configuration Wizard Under Windows” on page 126](#) or [“About Running the Database Configuration Wizard Under UNIX” on page 129](#). Do the following:
- a** Specify the following options:
 - Upgrade Options:** Prepare for Production Upgrade
 - Environment Type:** Production
 - b** Launch the Siebel Upgrade Wizard.

Upgrade the Siebel Database Schema (upprep + upgphys)

- 27** Create the staging database by performing the following tasks:

- a** [“Required Tasks before Creating the Staging Database” on page 137](#)
- b** [“Creating the Staging Database Schema DDL Files” on page 138](#).
- c** [“Transferring the Staging DDL to the z/OS Host” on page 139](#).
- d** [“Preparing the z/OS Host Upgrade Environment and Creating the Staging Database” on page 140](#).
- e** [“Removing Interface Tables and Triggers” on page 145](#).

- 28** Generate the upgrade files by performing the following tasks:

NOTE: Edit the generated files as required by Siebel Technical Notes, Alerts, Release Notes or other publications before transferring them to the z/OS host.

- a** [“Prerequisite Tasks for Generating the Upgrade Files” on page 150](#)
- b** [“Restarting the Siebel Upgrade Wizard After Pauses” on page 134](#)
- c** [“Preparing the Additive Schema and JCL Files on the z/OS Host” on page 151](#)
- d** [“Applying the Additive Schema Changes to the Production Staging Database” on page 155](#)
- e** [“Restarting the Siebel Upgrade Wizard After Pauses” on page 134](#)
- f** [“Preparing and Executing the Non-Additive Schema and JCL Files on the z/OS Host” on page 158](#)
- g** [“Restarting the Siebel Upgrade Wizard After Pauses” on page 134](#)
- h** [“Processing the Index Schema File” on page 162](#)
- i** [“Building JCL Templates for the Target Database” on page 164](#)

- 29** Upgrade the target database by performing the following tasks:

- a** Read [“Process of Upgrading the Target Database” on page 167](#)
- b** [“Creating and Loading Siebel Log Tables” on page 168](#)
- c** [“Applying Additive Upgrade Changes to the Target Database” on page 168](#)

- d Generate statistics for tables and indexes that were created or rebuilt when additive schema changes were applied to the database. You may also need to reorganize these tables and indexes.

For further information on generating database statistics, see [“Generating RUNSTATS Jobs” on page 182](#).

- e [“Renaming the Production Environment Repository” on page 118](#)

- f [“Performing the In-Place Target Database Upgrade” on page 170](#)

- 30 Extract the storage control file from the target database using the Database Configuration Wizard as follows:

- Specify the following values when prompted to do so:

- **Extract Options:** Extract from Catalog.

- **Storage Control File:** Specify the name of the storage control file to be extracted, for example, storage_postupg.ctl.

- Make sure you specify values for the target database when prompted for the names of the schema Qualifier, ODBC data source, and database user name and password.

The procedure to extract a storage control file using the Extract from catalog option is described in *Implementing Siebel Business Applications on DB2 for z/OS*. The extracted target database storage control file is used as input to the upgphys upgrade process.

- 31 Upgrade the repository and import seed data. See [“Upgrading the Repository and Importing Seed Data” on page 135](#).

- 32 [“Performing Intersection Table Maintenance” on page 234](#)

- 33 [“About the Siebel Upgrade Log Files” on page 239](#).

- 34 If the upgrade contains unacceptable errors, do the following:

- a Restore the backup of the database.

- b Correct the errors.

- c Rerun the Database Configuration Wizard.

- 35 **6.2.1 upgrades only:** If you have multilingual deployments, perform the steps in 477094.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 447. This document describes how to import language-specific repository strings and seed data into the upgrade repositories.

- 36 [“Manually Archiving Upgrade Log Files” on page 242](#).

- 37 Back up the upgraded database.

Postupgrade Tasks for Database and File System

- 38 Perform the relevant tasks in the following:

- [Chapter 19, “Postupgrade Tasks for Siebel Database and File System,”](#)

- The chapter of the *Siebel Database Upgrade Guide* that describes the postupgrade tasks for the Siebel database and file system.
- 39 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 and *Implementing Siebel Business Applications on DB2 for z/OS* for recommended parameter settings.
- 40 If you exported data from interface tables before the upgrade, review the database and import the data as desired.
- 41 Upgrading to Siebel RC2 or AES Encryption. See the chapter of the *Siebel Database Upgrade Guide* that describes the Siebel database and UI upgrade planning tasks.
- 42 Ensure Siebel database transactional tables with clustering indexes are organized in cluster sequence, and check whether you need to run the REORG utility on new tables.
- 43 Generate database statistics for new or rebuilt tables and indexes using the RUNSTATS utility. For more information about running statistics, see [“Generating RUNSTATS Jobs” on page 182](#).

The production test environment is now upgraded. The remaining sections deal with configuration and validation tasks.

Postupgrade Tasks for Applications Configuration

- 44 Perform the relevant tasks in the following to prepare for system testing:
 - [Chapter 20, “Postupgrade Tasks for Siebel Applications”](#)
 - The chapter of the *Siebel Database Upgrade Guide* that describes the postupgrade tasks for Siebel applications.
- 45 Verify the function of interfaces in integrated applications.
- 46 Activate and deploy workflows. To perform these tasks, see *Siebel Business Process Framework: Workflow Guide*.
- 47 If you have set up integrations for transferring data to or from third-party applications using Siebel EAI, verify the integrations are configured correctly. For information on using EAI, see *Overview: Siebel Enterprise Application Integration*.
- 48 If you have used EIM to set up batch processing jobs, verify EIM is configured correctly. For information on using EIM, see *Siebel Enterprise Integration Manager Administration Guide*.
- 49 If you customized style sheet or web template files in the previous release, you must implement those customizations in the new release again, if desired.

Carefully review the UI in the new release before implementing customizations to those files.

Perform System Tests

- 50 Use available test data to perform unit testing. Validate application function in the following areas:
 - User interface

- Data interfaces
- Integrity of migrated data
- Workflow function

Process of Tuning the Upgrade Performance

Upgrades: All upgrades.

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

This process is optional.

This process is part of a roadmap. See [“Roadmap for Performing a Siebel Database Upgrade” on page 47](#).

CAUTION: You are required to contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services for help with tuning your upgrade scripts. If you do not, you may invalidate your support agreement.

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. For help with this process and to implement more advanced tuning, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

You can also use this process to test the additive schema changes feature to verify that it does not adversely affect application functionality. The additive schema changes 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.

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

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

- 478308.1(Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 616. This document describes strategies for minimizing production environment downtime during an upgrade. The steps outlined in this topic are intended primarily for use with the baseline best practices described in Technical Note 616.
- [Chapter 21, “Tuning the Siebel Production Upgrade Scripts.”](#) This chapter provides information on how you can improve the performance of the production environment upgrade by tuning the production upgrade scripts in a production test environment.

Set Up the Target Database

- 1 Back up and remove the upgraded production test 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 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 is configured for optimum upgrade performance. Review the relevant topics in the following:
 - [Chapter 6, “Basic Database Preparations for a Siebel Upgrade”](#)
 - The chapter of the *Siebel Database Upgrade Guide* that describes the basic database preparations for a Siebel upgrade.
- 5 (Optional) Run statistics on the target database if the catalog statistics need to be updated.
- 6 Perform relevant tasks in the following:
 - [Chapter 7, “Preparing Siebel Application Data for Upgrade”](#)
 - The chapter of the *Siebel Database Upgrade Guide* that describes how to prepare Siebel application data for upgrade.

Upgrade the Target Database Schema (upgrep + upgphys)

- 7 Using the upgrade files you generated during the production test upgrade, upgrade the target database by performing the following tasks:
 - a [“Creating and Loading Siebel Log Tables” on page 168](#)
 - b [“Applying Additive Upgrade Changes to the Target Database” on page 168](#)
 - c [“Performing the In-Place Target Database Upgrade” on page 170](#)
- 8 Upgrade the repository and import seed data. See [“Upgrading the Repository and Importing Seed Data” on page 135](#). Specify the storage control file you extracted when you performed the production test upgrade (see [Step 30 on page 62](#)).
- 9 [“Performing Intersection Table Maintenance” on page 234](#).
- 10 Note the time required to upgrade the database.
- 11 Review the upgrade logs for errors. See [“About the Siebel Upgrade Log Files” on page 239](#).
- 12 If the upgrade contains errors that prevented completion or adversely affected performance, correct the errors and rerun the upgrade.
- 13 [“Manually Archiving Upgrade Log Files” on page 242](#).

Tune the Upgrade Files

- 14 Evaluate upgrade performance, particularly the time required to complete the upgrade.
- 15 Do one of the following:
 - If the time required to complete the upgrade is acceptable, no further tuning is needed. Perform the steps in [“Process of Upgrading a Siebel Production Environment” on page 66](#).

- If the time required to complete the upgrade is too long, perform the remaining steps in this section to continue tuning upgrade performance.
 - If the time required to complete the upgrade is too long and you cannot tune further, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services to apply advanced tuning.
- 16** Carefully review target database and database server configuration. Adjust as needed to further improve upgrade performance.
- 17** Tune the upgrade files:
- [“Optimizing Unload and Load Job Performance” on page 266.](#)
 - [“Adding the Statistics Clause to Load Cards” on page 267.](#)
- 18** Copy the tuned upgrade scripts to a safe location for use in the production upgrade.

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.

- 19** 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.
- 20** In the production test environment, perform another test upgrade and evaluate upgrade performance.
- 21** Repeat the tuning process and perform test-upgrades until upgrade performance is acceptable.
- 22** When you have completed tuning upgrade performance in the production test environment, delete and remove the target database.

Process of Upgrading a Siebel Production Environment

Upgrades: All upgrades.

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

This process is part of a roadmap. See [“Roadmap for Performing a Siebel Database Upgrade” on page 47.](#)

This topic lists the tasks required to transition your production test environment to production. Print this topic and use it as a checklist for doing the upgrade.

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

Review Siebel Technical Notes and Alerts on My Oracle Support

- 1 Check My Oracle Support for Siebel Alerts and Technical Notes regarding upgrade.
- 2 Review the chapter in *Siebel Database Upgrade Guide* that describes Siebel database and UI upgrade planning. This chapter lists important publications and resources for performing an upgrade.

Upgrade Third-Party Software

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

Upgrade the Servers

Verify you have identified all the maintenance releases, Fix Packs, and quick-fix patches required for the upgrade. These requirements are documented in *Siebel Maintenance Release Guide* 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 and *Implementing Siebel Business Applications on DB2 for z/OS*.

- 4 Install the Siebel 8.0 Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE).

The upgraded Siebel Servers will not work correctly with the RDBMS server until after you have upgraded the Siebel database to the new release.
- 5 Install the Siebel Database Configuration Utilities files on the Siebel Server you will use to perform the upgrade.
- 6 Install language packs for your currently deployed languages and any new languages.
- 7 If you have customized the configuration of Enterprise components, such as Siebel Servers, you must migrate the customizations to the upgraded environment. See *Going Live with Siebel Business Applications*.

Upgrade the RDBMS

- 8 If required, upgrade your version of DB2 for z/OS. Refer to the IBM documentation to perform the upgrade. For information on supported RDBMS systems and versions, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

Identify and Resolve Duplicate Row IDs

9 6.2.1 upgrades only: [“Identifying and Resolving Duplicate Row IDs” on page 103.](#)

After you install the Siebel Database Configuration Utilities software, but before you upgrade to this release, you must identify and resolve any duplicate row IDs in your Siebel database.

Preupgrade Tasks for the Siebel Database

- 10** Review guidelines for configuring DB2 for z/OS. See *Implementing Siebel Business Applications on DB2 for z/OS*.
- 11** Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 12** Stop the Siebel Servers.
- 13** Verify there are no open database connections.
- 14** Prepare the storage control file to use in the upgrade. See [“Preparing the Storage Layout of the Schema” on page 88.](#)
- 15** Perform the applicable tasks in the following:
 - [Chapter 6, “Basic Database Preparations for a Siebel Upgrade”](#)
 - The chapter of the *Siebel Database Upgrade Guide* that describes the basic database preparations for a Siebel upgrade.

Preupgrade Tasks for Application Data

- 16** Perform the tasks in 477519.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 521.
- 17** Perform the relevant tasks in the following:
 - [Chapter 7, “Preparing Siebel Application Data for Upgrade”](#)
 - The chapter of the *Siebel Database Upgrade Guide* that describes how to prepare Siebel application data for upgrade.

Some of these tasks are optional, depending on the currently installed Siebel products and your upgrade path. Review and perform these tasks as necessary.

Preupgrade Tasks for a Production Environment Upgrade

- 18** Copy application files to the environment:
 - a** Custom SRF file.
 - b** Reports files.
 - c** Custom Web templates and style sheets. See the topic on Copying UI Files to a New Siebel Environment in the chapter of the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge.

- 19 [“About Moving the Customized Repository and Schema Definition Files” on page 117.](#)
- 20 Preparing for a No-Development-Environment Siebel Upgrade. If you do not have a development environment, see [“Preparing for a No-Development-Environment Siebel Upgrade” on page 119](#) for information on this task.

Upgrade the Siebel Database Schema (upgprep + upgphys)

If you have completed a production test upgrade and have tuned the SQL and JCL upgrade files on the z/OS host, you can use these files to perform the target database upgrade in the production environment. This approach has several advantages:

- You do not have to generate upgrade files in the production environment and then manually transfer customizations to them from the production test environment.
- You do not have to run the Database Configuration Wizard in Prepare for Production mode again.
- You do not have to run the zSeries Staging of Files for Upgrade process using the Database Configuration Wizard to create the staging database and to generate upgrade files again.
- With some exceptions, you do not have to perform database-related configuration tasks required by Release Notes or Alerts again.

NOTE: Before using the tuned upgrade files that you generated in the production test environment, you must edit the upgrade files to change the production test environment values to production environment values.

- 21 Verify you have a current backup of the production environment database.
- 22 On the Siebel Server you used to upgrade the production test environment, create an ODBC to connect to the production environment database.
- 23 Edit the tuned upgrade files you generated during the production test environment upgrade. Replace any values in the upgrade files that are specific to the production test environment with production environment values. You may need to change the following values in the upgrade files:
 - Host/LPAR name where the target database resides
 - DB2 subsystem name of the target database
 - Schema/Tableowner qualifier name on the target database
 - ODBC data source name of the target database

You can edit the upgrade files using any utility that allows you to edit partitioned data sets (PDSs). For advice on editing upgrade files, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

- 24 Upgrade the target database using the tuned upgrade files, which now contain production environment information, by performing the following tasks:
 - a [“Creating and Loading Siebel Log Tables” on page 168](#)
 - b [“Applying Additive Upgrade Changes to the Target Database” on page 168](#)
 - c [“Renaming the Production Environment Repository” on page 118](#)

d [“Performing the In-Place Target Database Upgrade” on page 170](#)

25 Extract the storage control file from the target database using the Database Configuration Wizard as follows:

- Specify the following values when prompted to do so:
 - **Extract Options:** Extract from Catalog
 - **Storage Control File:** Specify the name of the storage control file to be extracted, for example, storage_postupg.ctl.
- Make sure you specify values for the target database when prompted for the names of the schema Qualifier, ODBC data source, and database user name and password.

The procedure to extract a storage control file using the Extract from catalog option is described in *Implementing Siebel Business Applications on DB2 for z/OS*. The extracted target database storage control file is used as input to the upgphys upgrade process.

26 Upgrade the repository and import seed data. See [“Upgrading the Repository and Importing Seed Data” on page 135](#).

27 [“Performing Intersection Table Maintenance” on page 234](#).

28 [“About the Siebel Upgrade Log Files” on page 239](#).

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

- a Restore the backup of the database.
- b Correct the errors.
- c Rerun the Database Configuration Wizard.

30 **6.2.1 upgrades only:** If you have multilingual deployments, perform the steps in 477094.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 447. This document describes how to import language-specific repository strings and seed data into the upgrade repositories.

31 [“Manually Archiving Upgrade Log Files” on page 242](#).

32 Back up the upgraded production database.

33 [“Deleting Redundant Upgrade Files” on page 237](#).

Postupgrade Tasks for Database and File System

34 Perform the relevant tasks in the following:

- [Chapter 19, “Postupgrade Tasks for Siebel Database and File System”](#)
- The chapter of the *Siebel Database Upgrade Guide* that describes the postupgrade tasks for the Siebel database and file system.

35 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 and *Implementing Siebel Business Applications on DB2 for z/OS* for recommended parameter settings.

- 36 If you exported data from interface tables before the upgrade, review the database and import the data as desired.
- 37 Upgrading to Siebel RC2 or AES Encryption. See the chapter of the *Siebel Database Upgrade Guide* that describes Siebel database and UI upgrade planning.
- 38 Generate database statistics for new or rebuilt tables and indexes. Determine whether any tables need to be reorganized by running RUNSTATS with the Report Only option specified. For more information about running statistics, see [“Generating RUNSTATS Jobs” on page 182](#).

The production environment is now upgraded. The remaining topics in this chapter deal with configuration and validation tasks.

Postupgrade Tasks for Applications Configuration

- 39 Review the results of the Person and Organization merge, if applicable. Make configuration changes as required. For further information, see the topic on the Party Model in the chapter of the *Siebel Database Upgrade Guide* that describes Siebel database and UI upgrade planning.
- 40 Perform the tasks in the following:
 - [Chapter 20, “Postupgrade Tasks for Siebel Applications”](#)
 - The chapter of the *Siebel Database Upgrade Guide* that describes the postupgrade tasks for Siebel applications.
- 41 Activate and deploy workflows. To perform these tasks, see *Siebel Business Process Framework: Workflow Guide*.
- 42 If you have set up integrations for transferring data to or from third-party applications using Siebel EAI, verify the integrations are configured correctly. For information on using EAI, see *Overview: Siebel Enterprise Application Integration*.
- 43 If you have used EIM to set up batch processing jobs, verify EIM is configured correctly. For information on using EIM, see *Siebel Enterprise Integration Manager Administration Guide*.
- 44 If you customized style sheet or web template files in the previous release, you have to implement those customizations again in the new release, if desired.

Perform System Tests

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

Deploy to Users

- 46** Upgrade your Siebel Mobile and Developer Web Clients. See the *Siebel Installation Guide* for the operating system you are using for further information.
- 47** If you have customized the configuration of Siebel Enterprise Server components, such as Siebel Servers, you must manually enter the customizations in the upgraded environments. See *Going Live with Siebel Business Applications*.
- 48** Use Siebel Packager to create language packs for remote installation. See *Going Live with Siebel Business Applications*.
- 49** Use Siebel Anywhere to create installation kits for deployment. See *Siebel Anywhere Administration Guide*.
- 50** Generate a Siebel Remote database template. See *Siebel Remote and Replication Manager Administration Guide*.
- 51** Set up database extraction for Siebel Mobile Web Clients. See *Siebel Remote and Replication Manager Administration Guide*.

5

Planning A Siebel Database Upgrade

This chapter describes some of the important database-related issues you need to consider when planning an upgrade to Siebel 8.0. Also review the applicable topics in the chapter of the *Siebel Database Upgrade Guide* that describes the Siebel database and UI upgrade planning tasks before starting your upgrade. This chapter includes the following topics:

- [Planning Resources for Upgrading to 8.0 on z/OS on page 73](#)
- [Planning Changes to the Physical Layout of the Schema on page 74](#)
- [Testing Before a Production Upgrade on page 76](#)
- [Considering Code Page Support on page 76](#)
- [About Executing Jobs Using Siebel-Scheduled Mode or Vendor-Scheduled Mode on page 77](#)
- [Staging and Target Database Planning on page 78](#)
- [Obtaining Required Software and Hardware on page 78](#)
- [Obtaining Required IBM Utilities on page 78](#)
- [About Using the DSNTIAUL Utility on page 79](#)
- [Obtaining Required Security Privileges on page 82](#)
- [Planning Backup and Recovery Stages on page 83](#)
- [About Creating a Schedule for the Upgrade on page 83](#)
- [About Estimating Database Size on page 84](#)
- [Upgrading Your DB2 Software on page 84](#)
- [About Multilingual Deployments on page 85](#)
- [About Database Clustered Indexes on page 85](#)

Planning Resources for Upgrading to 8.0 on z/OS

The chapter of *Siebel Database Upgrade Guide* that describes Siebel database and UI upgrade planning tasks also describes the planning resources available to you before you start to upgrade to Siebel 8.0. This topic describes additional resources available if you use Siebel Business Applications with DB2 UDB for z/OS.

Documentation

If you are upgrading from Siebel 6.2.1 to Siebel 8.0, the following documents, available on the Siebel Bookshelf, may be helpful:

- *Planning an Upgrade to Siebel 7* for information about upgrading to the Siebel 7 data model for the first time
- *Siebel Connector for SAP/R3* to obtain Siebel 6.2.1 upgrade instructions specific to this connector
- If you are upgrading from Siebel 6.2.1, see 477067.1 (Article ID) on My Oracle Support for important information on data migration. This document was formerly published as Siebel Technical Note 428.

Read *Implementing Siebel Business Applications on DB2 for z/OS* for information on configuring your Siebel application after you have upgraded to Siebel 8.0.

About Oracle's Application Expert Services

Oracle's Application Expert Services offers detailed implementation planning and technical consulting services. Oracle recommends that you engage Oracle's Application Expert Services for help in planning your upgrade on DB2 for z/OS. Oracle's Application Expert Services can help you to:

- Take advantage of the new features provided by Release 8.0
- Customize the upgrade scripts and the upgrade process as appropriate for your installation
- Carry out performance tuning on the upgrade scripts to minimize production downtime

For further information, see ["About Using Oracle's Application Expert Services"](#) on page 20.

Planning Changes to the Physical Layout of the Schema

In planning your upgrade, you must understand the existing physical layout of your schema and determine whether or not you need to change this for the upgrade to Siebel 8.0. Also consider database space requirements and whether or not you need to move table spaces. These issues are discussed in this topic.

New Database Schema Structure Since Release 7.7

In Release 7.7, Siebel introduced the current database schema structure, referred to as the 1:1:1 model. It has the following characteristics:

- One table in each table space
- One table space in each database

Prior to Release 7.7, the database schema was built using approximately 20 databases, each of which contained multiple table spaces. Each of these table spaces (if non-partitioned) contained multiple tables. Release 8.0 contains thousands of databases. For example, an SIA installation has approximately 2700 databases. Each database has one table space and each table space has one table.

This model meets IBM recommendations and prevents database descriptor (DBD) locking and logging. These issues arise due to the increasing intensity of DB2 DML and DDL operations and the interaction of these operations with the DBD. The DBD is locked when information about the DB2 objects contained by the DBD is requested and accessed. In general, the more objects a DBD contains, the more probable that a DBD lock will be requested when information about a child object of the DBD is accessed.

Locks are acquired on the DBD table space (DBD01) if a DBD is not in memory (EDM pool). If the DBD is in the EDM pool, no lock is acquired on it if the SQL being run is static. However, most SQL executed by Siebel applications is dynamic; this means locks are acquired on the DBD. For more information on DBD locking, refer to the relevant IBM documentation.

The adoption of the 1:1:1 model since Release 7.7.x means that you must decide how much of this model to deploy. You have the following options:

- Create all tables in the 1:1:1 model.

New 8.0 tables are created in the 1:1:1 model and the Siebel supplied storage control file is used for the upgrade.

- Create new tables in the 1:1:1 model and maintain existing tables in their current table space if possible.

The following scenarios arise if you select this latter option:

- Some existing tables need to be moved to incorporate the addition of new columns.
- Some existing tables need to be moved, because they have been extended and the addition of new columns causes the table's LRECL to exceed that of the table space. This necessitates the use of the extract and merge methodology to create the storage control file. For more information on this methodology, see ["Extracting the Storage Control File" on page 91](#).
- Tables that are to retain the existing format are *merged* into the template control file which employs the 1:1:1 model.

For both options, enter existing tables that are to be migrated to the 1:1:1 model in the file `override.inp`. See ["About the Override File" on page 44](#) for further information.

For more information on using storage control files, see *Implementing Siebel Business Applications on DB2 for z/OS*.

Preparing a Storage Control File

A key task for a successful upgrade is the building of a suitable storage control file for both the development and production upgrade. You must consider space requirements. This is particularly important for the development upgrade, because three new repositories are imported into the database (one extra repository is imported during the production upgrade). Some repository tables will increase significantly in size, so you must provide sufficient space for expected database growth. For information on preparing a storage control file, see ["Preparing the Storage Layout of the Schema" on page 88](#).

About Moving Table Spaces

If you want to move tables from one table space to another, you need to recreate the tables in the new table space and then drop the existing table space. You cannot change the bufferpool designation in the storage control file to move tables because the page size is associated with the table space.

For example, if you are making changes to an existing table space that is using BP1 or a 4 KB bufferpool and these changes cause you to receive a warning from ddlimp that the table will now need to be in a 16 KB bufferpool, do not just change the bufferpool designation in the storage control file from BP1 to BP16K1. Doing so can cause any LONG VARCHAR column in the table to be bigger than it needs to be, resulting in performance problems.

Testing Before a Production Upgrade

Careful testing is critical for a successful upgrade. In particular, the production upgrade must be thoroughly tested to avoid data-specific issues and gain the best possible performance during your upgrade.

CAUTION: Do not go live on Siebel 8.0 without exhaustive performance testing.

Considering Code Page Support

Siebel 8.0 supports ASCII-, EBCDIC-, and Unicode-based coded character set IDs (CCSIDs) on DB2 for z/OS. Development databases, however, can not use EBCDIC code pages, because databases with EBCDIC code pages do not support two critical procedures in a development environment upgrade:

- Merging prior configuration changes into a new custom configuration repository
- Compiling a new Siebel repository file (.srf file) from the new repository

Additional limitations on databases with EBCDIC code pages include the following:

- Siebel Web Client migration is not supported
- Siebel Dun & Bradstreet server components are not supported

Before you conduct an upgrade, carefully read *Siebel Release Notes* on My Oracle Support for information about known restrictions. For guidelines about choosing the code page for your subsystem, see *Implementing Siebel Business Applications on DB2 for z/OS*. In addition, make sure you follow the rules specified by IBM for character conversion as described in the IBM DB2 Version 8 installation documentation.

About Code Page Conversion

Siebel supports the ASCII 5348 and EBCDIC 1140 code pages on DB2 for z/OS (see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network for further information). If your current Siebel release runs on a different code page, you must migrate to one of the supported code pages before upgrading to Siebel 8.0.

Since Release 7.5, Siebel Business Applications has supported the Unicode character set. On the z/OS platform, however, Siebel Business Applications only supports the Unicode character set on DB2 for z/OS v8 running in New Function Mode. When necessary, therefore, the z/OS Unicode Conversion Services are used for character conversion. The conversion mechanisms used since Siebel 7.5.x (enforced subset conversions) are stricter than those that were used in earlier releases, therefore, characters unsupported on the target schema (for example, the euro symbol) may no longer display correctly. For this reason, it is important to check the Siebel schema code page before starting the Siebel upgrade, especially if you are upgrading from releases earlier than Siebel 7.5.x.

Schedule sufficient time to perform and test the code page conversion before beginning the upgrade. For advice and assistance on converting DB2 code pages, contact IBM.

About Executing Jobs Using Siebel-Scheduled Mode or Vendor-Scheduled Mode

Before you perform a production upgrade, determine whether you will execute the jobs using Siebel-provided job scheduling or a third-party vendor scheduler. Choose your scheduling mode carefully, because once you begin an upgrade process under a selected mode, you cannot change your scheduling mode or reverse this decision.

In Siebel Scheduled Mode, the Siebel job scheduler uses job submission EXECs to run the following upgrade jobs:

- Pret
- Pret_FINS
- Household
- Household_FINS
- Preschm
- Preschm-FINS
- Prodcfg
- Prodcfg_FINS
- Upglss

The Siebel job scheduler automatically submits dependent jobs by their predecessors.

Using Siebel Scheduled Mode, if a job ends abnormally or returns an invalid return code, the upgrade process is halted. You can check the job status by querying the Siebel job log. For further information, see [Chapter 17, "Reviewing the Siebel Upgrade Log Files."](#)

If you use a third-party job scheduler, jobs are not submitted automatically. In this case, you will find it useful to first generate the upgrade scripts using Siebel scheduling to gain an understanding of job dependencies.

NOTE: Logging is provided for both Siebel-scheduled and vendor-scheduled jobs using a DB2 table. Each job contains a step that checks whether or not all the other steps in the job completed successfully. An SQL UPDATE statement is then executed against the log table specifying the job status.

Staging and Target Database Planning

When scheduling your upgrade, be aware that the target database schema must not be changed after the staging DDL is produced. If target database schema changes are applied before the upgrade is completed, you must recreate the staging database and generate the upgrade files again.

When planning your upgrade, keep in mind the following:

- The staging database must be in a separate DB2 subsystem to the target database
- The staging database table space names must be same as those in the target database
- The staging database tableowner and storage group names may be same or different to the target database names

DB2 DSNZPARM Settings For the Target Database

For Siebel Business Applications to run correctly and efficiently, the DSNZPARM parameters for your target database must be set correctly. For a list of the Siebel required and recommended DSNZPARM parameter settings for DB2 for z/OS v8, see *Implementing Siebel Business Applications on DB2 for z/OS*.

Obtaining Required Software and Hardware

Because each enterprise has specific needs for hardware and software resources, it is recommended that you discuss your particular situation with your Siebel technical resource. Make sure your hardware supports the needs of your Siebel 8.0 upgrade.

Read *Siebel System Requirements and Supported Platforms* on Oracle Technology Network to verify the supported computer and operating system platforms and supported third-party programs for this release of your Siebel Business Applications.

Obtaining Required IBM Utilities

Verify that the IBM utilities listed below are available for your upgrade to Siebel 8.0:

- **DSNTEP2.** The upgrade uses DSNTEP2 to execute SQL.

Prepare and bind DSNTEP2 by following the procedures in your IBM installation documentation. If you made local modifications, you might need to prepare and bind a separate version. Also, if you are using a separate version of DSNTEP2, you need to change the SIEBSQL* members to reflect the new plan and program names.

- **DFSORT.** The utility DFSORT is used to manipulate data for data migration during upgrade.
- **LOAD.** The IBM DB2 Load utility is used to load data during data migration.
- **REBUILD INDEX.** The IBM DB2 REBUILD INDEX utility (DSNUTILB) is used to build indexes after they are created using DEFER YES.
- **IEBCOPY.** The utility IEBCOPY is used to create members in installation datasets. Sequential datasets contains control information used by IEBCOPY.
- **IEBGENER.** The utility IEBGENER is used to copy sequential datasets.
- **DSNTIAUL.** The upgrade uses DSNTIAUL to unload Siebel CRM application data.

Compile, link-edit, and bind DSNTIAUL by following the procedures in your IBM DB2 installation documentation.

For information on these IBM utilities, see your IBM documentation.

CAUTION: You can choose to use alternate third-party utility products that are preferred for your environment. Evaluate utilities by individual job. Be aware, however, that if you do use utilities other than those recommended below, you might have to modify the Siebel-supplied SQL to accord with the rules for those utilities. For help with modifying Siebel-supplied SQL, you *must* contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

About Using the DSNTIAUL Utility

In Siebel 8.0, the DSNTIAUL utility is used to unload data from tables. However, the SELECT statements that Siebel 8.0 uses to specify the selected columns to be unloaded cause DSNTIAUL to issue a warning message and return a non-zero return code for every unload job. This makes the process of determining whether an unload job completed successfully or not more difficult.

To avoid this problem, Siebel 8.0 contains a patch for DSNTIAUL which causes a successful unload job to generate a zero return code by changing the selective SELECT warning message to an informational message. Jobs that generate a zero return code require no further consideration. After you apply the patch to DSNTIAUL, all jobs resulting in non-zero return codes must be reviewed. For further information, see ["About Applying the DSNTIAUL Patch" on page 80](#).

About Applying the DSNTIAUL Patch

The DSNTIAUL patch, @@TIAUL USERMOD, is packaged as an SMP/E format USERMOD. The patch is a MACUPD (macro update) because DSNTIAUL is delivered as a macro and not as a source object. @@TIAUL updates the DSNTIAUL assembler source code so that non-zero return codes generated solely by the selective SELECT statements are suppressed. The USERMOD is generic in that it applies to known fix levels of the DSNTIAUL utility.

NOTE: It is recommended that a DB2 systems programmer who has knowledge of SMP/E and your maintenance process apply the patch.

To apply the patch, copy the @@TIAUL USERMOD code, listed in “The @@TIAUL USERMOD” on page 80, and apply it to the SMP/E DB2 target zone on z/OS. You can apply the patch using various methods, for example:

- Use the SMP/E APPLY command to apply @@TIAUL USERMOD directly to the system, then run assemble, link, and bind jobs to specify it as the default program
- Copy the source member containing the original DSNTIAUL code and add the code from @@TIAUL USERMOD to the copy. Then create an executable with a new name for the new source member.

NOTE: If you use this method, you might need to change the Siebel upgrade scripts to accommodate any changes to the way in which DSNTIAUL is invoked.

Ask your z/OS system administrator for the most appropriate method to use to apply the @@TIAUL USERMOD at your site.

The @@TIAUL USERMOD

This section lists the @@TIAUL USERMOD. Before copying the following code and applying the USERMOD, change the modification control statements to reflect the applicable maintenance level of the existing DSNTIAUL utility; comments in the @@TIAUL USERMOD provide specific instructions.

```
++ USERMOD (@@TIAUL) REWORK(2006297)
/*
```

```
    USERMOD @@TIAUL:
        ALLOW RETCODE 0 EVEN WHEN SELECTING LIMITED COLUMNS.
```

```
    THIS USERMOD WILL CAUSE TWO SECTIONS OF CODE TO BE BYPASSED.
    THAT CODE SETS RETURN CODE 4 IN THE CASE OF A SELECT OF
    COLUMNS LESS THAN THE TOTAL NUMBER OF COLUMNS OF THE TABLE.
```

```
    NOTE:
    THE PRE-REQUISITE (PRE) OPERAND <<MUST>> BE CHANGED ON THE
    ++VER CONTROL STATEMENT TO REFLECT CURRENT MAINTENANCE LEVEL.
```

```

MACRO DETAILS:
  MACRO      FMI D      RMID      SYSLIB  DI STL I B
  DSNTIAUL  HDB8810  UK17727  SDSNSAMP ADSNMACS
*/ .

++ VER (P115)
FMI D(HDB8810)
PRE  (UK17727)
/*
*****
* DO NOT ADD LINE NUMBERS TO THIS USERMOD! *
*****
* SKIP LINES WITH SEQ NUMBERS 11900000 TO 11930000 *
* SKIP LINES WITH SEQ NUMBERS 25660000 TO 25700000 *
*****
*/ .

++MACUPD(DSNTIAUL) DI STL I B(ADSNMACS) .
./ CHANGE NAME=DSNTIAUL
*** DO NOT SET WARNING RETURN CODE *** @ORACLE 11895000
      AGO .NOSLCT_BYPASS @ORACLE 11895001
      .NOSLCT_BYPASS ANOP , @ORACLE 11935000
*** DO NOT SET WARNING RETURN CODE *** @ORACLE 25655000
      AGO .SKIP_RETWRN2 @ORACLE 25655001
      .SKIP_RETWRN2 ANOP , @ORACLE 25705000
./ ENDUP

```

About DSNTIAUL CCSID Conversion Errors

DSNTIAUL unloads all Siebel CRM application data in an EBCDIC CCSID. If the data to be unloaded is in ASCII format, conversion errors can occur for characters that are supported in ASCII but that are not supported in EBCDIC, for example, the euro and copyright symbols. CCSID conversion errors generate a RC=4 and produce an SQLSTATE=01517.

If the ASCII data contains a character that cannot be converted to EBCDIC, DSNTIAUL stops the unload process at that point. To correct this problem, you can either:

- Update the source data

- Specify the DSNTIAUL TOLWARN (YES) parameter

CAUTION: It is recommended that you do NOT use the TOLWARN (YES) parameter because it suppresses conversion error messages and can result in data corruption.

The TOLWARN (YES) parameter forces DSNTIAUL to continue the unload process by inserting substitution characters for characters that cannot be converted; this ensures all records are unloaded even if they contain data that cannot be directly converted from ASCII to EBCDIC.

However, using the TOLWARN (YES) parameter can also cause corrupted data in your upgraded Siebel application. DSNTIAUL provides the same substitution character for all ASCII characters that cannot be converted to EBCDIC. When the data is converted back to ASCII, the substituted characters are all converted back to the same ASCII character, for example, the euro and copyright symbols will both be represented by the same character. After the upgrade, your Siebel application will contain corrupted data and you will have to review the source data to determine which symbol a corrupted character corresponds to.

Obtaining Required Security Privileges

For detailed information about security for DB2 UDB for z/OS installations and upgrades, see *Implementing Siebel Business Applications on DB2 for z/OS*.

In Siebel 8.x, access privileges to database resources such as tables, views, and triggers are granted to a user group. A user group is a definition within the security package (for example, RACF) that has a common set of users attached to it. Access to the DB2 tables is granted to the user group, and user authentication is performed at the group level. All users belonging to the group are allowed access. All users that are not part of the group are denied access.

The user who executes the upgrade must be a member of a qualified group. To grant this user tableowner privileges, the tableowner must be set up as a qualified group, and the DBA who executes DDLs must be a member of this qualified group. The group ID is the qualifier (for example, RACF group ID).

The Siebel installation process allows the installer to specify the group user name for client access (the default is SSEROLE), and the resulting installation scripts generate the appropriate GRANT statements. GRANT statements for additional security groups that may be required must be created manually.

NOTE: The GRANT statements must be executed by either the tableowner, a database administrator, or a system administrator.

The following privileges are necessary for the user who performs the upgrade:

- Read the DB2 catalog
- Execute stored procedures
- Bind stored procedures

Because each enterprise has specific needs, it is recommended that you discuss your particular situation with your Siebel technical resource.

Planning Backup and Recovery Stages

You may need to recover all Siebel DB2 objects to a prior point in time. Your usual point-in-time recovery techniques may be insufficient in this environment.

In addition to the backup and recovery procedures that are standard for your environment, take a set of DB2 backups at key stages during the upgrade, using your preferred utility. A snapshot of your repositories and environment at these stages protects the progress of your upgrade in the event of a failed subsequent process.

It is recommended that you back up your Siebel schema at these key stages of the upgrade:

- Before any upgrade activity is started
- Before performing unloads

NOTE: Unloads must be performed when there is no system activity, so that the database is at a point of consistency.

- After upgrading the Siebel database schema
- After the repository merge
- After upgrading the custom database schema

Review the results of all JCL jobs that you execute during the install or upgrade process. You can use a spool viewer such as IBM's SDSF to inspect the output from these jobs. You can review this information in addition to reviewing the upgrade log files.

About Creating a Schedule for the Upgrade

Develop a plan for your upgrade based on the objectives and constraints for your deployment.

If you are migrating multiple languages from a prior version, plan extra time for the repository merge process. The expected merge time may increase with the number of languages in the repository. You also might need to plan for additional installation-related tasks.

The following procedures may reduce the time required for your upgrade:

- Run selected processes in advance of the upgrade.

Certain preupgrade tasks can be run at any time prior to the upgrade. These procedures can be performed in advance either for testing purposes or to accommodate down-time constraints.

Examples of procedures that can be performed by a database administrator in advance of your upgrade include [“Preparing the Storage Layout of the Schema” on page 88](#).

- Apply additive schema changes to the production database ahead of the target database in-place upgrade. See [“About Siebel Additive Schema Upgrade Changes” on page 24](#).
- Prepare select processes to run in parallel.

If a large table such as S_EVT_ACT is partitioned, it can run in parallel by transferring shipped statements into the numbered SQL statement.

Do not start a new development effort until after the new version has been rolled out. This is especially important when you are upgrading from Release 6.2.1 to Release 8.0. There are significant user interface differences between the two versions, and you must change existing functionality before moving on to customized functionality.

About Estimating Database Size

Database upgrade is resource intensive. If the upgrade exceeds available resources, the upgrade halts. You must then resolve resource issues before resuming the upgrade.

To help you estimate the database size required when upgrading to release 8.0, [Table 7](#) shows the number of tables in 4-KB, 16-KB, and 32-KB table spaces in a sample SIA database in a 7.0.x, 7.5.x and a 7.8 release. [Table 7](#) also shows the space required by the tables.

Because Siebel has adopted a 1:1:1 database schema structure since release 7.7 (one table in each table space, one table space in each database), these releases require many more 16-KB and 32-KB table spaces than pre 7.7 releases. However, some tables may not require 16-KB and 32-KB table spaces if you convert LONG VARCHAR columns to CLOB columns.

Actual expected growth may also vary widely from these estimates, depending on which Siebel application you are using (Siebel Business Applications or Siebel Industry Applications), database configuration, row size of tables, data content, and code page. The number of tables and space estimates shown in [Table 7](#) for Release 8.0 are for an EBCDIC database.

Table 7. Number of Tables and Space Required in a Sample Siebel DB2 Database by Release

Release...	4-KB table space	8-KB table space	16-KB table space	32-KB table space
8.0	4229 (2,759,008 KB)	331 (312,112 KB)	171 (62,272 KB)	105 (52,160 KB)
7.8	3857 (569,442 KB)	N/A	459 (237,735 KB)	86 (224,62 KB)
7.5.x	2542 (637,706 KB)	N/A	404 (336,384 KB)	22 (143,28 KB)
7.0.x	1995 (398,068 KB)	N/A	357 (135,469 KB)	10 (102,39 KB)

NOTE: [Table 7](#) shows the space required by the tables in a release but does not include the space required by indexes.

Upgrading Your DB2 Software

Upgrades: All upgrades.

Environments: Development environment only.

Before you upgrade, carefully review *Siebel System Requirements and Supported Platforms* on Oracle Technology Network to verify that you are using the currently supported versions of the following DB2 software:

- **DB2 Version 8.** If you are using an unsupported version of DB2, you must migrate to and thoroughly test the currently supported version before you upgrade to Siebel 8.0. Perform your DB2 version migration before upgrading your Siebel database. Allow sufficient time to verify the migration before proceeding with the upgrade.
- **DB2 Connect.** Siebel Developer Web Clients (Siebel Mobile Web Client in connected mode) and Siebel Servers communicate with DB2 UDB for z/OS through DB2 Connect middleware. Verify that you are using the version of DB2 Connect supported for Siebel 8.0.

NOTE: Make sure that all software meets the requirements specified in *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

About Multilingual Deployments

Upgrades: Release 6.2.1 only.

Environments: All environments.

As of Siebel 8.0, the upgrade process upgrades both the primary (base) language and all deployed languages for upgrades from Siebel 7.x. If you are upgrading to the current release from Release 6.2.1, however, the upgrade process only upgrades the primary (base) language. All other deployed languages must be upgraded manually.

If you have multilingual deployments, therefore, you must import multilingual repository strings and seed data after performing the upgrep upgrade process. For instructions on how to perform these imports, see 477094.1 (Article ID) on My Oracle Support. This document was formerly published as Siebel Technical Note 447.

For instructions on installing Siebel language packs, see the *Siebel Installation Guide* for the operating system you are using.

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

About Database Clustered Indexes

Upgrades: Release 6.2.1 only.

Environments: All environments.

If you created clustered indexes on base tables, and the upgrade introduces a different clustered index on the same table, the upgrade process re-creates custom indexes as nonclustered and creates the Siebel index as clustered.

For IBM DB2, indexes that reside on tables with append mode enabled are re-created as nonclustered indexes during the upgrade. (Tables created with append mode enabled do not support clustered indexes.)

6

Basic Database Preparations for a Siebel Upgrade

This chapter describes the DB2 for z/OS database tasks you must complete before upgrading to release 8.0. Also review the chapter of the *Siebel Database Upgrade Guide* that describes the basic database preparations for a Siebel upgrade and perform any applicable tasks before starting your upgrade. This chapter includes the following topics:

- [Verifying Database Configuration on page 87](#)
- [Creating Storage Groups on page 88](#)
- [Preparing the Storage Layout of the Schema on page 88](#)
- [Reviewing EIM Table Partitioning and Data Distribution on page 99](#)
- [Backing Up the Database on page 100](#)
- [Granting a Siebel User Upgrade Authorization on page 100](#)

Verifying Database Configuration

Upgrades: All upgrades.

Environments: Development environment only.

Verify that your development Siebel database configuration meets or exceeds Siebel requirements as described in *Implementing Siebel Business Applications on DB2 for z/OS*. Be aware of the following:

- Informational APAR pq83289 is required for loading multiple tables to the same database in parallel.
- The upgrade sets the TRACKMOD parameter to YES, which is the IBM default value for table space objects. Oracle recommends that you set the TRACKMOD parameter to NO to reduce data sharing overhead.
- Set the DSNZPARM CMTSTAT to INACTIVE to prevent timeout errors from occurring. If you do not run INACTIVE, then set IDTHTOIN to 0 (Inactive).
- Ensure the z/OS Unicode Conversion Services are correctly installed and configured, as provided by the IBM APAR OA04069. For further information on z/OS Unicode Conversion Services, see *Implementing Siebel Business Applications on DB2 for z/OS*.

CAUTION: It is very important that the required maintenance level listed in informational APAR ii13012 is fully applied, and that all DSNZPARMs required for Siebel implementation on DB2 v8 are set up correctly.

Before you begin your upgrade, verify your database configuration. The consequence of exceeding available resources is a halted upgrade that requires you to allocate time to adjust the environment and then resume the upgrade.

When you upgrade from a previous version of Siebel Business Applications to Release 8.0, the size of your database is likely to increase. The amount by which your database grows may vary widely, depending on your database configuration, row size of tables, and data content. For information on estimating the growth, see [“About Estimating Database Size” on page 84](#).

The growth percentage will also increase depending on how you size your database and configure default storage for database table spaces. For example, if you set the default storage for your initial or next extent in a given DB2 table space to 10 KB, that table space will grow by a smaller percentage than if you set it to 100 KB.

Creating Storage Groups

Upgrades: All upgrades.

Environments: Development environment only.

Before upgrading the Siebel database, your DBA must create storage groups on the staging database. The names used to define the storage groups for the staging database can be the same or different to those defined on the target database.

Preparing the Storage Layout of the Schema

Upgrades: All upgrades.

Environments: All environments.

Before starting the 8.0 upgrade, you have to prepare the storage control file you will use during the upgrade. The DBA can prepare the storage control file at any time before the upgrade.

The storage control file contains storage information, including bufferpools and storage groups, that is used as the basis for the storage layout of your new 8.0 Siebel database. Even if you are using a preconfigured storage layout, you must make sure that the layout is valid for your schema.

Preparing the storage control file process consists of four procedures:

- [“Modifying the Storage Control File” on page 89](#)
- [“Extracting the Storage Control File” on page 91](#)
- [“Validating the Extracted Storage Control File” on page 95](#)
- [“Reviewing the Storage Control File” on page 97](#)

NOTE: You must validate the storage control file after you extract it and after you modify it.

There are different starting points from which you can customize your storage layout:

- **Scenario 1.** Begin with a Siebel-provided storage layout template, import the template into the Siebel Database Storage Configurator (dbconf.xls), customize it, then export it as your customized layout.

- **Scenario 2.** Use your current configuration from an existing database layout and merge it with one of the Siebel-provided templates. This can then be imported to the Siebel Database Storage Configurator for further manipulation.

NOTE: For more information on the Siebel Database Storage Configurator, see *Implementing Siebel Business Applications on DB2 for z/OS*.

To prepare the storage control file, use the Database Configuration Wizard to extract the storage layout of your database from the DB2 catalog. As part of the extraction process, you can merge the storage layout information from your existing database with information you already input into a storage template file or information in a template provided by Siebel. As an alternative, instead of extracting the storage layout of your existing database, you can use a Siebel-provided template as the storage control file.

Siebel provides the templates listed in [Table 8 on page 89](#) for the storage control file:

Table 8. Storage Control File Templates

Template	Description
storage_np.ctl	Database storage layout for a non-Unicode Siebel schema with no partitioning scheme
storage_np_u.ctl	Database storage layout for a Unicode Siebel schema with no partitioning scheme
storage_p.ctl	Database storage layout for the Siebel schema with partitioning for a set of tables on an ASCII database
storage_p_e.ctl	Database storage layout for the Siebel schema with partitioning for a set of tables on an EBCDIC database
storage_p_u.ctl	Database storage layout for the Siebel schema with partitioning for a set of tables on a Unicode database

For more information about templates for the storage control file, see *Implementing Siebel Business Applications on DB2 for z/OS*.

Modifying the Storage Control File

There are three methods by which you can configure storage space:

- **Method 1.** This method consists of performing a standard Siebel database installation by running the Database Configuration Wizard, choosing the Generate DDL into a File installation option, and specifying as input one of the Siebel storage control file templates. This process generates the following:
 - A storage control file, based on the Siebel template file you selected, that incorporates the configuration information you entered when you ran the Database Configuration Wizard. This file is generated in the dbsrvr\DB2390 (Windows) or dbsrvr/DB2390 (UNIX) directory

- A schema.sql file that is applied on the z/OS host to create the Siebel schema. The schema.sql file is based on the customized storage control file generated by the database install

Using these files, you can then configure storage space in any of the following ways:

- Amend the storage control file generated during the database install in the dbsrvr\DB2390 (Windows) or dbsrvr/DB2390 (UNIX) directory (see also Method 2).
- Apply the schema.sql file generated by the database install on the DB2 host to create the Siebel schema, then amend the schema using native DB2 tools. Extract the storage control file from the DB2 catalog; the file will include the changes you have made.

NOTE: You access the extract utility Siebel provides through the Database Configuration Wizard. This utility allows you to extract information from the DB2 catalog. You can use this extract utility any time you want to create a new storage control file, based on the DB2 catalog.

- Amend the schema.sql file generated by the database install directly, apply it on the DB2 host to create the schema, and then extract the storage control file, which will include the changes you made.

For information on installing the Siebel database and extracting storage control files, see *Implementing Siebel Business Applications on DB2 for z/OS*.

- **Method 2.** This method consists of manipulating the storage control file (storage.ctl) directly by opening it with a text editor program. This method can be used if you understand the file structure.

You can use this method to amend one of the Siebel-supplied storage control files or to amend a storage control file that you have extracted from another Siebel schema.

- **Method 3.** This method consists of using the Siebel Database Storage Configurator tool. The Siebel Database Storage Configurator tool is a Microsoft Excel macro (dbconf.xls) that is installed in the dbsrvr\db2390 (Windows) or dbsrvr/db2390 (UNIX) subdirectory of your installation directory. This tool allows you to import a storage control file, amend it, validate the syntax of your changes, and then save it.

UNIX customers must transfer dbconf.xls and the .ctl files to their Microsoft Windows environment. Use BINARY FTP transfer for the dbconf.xls file.

For information on using the Siebel Database Storage Configurator, see *Implementing Siebel Business Applications on DB2 for z/OS*.

NOTE: Validate the storage control file after you modify it. See [“Validating the Extracted Storage Control File” on page 95](#) for further information.

About Extracting the Storage Control File

The information in your storage control file comes from the target database (database to be upgraded). When you extract the storage control file, you can choose one of two methods:

- **Extract from Catalog.** This method extracts the storage layout of your existing database from the DB2 catalog as-is. The output is a new storage layout.

NOTE: This method can also be used to perfect your target database layout by performing a *dummy* installation, manipulating the schema layout through native utilities, then extracting the customized layout.

For information on the Extract from catalog option, see *Implementing Siebel Business Applications on DB2 for z/OS*.

- **Extract from Catalog and Merge with Template.** This method preserves your existing layout. This merges storage layout information from your existing database with information that you already entered into a storage template file provided by Siebel. This preserves your prior layout, and your output is the prior physical layout merged with an 8.0 storage layout template.

The first time that you run an upgrade, when you extract your storage control file, you must use the Extract from Catalog and Merge with Template option to preserve your existing layout. The list below details scenarios that may occur during the extract and merge process and the behavior you can expect in these cases:

- **A database object exists in the existing database but not in the template.** The definition of the database object is output to the new storage control file.
- **A database object is specified only in the template file.** The definition of the database object is output to the new storage control file.
- **A database object is specified in both the existing database and the template.** The layout of the existing database is extracted as the default. However, you can manually override this behavior by creating a file called `override.inp` in the BIN directory under `SIEBEL_ROOT`. Place any tables that you want to override into this file.

CAUTION: Partitioned tables were not supported for the 6.x releases. If you applied partitioning to release 6.x tables, carefully review the extracted data and modify it to conform with Siebel naming conventions as appropriate. If you have questions about amending the storage control file, create a service request (SR) on My Oracle Support.

Extracting the Storage Control File

After you have created the staging database but before you begin to generate the upgrade files, you must extract the storage control file from the staging database and merge it with a Siebel 8.0 template. You do this using the Database Configuration Wizard Extract from Catalog and Merge with Template option.

To extract the storage control file

- 1 Run the Database Configuration Wizard.

For information on running the Database Configuration Wizard, see [“About Running the Database Configuration Wizard Under Windows” on page 126](#) or [“About Running the Database Configuration Wizard Under UNIX” on page 129](#).

- 2 Enter the information in [Table 9 on page 92](#) when prompted by the Database Configuration Wizard. Collect this information and verify it before running the utility.
- 3 Save the information you have entered and launch the Upgrade Wizard as described in [Chapter 10, “Running the Database Configuration Wizard to Perform Upgrade Tasks.”](#)

The database catalog is read and your prior custom database layout is merged with one of the 8.0 Siebel database layout templates (located in the `DBSRVR_ROOT\db2390` directory). New objects take a layout from one of the layout templates.

NOTE: If you choose a Siebel storage control file template that includes partitioning, and the existing database schema does not include partitioning, by default, the existing database objects are not partitioned in the storage control file generated (that is, the database catalog overrides the templates) unless you specify the table names in the override input file (`override.inp`).

By default, new tables are created with one table in each database to prevent concurrency and locking errors.

Once the merged storage control file is created, you must verify it against the staging database. See [“Validating the Extracted Storage Control File” on page 95](#) for information on this task.

Information Required For the Database Configuration Wizard Extract Option

[Table 9](#) lists the information you must enter to run the Database Configuration Wizard Extract and Merge option.

Table 9. Information Required for the Database Configuration Wizard — Extract and Merge Option

Field Name or Menu	Required Information
Siebel Gateway Name Server Address Enterprise Server name	The Siebel Gateway Name Server machine name and the Enterprise Server name.
Siebel Server Directory	The absolute path of the directory where the Siebel Server is installed, for example, <code>C:\sba80\si ebsrvr</code> (Windows) or <code>si ebel /si ebsrvr</code> (UNIX). For UNIX, do <i>not</i> enter the string <code>\$SIEBEL_ROOT</code> .
Siebel Database Server Directory	The absolute path of the directory where the Siebel Database Configuration Utilities are installed—for example <code>C:\sba80\dbsrvr</code> (Windows) or <code>si ebel /dbsrvr</code> (UNIX).
RDBMS Platform	Choose IBM DB2 for z/OS.
Siebel Database Operation	Choose Run Database Utilities.
Database Utilities Options	Choose Configure Database
Database Configuration Options	Choose Extract Storage File to extract a storage control file.

Table 9. Information Required for the Database Configuration Wizard — Extract and Merge Option

Field Name or Menu	Required Information
Extract Options	<p>Choose the Extract from Catalog and Merge with Template option.</p> <p>This option preserves your existing layout. This option merges storage layout information from the database you specify with information that you already entered into a storage control file, only taking objects from the template that do not already exist in the catalog.</p> <p>NOTE: The first time that you run an upgrade, you must use the Extract from Catalog and Merge with Template option, thereby preserving your existing layout.</p>
ODBC Data Source Name	<p>Verify the ODBC name for connecting to the staging Siebel database.</p> <p>Windows: To find the name of your ODBC data source, navigate to Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC). Click the System DNS tab to find the name of your ODBC data source.</p> <p>UNIX: To find the name of your ODBC data source, type: vi \$ODBCINI.</p>
Database User Name	<p>Enter the user name and password for the Siebel administrator of the staging database.</p> <p>NOTE: The staging database user name (user ID) needs to have authorization to set CURRENT SQLID.</p>
Siebel Schema Qualifier	<p>Enter the eight-character identifier that designates the Siebel schema for your staging database. This is also an authorization ID. The schema qualifier must start with a letter, cannot contain special characters, and must be entered in uppercase.</p>
Database Encoding Scheme	<p>Specify whether your database is UNICODE or Non-UNICODE.</p> <p>When you are upgrading to Siebel 8.0, choose Non-UNICODE.</p>
Code Page Encoding Scheme	<p>Indicate whether your DB2 subsystem is ASCII or EBCDIC.</p>
Environment Type	<p>Indicate whether your database environment is production or development.</p>
Storage Control File Layout	<p>Choose Siebel Schema without Partitioning if you want all tables only in segmented table spaces.</p> <p>Choose Siebel Schema with Partitioning if you want a layout that includes a set of tables that is recommended for partitioning. The remaining nonpartitioned tables are in segmented table spaces.</p>
Default Tablespace	<p>Enter the name of the default table space.</p>

Table 9. Information Required for the Database Configuration Wizard — Extract and Merge Option

Field Name or Menu	Required Information
Storage Group for Tablespaces Storage Group for Indexes	<p>Indicate the values for the following parameters:</p> <p>Storage Group for Tablespaces. Enter the name of the table storage group.</p> <p>Storage Group for Indexes. Enter the name of the index storage group.</p>
4KB Buffer Pool Name 8KB Buffer Pool Name	<p>Indicate the values for the following parameters:</p> <p>4KB Buffer Pool Name. Enter the 4-KB buffer pool name for your table spaces or accept the default name, BP1. This buffer pool must already be activated and have access to it granted by the DBA.</p> <p>8 KB Buffer Pool Name. Enter the 8-KB buffer pool name for your table spaces or accept the default name, BP8K1. This buffer pool must already be activated and have access to it granted by the DBA.</p>
16KB Buffer Pool Name 32KB Buffer Pool Name	<p>Indicate the values for the following parameters:</p> <p>16KB Buffer Pool Name. Enter the 16-KB buffer pool name for your table spaces or accept the default name, BP16K1. This buffer pool must already be activated and have access to it granted by the DBA.</p> <p>32KB Buffer Pool Name. Enter the 32-KB buffer pool name for your table spaces or accept the default name, BP32K1. This buffer pool must already be activated and have access to it granted by the DBA.</p>
Index Buffer Pool Name Database Name Prefix	<p>Indicate the values for the following parameters:</p> <p>Index Buffer Pool Name. Enter the buffer pool name for indexes or accept the default name, BP2. This buffer pool must already be activated and have access to it granted by the DBA.</p> <p>Database Name Prefix. Enter the prefix for your target database name. The default prefix is SIDB.</p> <p>This prefix consists of up to four of the first characters in the names of your logical Siebel Databases. This prefix must start with a letter and cannot contain any special characters. All database names end in numbers.</p>

Table 9. Information Required for the Database Configuration Wizard — Extract and Merge Option

Field Name or Menu	Required Information
Storage Control File	<p>Enter the directory path and name for the storage control file created by this process. You must accept the default value displayed in the Storage Control File field; this varies depending on the type of upgrade you are performing:</p> <ul style="list-style-type: none"> ■ Development environment upgrade: storage_upg_dev.ctf ■ Production environment upgrade: storage_upg_prod.ctf.
Log Output Directory	<p>Accept the default or enter the directory name. If the directory does not exist, it will be created. Do not use special characters such as spaces, slashes, or symbols in the name of the log output.</p>

Validating the Extracted Storage Control File

When you have extracted your existing database storage control file and merged it with a Siebel 8.0 template in preparation for the 8.0 upgrade, you must validate the storage control file. (You must also validate the storage control file any time you modify it.) The validation process checks that the tables are the right length for the target schema. Do not proceed with the Siebel 8.0 upgrade until the validation process runs without error.

The following procedure describes how to validate the storage control file you extracted and merged with a Siebel 8.0 template as described in [“Extracting the Storage Control File” on page 91](#).

To validate the storage control file

- 1 Run the Database Configuration Wizard as described in [“Extracting the Storage Control File” on page 91](#).

NOTE: The procedure to validate a storage control file using the Database Configuration Wizard is also described in *Implementing Siebel Business Applications on DB2 for z/OS*.

- 2 Specify the following values:
 - a On the Database Configuration Options screen, select the Validate Storage File option.
 - b On the Data Transport Method screen, select the Batch - Generate Unload/Load option.
 - c On the following screens, make sure you specify values for the *staging* database:
 - ODBC Data Source Name
 - Database User Name
 - Siebel Schema Qualifier screen
 - d On the Schema File screen, specify the following values:

- ❑ **Schema File:** Specify the directory path and filename of the file against which the extracted file is to be validated. For development environment upgrades, specify the *ddl.ctl* file. For production environment upgrades, specify the *schema.ddl* file.
 - ❑ **Storage Control File:** Specify the name of the storage control file you extracted and merged in [“Extracting the Storage Control File” on page 91](#). For development environment upgrades, specify *storage_upg_dev.ctl*. For production environment upgrades, specify *storage_upg_prod.ctl*.
- 3 When you have entered all the required values and reviewed them, launch the Siebel Upgrade Wizard to start the validation process. See [Chapter 10, “Running the Database Configuration Wizard to Perform Upgrade Tasks.”](#)
 - 4 When the validation is completed, review the log files, *dbvalidate_report.log* and *dbvalidate_schema.log*, that are generated in the *SIEBEL_ROOT\LOG\dbconfi g_val i date_mf\output* directory (Windows) or the *SIEBEL_ROOT/LOG/dbconfi g_val i date_mf/output* directory (UNIX).

If any validation errors occurred, correct them, then run the validation process again.

About Validation Errors

If you are validating a storage control file for an EBCDIC database, the validation process does not report all errors the first time it is run. You must rerun the validation process after fixing all reported errors. If new errors are reported on a subsequent validation, you must fix the errors and run the validation again. You may proceed after the validation reports no errors.

The most common reason that the validation process fails is because table spaces for the Siebel schema are not large enough to hold the new table definitions. If this error occurs, examine the *dbvalidate_report.log* file and identify the names of the buffer pools associated with the table spaces generating errors. Increase the bufferpool sizes as necessary in the *storage.ctl* file.

You can amend the storage control file using any of the methods described in [“Modifying the Storage Control File” on page 89](#). To amend buffer pool sizes associated with table spaces generating errors using the Database Storage Configurator (*dbconf.xls*), perform the following procedure.

To amend bufferpool sizes using *dbconf.xls*

- 1 Open *dbconf.xls* and select Enable Macros when prompted.
- 2 Import the storage control file that generated the validation errors:
 - a With the Home tab active, click Import.
 - b Go to the directory where your storage control file is located and double-click the appropriate file.
 - c When the import process is completed, click OK.
- 3 The following message appears:

Please enter default values for your system

Either amend the values for the displayed parameters, or accept the default values. (This screen does not appear if you have already set the default values.)

- 4 Click Set.
- 5 You are prompted to indicate whether or not you want to import row lengths. Select No.
- 6 Select the Functions tab, then click the Tools tab.
- 7 Click the Repair BP Validation button.
- 8 Select the log file generated by the validation process by double-clicking on the file, or selecting the file and clicking Open. This file lists the names of the buffer pools associated with the table spaces generating errors.
- 9 The Database Storage Configurator updates the buffer pool sizes for the table spaces generating errors in the storage control file. When the process is completed, the following message is displayed.

Bufferpools have been updated successfully!

- 10 Click OK.
- 11 Click the Home tab, and then click Export to save the amended storage control file. Save the file with the same filename. A message is displayed stating that the file will be validated (the syntax is validated).
- 12 Click OK. When the validation process is completed, a message is displayed if the file contains any values that you need to review.
- 13 Click OK. The values in the file that need to be reviewed are displayed, highlighted in red (errors) or yellow (warnings).
- 14 Make a note of the object that is generating the error or warning—the relevant *object type* tab and the object are highlighted. Click OK.
- 15 Navigate to the object that generated the validation warning or error by selecting Structures > *object type*. Amend the highlighted values if required, then export the file again.
- 16 When the validation process is completed successfully and you have exported the file, exit from the Database Storage Configurator.
- 17 Validate the storage control file against the target schema again using the Database Configuration Wizard.

For further information on using the Database Storage Configurator, see *Implementing Siebel Business Applications on DB2 for z/OS*.

Reviewing the Storage Control File

After you have extracted the storage control file, you must review it. Your storage control file is located in the directory that you specified when you ran the Database Configuration Wizard to extract the file. Check the following parameters in the control file and modify them as appropriate for your database needs. Once you have a storage layout that you are satisfied with, you are ready to continue with your upgrade.

NOTE: You must carefully review and edit the storage control file to meet your needs. Every time you modify the file, you must validate it again. See [“Validating the Extracted Storage Control File” on page 95](#).

Do not change the defaults for the following parameters in [Object 1] in the storage control file:

- IndexStogroup
- IndexBp
- PriQty
- SecQty

[Object 1]

Type = Defaults
Name = Defaults
Database = \$DbnamePrefix0000
Tablespace = SIEBTS00
Stogroup = \$StogroupTables
IndexStogroup = \$StogroupIndexes
IndexBp = \$IndexBufferPool
Bufferpool = \$4KBufferPool
Locksize = Page
Segsize = 32
LockMax = 0
PriQty = 48
SecQty = 1440
PctFree = 17
FreePage = 0
Compress = Yes
Define = No
Erase = No
CCSID = \$DbType

[Object 2]

Type = Database
Name = \$DbnamePrefix0000
Locksize = Page

[Object 3]

Type = Tablespace
Name = SIEBTS00
Database = \$DbnamePrefix0000
Bufferpool = \$4KBufferPool
Stogroup = \$StogroupTables
LockSize = PAGE
LockMax = 0
SegSize = 32
PriQty = 48
SecQty = 1440
PctFree = 17
FreePage = 0
Compress = No
Partitions = 0
Define = 0
Erase = 0

Reviewing EIM Table Partitioning and Data Distribution

Upgrades: All upgrades.

Environments: All environments.

Partitioning EIM table spaces can improve EIM processing performance. However, you need to check that the partitioning keys you are using distribute the data evenly across the table space partitions.

Siebel provides a partitioning column, `PARTITION_COLUMN`, designed specifically to produce even data distribution. This column is populated with data using a `BEFORE INSERT` trigger to generate the partitioning value for each row.

In releases of Siebel Business Applications before 8.0, the trigger that is used to populate `PARTITION_COLUMN` in the Siebel prepartitioned tables is based on the `ROW_ID` column, or another column, from a parent table. In Siebel 8.0, the `BEFORE INSERT` trigger generates a random number between 00 and 10 and uses this number to populate `PARTITION_COLUMN`.

When you upgrade from a pre-8.0 release of Siebel to Siebel 8.0, the `PARTITION_COLUMN` trigger is not changed. For Siebel prepartitioned tables that are partitioned based upon `PARTITION_COLUMN` using the 7.x release trigger, you may have to modify the partitioning values you use to achieve balanced partitions.

The mechanism by which EIM generates the `ROW_ID` can result in an uneven distribution of data if you use EIM with one of the default partitioning schemes to import data into a base table. When EIM imports data into a base table, EIM automatically generates the base table `ROW_ID` by concatenating the prefix with a unique number; for example, 1-SB3-123. EIM relies on database functions to make sure that each number is unique under the same prefix.

Uneven data distribution occurs because one of the default partitioning approaches uses the last two characters of `ROW_ID` as the partition key for partitioning of base tables; these characters can be letters, numbers, or a combination of letters and numbers, but the last two characters in a `ROW_ID` generated by EIM are numbers.

If your storage control file is to use partitioning keys that reflect the nature of your data, consider your data distribution with regard to your EIM process before you begin your database upgrade. Then, modify your partitioning keys accordingly. Alternatively, you can repartition the table after an uneven distribution has occurred.

Backing Up the Database

Upgrades: All upgrades.

Environments: Development environment only.

Perform a full backup of the database. This backup protects your repositories and environment.

It is a recommended practice that you back up your database at key stages of the upgrade:

- Before any upgrade activity is started
- After upgrading the Siebel Database Schema or Custom Database Schema (`upgprep + upgphys`)
- After the repository merge

Perform any necessary maintenance on your Siebel database, for example running `REORG` or `RUNSTATS`, before backing it up. This ensures that your database is ready for use if you have to perform a database recovery.

Granting a Siebel User Upgrade Authorization

Upgrades: All upgrades.

Environments: All environments.

The Siebel user who executes the Database Configuration Wizard and performs the upgrade must be set up as an employee on Siebel. This is the Siebel user whose user ID is entered when the Database Configuration Utility prompts for Database User Name.

The user name (user ID) of the target database needs to have authorization to set CURRENT SQLID and must have the Siebel administrator responsibility. SADMIN is the default administrator user name and password. If this user does not already exist in your database, or does not have Siebel administrator privileges, then you must add this to your database before starting the upgrade.

For further information on adding Siebel users, see *Siebel Security Guide*.

7

Preparing Siebel Application Data for Upgrade

This chapter covers the z/OS-specific tasks involved in preparing Siebel application data for upgrade. Also review the topics in the chapter of the *Siebel Database Upgrade Guide* that describes how to prepare Siebel application data for upgrade and perform any applicable tasks before starting your upgrade. This chapter includes the following topics:

- [Identifying and Resolving Duplicate Row IDs on page 103](#)
- [Preparing Siebel eChannel Data for Upgrade on page 104](#)
- [Preparing Forecasting Data for Upgrade on page 104](#)
- [Preparing Products and Quotes for Upgrade on page 105](#)
- [Preparing Financial Services Application Tables for Upgrade on page 106](#)
- [Preparing Siebel Configurator Data for Upgrade on page 106](#)
- [Setting the Value of the TYPE_CD Column on page 107](#)
- [Converting LONG VARCHAR Columns to CLOB Columns on page 107](#)

Identifying and Resolving Duplicate Row IDs

Upgrades: Release 6.2.1 only.

Environments: All environments.

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

During the upgrade from Release 6.2.1 to 8.0, ROW_IDs from S_EMPLOYEE, S_CONTACT, S_ORG_INT, S_ORG_EXT, S_POSTN, and S_ORG_GROUP are inserted into S_PARTY.ROW_ID. Although ROW_IDs are typically unique across the entire schema, there may be rare instances when ROW_IDs are shared across these tables.

If duplicate ROW_IDs are not resolved before the upgrade, the upgraded S_PARTY data will be defective because there will be mismatches between the base S_PARTY record and the corresponding extension table record.

After you install the Siebel Database Configuration Utilities software, but before you upgrade, you must identify and resolve any duplicate ROW_IDs in your Siebel database.

To identify and resolve duplicate ROW_IDs across tables

- 1 Run the script, `Find_DUP_S_PARTY_ROW_IDs.sql`, located in the following directory:
Windows: `DBSRVR_ROOT\DB2390`
UNIX: `DBSRVR_ROOT/DB2390`
This SQL script generates a list of duplicate ROW_IDs (if any) and the tables containing the rows.
- 2 Choose one of the records that has a duplicate ROW_ID.
Choose the record with the smallest number of references to other records.
- 3 Copy the record.
This creates a new record with a unique ROW_ID.
- 4 Use the Merge Records command to merge the original record with the copied record.
Records that were associated with the original record are now associated with the copied record.

Preparing Siebel eChannel Data for Upgrade

Upgrades: Release 6.2.1 only.

Environments: Production test, production.

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

For upgrades from Release 6.2.1 to 8.0, if you have modeled your partners as both Accounts and Divisions (or Organizations), merge these records so that there is only one record for each partner company.

Similarly, if you have modeled partner employees as both Contacts and Employees, merge these records so that there is one record for each person.

For assistance with merging records, contact your Oracle sales representative to request assistance from Oracle's Professional Services.

Preparing Forecasting Data for Upgrade

Upgrades: Release 6.2.1 only.

Environments: Production test, production.

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

For upgrades from Release 6.2.1 to 8.0, decide whether or not to upgrade the revenues for nonprimary sales team members. Consider upgrading nonprimary sales team members' revenues if your organization does any of the following:

- Uses nonprimary forecasting and wants to continue using a similar nonprimary forecasting approach. (Running queries on the Revenue table may not be adequate to meet this need.)

- Wants each sales team member to have a different opinion on the close date and revenue amount for each opportunity.
- Wants to assign, to each sales team member, some portion of each opportunity that is not related to a specific product or some percentage of the overall opportunity value.

For each opportunity in previous versions of Siebel Business Applications, the upgrade evaluates the opportunity sales team records for nonprimary sales team members. Each sales team member record is used to create a revenue record, if the following is true:

- The primary flag is not checked.
- The revenue amount is nonzero.
- The commit flag is checked.

Before upgrading, make sure that your opportunity sales team records are appropriately marked as committed or not committed. Records marked as committed are automatically upgraded; records marked as not committed are not upgraded.

Typically, uncommitted sales team records are not upgraded.

Preparing Products and Quotes for Upgrade

Upgrades: Release 6.2.1 only.

Environments: Production test, production.

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

Customizable Products

Determine which of your Release 6.2.1 configuration solutions are to be upgraded to Customizable Product quote items in Release 8.0. For Customizable Products, make sure that the Model Product field is checked in the Product Administration screen. This allows these products to be upgraded as Customizable Products.

For products that do not need to be upgraded as Customizable Products, verify that the Model Product field is unchecked. This verifies that quotes using these products in 6.2.1 are upgraded as Packaged products.

All Siebel Configurator model associations for these products are removed, and these products behave like hierarchical product bundles. For more information on Packaged products and Configurator, see *Siebel Product Administration Guide*.

Discount Amount Field

Read this section if you have implemented the Quotes line item Discount Amount Field so that it allows creation of a header-level discount when the field is either NULL or contains \$0.00.

The Siebel 6.2.1 default behavior is to populate this field with \$0.00, and the user must then clear this before entering a header-level discount.

In Release 8.0, the default for this field is NULL. This means the user does not have to clear the field to create a header-level discount.

If you have altered the default behavior so that the user can create a header-level discount when the Discount Amount field is either NULL or contains \$0.00, you must run the following script on the Siebel database before doing the upgrade:

```
update S_QUOTE_ITEM
set    DISCNT_AMT = NULL
where DISCNT_AMT = 0
```

Preparing Financial Services Application Tables for Upgrade

Upgrades: Release 6.2.1 only.

Environments: All environments.

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

DESC_TEXT is a column in two tables, S_CONDITION and S_PROD_BNFT. To broaden platform support, this column is reduced in length from 250 to 200 characters during the upgrade from Release 6.2.1 to Release 8.0.

Before the upgrade, examine these two tables for records that contain more than 200 characters. Manually reduce the size of these records to 200 characters or less.

Preparing Siebel Configurator Data for Upgrade

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

The Siebel Configurator's data model migration is embedded in the Siebel standard upgrade process. However, in order to make sure that the model is upgraded properly, customers must have appropriate model product associations. For more information on this and on other data migration issues when upgrading from release 6.2.1, see 477067.1 (Article ID) on My Oracle Support. This document was formerly published as Siebel Technical Note 428.

Setting the Value of the TYPE_CD Column

Upgrades: Releases 6.x, 7.0.x.

Environments: All environments.

At Siebel 7.5, S_SRC_PAYMENT.TYPE_CD became a required (NOT NULL) column. If TYPE_CD is null, the value is set to Payment during the upgrade.

Payment indicates that this is a payment from your organization to an external organization. For example, this might denote a payment of marketing funds to your customer.

Before the upgrade, review records where TYPE_CD is null. For those where you do not want the upgrade to set the value of TYPE_CD to Payment, set the value of TYPE_CD.

To set the value of TYPE_CD

- 1 In S_SRC_PAYMENT, query for records where TYPE_CD is null.
- 2 For the desired records, use Siebel EIM to set a value for TYPE_CD.
Use the values in the field's LOV.

Converting LONG VARCHAR Columns to CLOB Columns

Upgrades: Releases 6.x, 7.0.x, 7.5.x, 7.7.x

Environments: All environments.

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

In Siebel 8.0 and Siebel 7.8.x, the LONG VARCHAR columns of the following Siebel tables are converted to CLOB columns on z/OS to make sure data truncation problems do not occur:

- S_BITMAP_DATA
- S_DMND_CRTN_PRG
- S_EVT_MAIL
- S_NOTE
- S_NOTE_ACCNT
- S_NOTE_CON
- S_NOTE_OPTY
- S_SCHMST_DBSCPT
- S_SCHMSTEP_SCPT
- S_SERVICE_SCRPT

If you are upgrading to Siebel 8.0 from a pre-7.8.x release of Siebel Business Applications, if any of these tables are already in 32 KB buffer pools, convert the LONG VARCHAR columns in these tables to CLOB data types to make sure the columns are not truncated if, for example, a column is added to the table during the upgrade.

CAUTION: For pre-7.8.x upgrades, you must convert the SCRIPT column of the S_SERVICE_SCRPT table from a LONG VARCHAR data type to a CLOB data type before you begin the upgrade. If you do not, the repository merge process fails because the row length of the table exceeds the DB2 limit.

Changing the data type of the SCRIPT column in the S_SERVICE_SCRPT table involves deleting and re-creating the table. Ask your DBA to perform this task or create a service request (SR) on My Oracle Support requesting assistance.

8

Preparing a Development Environment for a Siebel Upgrade

This chapter describes the steps in preparing a development environment for upgrade. It includes the following topics:

- [Requirements for a Development Upgrade on page 109](#)
- [About Moving Tables on page 109](#)
- [Checking In Development Repository Projects on page 110](#)
- [Preparing EIM Tables for Upgrade on page 111](#)
- [Preparing for Intersection Table Maintenance and Deduplication on page 113](#)
- [Determining Which Template File Was Used During an Extract or Merge on page 114](#)

Requirements for a Development Upgrade

Upgrades: All upgrades.

Environments: Development environment only.

Before you upgrade your development environment, make sure that the development database configuration meets the database requirements outlined in [“Verifying Database Configuration” on page 87](#), and meets the requirements depicted in the *Siebel Installation Guide* for the operating system you are using.

If your development environment platform is DB2 UDB for Windows and UNIX, see *Siebel Database Upgrade Guide*.

NOTE: Make sure that the development database uses binary sort order. Production databases are not constrained by this requirement.

If you have not already done so, copy the Upgrade Planning Worksheet, located in [Appendix A, “Siebel Upgrade Planning Worksheet,”](#) and fill out the appropriate fields with the information you will need to perform the upgrade. Contact your database administrator or systems programmer for help in completing the worksheet. Also, refer to [“Information Required by the Database Configuration Wizard” on page 122](#) for a description of the information you are required to enter when you run the Database Configuration Wizard to perform upgrade operations.

About Moving Tables

Upgrades: All upgrades.

Environments: Development environment only.

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

Some of the Siebel tables will have columns added to them as part of the upgrade. When this happens, the length of the record will increase causing it to require a larger table space or bufferpool. Moving the identified tables to a new, larger and differently named table space allows you to maintain the model of multiple tables in each table space which is present for all pre-7.7 schemas.

Before you move the tables, you need to remove any standard or custom views using an SQL DROP command. Complete the following procedure to find the views that are defined on a table.

NOTE: The example used in the procedure assumes that you are creating a new 16K table space within the same database as the older, smaller table space.

To find views that are defined on a table

- Run the following SQL statement to produce the list of views:

```
select * from sysibm.sysVIEWdep  
  
WHERE Bcreator = 'yourschema'  
  
AND BNAME IN ('S_ASSET', 'S_PROD_INT');
```

This SQL statement produces the drop statements into a sequential dataset:

```
SELECT DISTINCT 'DROP VIEW ' || 'yourschema.' || V.DNAME || ' ;'  
  
FROM SYSIBM.SYSVIEWDEP V  
  
WHERE V.BNAME IN ('S_ASSET', 'S_PROD_INT')  
  
AND V.BNAME = 'S_ASSET'  
  
AND BCREATOR = 'yourschema';
```

To process these drop statements on the z/OS host, you need to create a PDS member with the output from the preceding statement. Then submit this member through DSNTEP2 using JCL.

You can also list these tables in the override.inp file. This will cause the tables to be recreated in a new database with one table space and one table.

If you are using the preconfigured storage control file, you do not need to move any tables as a preupgrade task. This is because the existing schema is not being preserved and all tables will be recreated in the 1:1:1 model.

Checking In Development Repository Projects

Upgrades: All upgrades.

Environments: Development environment only.

Developers who are using Siebel Tools with a local SQL Anywhere database must check in their projects to the development repository.

Make sure that all project locks in your current Siebel repository have been released to prevent inadvertent loss of development work during the upgrade.

If you are using Siebel Workflow Manager, you must run the Workflow Monitor Agent and Workflow Action Agent to completion before upgrading to Release 8.0. The S_ESCL_REQ table will not have any rows if Workflow Manager has completed successfully.

Preparing EIM Tables for Upgrade

Upgrades: All upgrades.

Environments: Development environment only.

This section describes how to drop EIM tables and prepare S_ASSET_POSTN and S_ETL_TIME_DAY for upgrade.

To drop EIM tables and prepare S_ASSET_POSTN and S_ETL_TIME_DAY for upgrade

- 1 Navigate to the *SIEBEL_ROOT*\dbsrvr\db2390\ (Windows) or \$SIEBEL_ROOT/dbsrvr/db2390/ (UNIX) directory and open the predevftp.txt file.
- 2 Edit the predevftp.txt file using the following information:
 - a Change &IP to the IP address of your DB2 host, for example, ZM01.
 - b Change &Username to the your own user name, for example, sadmi n.
 - c Change the constant SIEBELQ1 to your own dataset high-level qualifier (*DSNHLQ*), for example, sadmi n, using a command like Replace All.

Use the same high-level qualifier for all z/OS upgrade datasets. The *DSNHLQ* must be 18 characters or less and can have multiple nodes separated by periods. A node must be one to eight characters in length. The *DSNHLQ* may be the TSO ID.
 - d Change &SiebelRoot to your *SIEBEL_ROOT* directory, for example, *C: \sba80* (Windows).
- 3 After modifying the predevftp.txt file, save it, and then close the file.
- 4 Double-click ftppredev.bat (located in *SIEBEL_ROOT*\dbsrvr\db2390) or if you are using a UNIX operating system, issue the following command: `Ftp -i > ftpfile.bat`.

You are prompted to enter the password for the user name that you entered in predevftp.txt to replace &Username.
- 5 Enter the password, and then press Enter.

Files to prepare the development environment for the upgrade are transferred from the midtier to the z/OS host.
- 6 After the transfer is successful, log on to the mainframe, and navigate to the *DSNHLQ*.PREDEVO.JCL file, for example, SIEBELQ1.PREDEVO.JCL.
- 7 In the job card set the following parameters:

```
CLASS=S
```

```
MSGCLASS=X
```

```
JOBPARAM S=YOUR_HOST_NAME
```

If your Host server is QA01, the job card should look like the following:

```
//###PREDV JOB ACCNT#, CLASS=S, MSGCLASS=X, MSGLEVEL=(1, 1),
```

```
// TIME=1440, REGION=OM, NOTIFY=&SYSUID, LINES=9999
```

```
//*
```

```
/*JOBPARAM S=QA01
```

8 From the command line, execute the following commands:

■ C SIEBELQ1 *DSNHLQ* all

Where:

DSNHLQ is the high level qualifier you specified in [Step 2 on page 111](#).

■ C <SCHEMA> *YOUR_DB2_SCHEMA*

For example,

```
C <SCHEMA> DB2QUAL ALL
```

■ C <DEVHOST> *YOUR_HOST_LPAR*

For example,

```
C DEVHOST QA01 ALL
```

■ C <DB2LIB1> *YOUR_DB2_LOAD_DSN*

For example,

```
C DB2LIB1 DSN710.SDSNLOAD ALL
```

■ C <DB2LIB2> *YOUR_SECOND_DB2_LOAD_DSN*

For example,

```
C <DB2LIB2> DSN710.RUNLIB.LOAD ALL
```

■ C <DB2SYS> *YOUR_DB2_SUBSYSTEM*

For example,

```
C <DB2SYS> Q10K ALL
```

9 After submitting the job, enter cancel on the command line or press PF3 to save changes.

10 Verify that the job ran successfully:

a Verify that the RC=0.

b Verify that the following datasets were created.

NOTE: If you are upgrading from a FINS 7.0.4 application, you need to use the SIS 7.0.4 library.

- For upgrades of Siebel Business Applications:

DSNHLO.PRE.HOR621.JCLLIB

DSNHLO.PRE.HOR704.JCLLIB

DSNHLO.PRE.HOR752.JCLLIB

- For upgrades of Siebel Industry Applications:

DSNHLO.PRE.SIA621.JCLLIB

DSNHLO.PRE.SIA752.JCLLIB

DSNHLO.PRE.SIS63.JCLLIB

DSNHLO.PRE.SIS704.JCLLIB

- 11 Edit the dataset from the list above that applies to your upgrade path.

NOTE: Each library has an @@README member to guide you through what to submit.

- 12 Submit the jobs as instructed in the @@README file.

- 13 Verify that the job ran successfully, RC=0 or RC=4.

- 14 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Preparing for Intersection Table Maintenance and Deduplication

Upgrades: All upgrades.

Environments: Development environment only.

Perform the following task to create the storage_dedup.ctl file. This file is used for deduplication procedures and intersection table maintenance.

To create a storage_dedup.ctl file

- 1 Create a copy of the storage_dedup.ctl file located in the DB2390 directory under *DBSRVR_ROOT*.

- 2 Rename the copy of storage_dedup.ctl to preserve it as a backup, for example, save it as ORIGstorage_dedup.ctl.

- 3 Navigate to the *DBSRVR_ROOT\DB2390* directory and open dbconf.xls.

NOTE: If your midtier is UNIX, then the dbconf.xls and storage_dedup.ctl files need to be transferred to Windows using the ftp command. The dbconf.xls file needs to be transferred in *binary mode* while the storage_dedup.ctl file needs to be in *ASCII mode*. Using these transfer modes prevents file corruption.

- 4 Select the HOME tab page, and click Import.

- 5 When you are prompted to select the `storage_dedup` file, navigate to the `DBSRVR_ROOT\DB2390` directory and select `storage_dedup.ctl`.

The default values screen appears.

- 6 On the default values screen, enter the values for your system:

- a Enter your values for the following parameters:

- Table Storage Group for Table
- Index Storage Group for Indexes
- 4-KB Buffer Pool Name
- 16-KB Buffer Pool Name
- Index Buffer Pool Name

- b For the Database Name Prefix, replace the placeholder, `SIDB`, with your database name prefix.

Your database name prefix must be a maximum of four characters. This may be the last four letters of your tableowner ID.

- c For the Encoding Scheme, specify your database encoding scheme, either ASCII or EBCDIC.

After entering and verifying all values, click Set.

- 7 Return to the HOME tab page, and click Export.

Save this database configuration as `storage_dedup.ctl` in the `DBSRVR_ROOT\DB2390` directory.

Determining Which Template File Was Used During an Extract or Merge

Upgrades: All upgrades.

Environments: Development environment only.

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

This section describes how to determine which template file that was used for an extract/merge when a storage control file is created.

To determine the template file that was used for an extract/merge

- 1 Open the `dbextract. log` file and review the command line that was entered.
- 2 Check the value for the `/i` parameter. You can also check the `upgwi z. log` file for the `strgxrct. exe` command string.

```
/i D:\18025\dbsrvr\db2390\storage_p_e.ctl
```

```
2004-01-28 09:10:32D:\18025\si_ebsrvr\bin\strgxtxct.exe /u QADMIN /p ***** /c Q10K /
d yourschema /I D:\18025\si_ebsrvr\log\dbconfig_extract_merge_mf_lk5\output/
dbextract.log /i D:\18025\dbsrvr\db2390\storage_p_e.ctl /o D:\18025\dbsrvr\db2390/
SStempstore.txt /v D:\18025\dbsrvr\db2390\overri de. i np /8 D029 /9 SIEBTS /1 SYSDEFLT
/2 SYSDEFLT /3 BP1 /4 BP2 /7 EBCDIC
```


9

Preparing a Production Environment for a Siebel Upgrade

This chapter describes the steps in preparing a production environment for upgrade. It includes the following topics:

- [Requirements for a Production Upgrade on page 117](#)
- [About Moving the Customized Repository and Schema Definition Files on page 117](#)
- [Renaming the Production Environment Repository on page 118](#)
- [Preparing for a No-Development-Environment Siebel Upgrade on page 119](#)

Requirements for a Production Upgrade

Upgrades: All upgrades.

Environments: Production test, production.

CAUTION: You must be thoroughly familiar with the upgrade process before beginning the production upgrade. Before upgrading your production environment, perform a test upgrade in your production test environment to familiarize yourself with the process and to eliminate errors that can affect upgrade success or performance.

Before beginning the upgrade of your production environment, verify that the production database configuration meets the database requirements outlined in the section [“Verify Database Configuration” on page 87](#).

If you have not already done so, copy the Upgrade Planning Worksheet, located in [Appendix A, “Siebel Upgrade Planning Worksheet,”](#) and fill out the appropriate fields with the information you will need to perform the upgrade. Contact your database administrator or systems programmer for help in completing the worksheet. Also, refer to [“Information Required by the Database Configuration Wizard” on page 122](#) for a description of the information you are required to enter when you run the Database Configuration Wizard to perform upgrade operations.

About Moving the Customized Repository and Schema Definition Files

Upgrades: All upgrades.

Environments: Production test and Production environments.

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

When you upgraded your development environment, the new customized repository was exported to a file called `custrep.dat` and the modified schema definition was exported to a file called `schema.ddl` in the `$IEBEL_ROOT\dsrvr\db2390\` directory (Windows) or the `$IEBEL_ROOT/dsrvr/db2390/` (UNIX) directory on the Siebel Server client machine from which you ran the upgrade. These files are used as the schema input on the production upgrade.

If the development and production upgrades are run on different mid-tier machines, then you must copy the `schema.ddl` and `custrep.dat` files to the production mid-tier machine before running the production upgrade.

If you modify repository objects or schema definitions after completing the development upgrade (`upgphys`), you must regenerate the `schema.ddl` and `custrep.dat` files. See [“Regenerating the Siebel Repository Definition Files” on page 198](#) for further information. You must then copy the files from the development to the production mid-tier machine again.

In the production environment, the `custrep.dat` file is used by the Siebel Upgrade Wizard to import the New Customer Repository and the `schema.ddl` file is used by the Siebel Upgrade Wizard to create the new database schema.

Renaming the Production Environment Repository

Upgrades: All upgrades.

Environments: Production test, production.

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

Two separate repositories are used during the production upgrade process:

- **Your existing production repository**
- **New Customer Repository**

The New Customer Repository is loaded when you run the Siebel Upgrade Wizard.

To prevent a naming conflict, before you take your production database offline to run the in-place target database upgrade, rename your existing production repository (*Siebel Repository*) to *Prior Customer Repository*. After the upgrade, your new Release 8.0 production repository is given the name *Siebel Repository*.

Rename your existing production repository using the procedure described in the chapter in *Siebel Database Upgrade Guide* that describes how to upgrade the Siebel database.

CAUTION: Your upgrade will encounter errors if you have more than one existing repository for a production upgrade. Export, archive, and delete from the Siebel schema to be upgraded any redundant repositories before you upgrade your production environment.

For further information about renaming repositories, see *Configuring Siebel Business Applications*.

Preparing for a No-Development-Environment Siebel Upgrade

Upgrades: All upgrades.

Environments: Production test, production.

Platforms: All platforms.

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

If your installation does not include a development environment, you do not need to merge your Siebel Repository. Instead, you can use the repository and schema definition files included in the Siebel Database Server installation.

Before performing the upgrade, you must move and rename these files.

To prepare for a no-development-environment upgrade

- 1 Navigate to `DBSRVR_ROOT\common` (Windows) or `DBSRVR_ROOT/common` (UNIX) 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\DB2390` (Windows) or `DBSRVR_ROOT/DB2390` (UNIX) directory.
- 4 In the `DB2390` directory, copy the `ddl.ctl` 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.

10 Running the Database Configuration Wizard to Perform Upgrade Tasks

This chapter describes how to run the Database Configuration Wizard to perform the midtier tasks for an upgrade. Also review the topics in the chapter of the *Siebel Database Upgrade Guide* that describes how to upgrade the Siebel database and perform any applicable topics before starting your upgrade. Refer to the roadmap for your upgrade in [Chapter 4, “How to Perform a Siebel Database Upgrade”](#) for a complete list of all the tasks you must perform to upgrade to Siebel 8.0.

This chapter contains the following topics:

- [Information Required By the Database Configuration Wizard on page 121](#)
- [About Running the Database Configuration Wizard Under Windows on page 126](#)
- [About Running the Database Configuration Wizard Under UNIX on page 129](#)
- [Starting the Siebel Upgrade Wizard on page 132](#)
- [Upgrading the Repository and Importing Seed Data on page 135](#)

Information Required By the Database Configuration Wizard

Upgrades: All upgrades.

Environments: All environments.

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

Use this topic to identify the information you must enter when running the Database Configuration Wizard. Make sure you have prepared your upgrade environment, and have collected and verified this information before running the wizard.

The Database Configuration Wizard requests information about the upgrade process you want to perform. It then adds this information to a master upgrade file. The Siebel Upgrade Wizard runs after the Database Configuration Wizard exits. The wizard generates the upgrade SQL files and executes some of the SQL files against the Siebel database.

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

Information Required by the Database Configuration Wizard

Table 10 lists the information that you must enter in the Database Configuration Wizard when performing the upgrade processes. Table 11 on page 126 lists additional information the utility requires when you perform a production environment upgrade.

Table 10. Information Required for the Database Configuration Wizard—Upgrade Database

Field Name or Menu	Required Information
Gateway Name Server Address	The Siebel Gateway Name Server machine name.
Enterprise Server Name	The Enterprise Server name.
Siebel Server Directory	The absolute path of the directory where the Siebel Server is installed. For UNIX, do <i>not</i> enter the string \$SIEBEL_ROOT.
Siebel Database Server Directory	The absolute path of the directory where the Siebel Database Configuration Utilities are installed—for example C: \sba80\dbsrvr (Windows) or siebel /dbsrvr (UNIX).
RDBMS Platform	Choose IBM DB2 UDB for z/OS.
Siebel Database Operation menu	Choose Upgrade Database. The other menu choices are for database installation and administration.
Environment Type	<ul style="list-style-type: none"> ■ Choose Development for development environment upgrades. ■ Choose Production for production and production test environment upgrades.
Upgrade Options	Choose one of the following: <ul style="list-style-type: none"> ■ Upgrade Siebel Database Schema (upgrep): development upgrades only ■ Upgrade Custom Database Schema (upgphys): development upgrades only ■ Prepare for Production Upgrade: production upgrades only ■ Upgrade Siebel Database Schema (upgrep + upgphys): production upgrades only
Upgrade Process	Choose one of the following: <ul style="list-style-type: none"> ■ zSeries Staging of Files for Upgrade: to create the staging database DDL and to generate upgrade files ■ zSeries Seed/Repository Upgrade: to automatically populate and upgrade data on the database

Table 10. Information Required for the Database Configuration Wizard—Upgrade Database

Field Name or Menu	Required Information
Siebel Industry Application	<p>This screen appears if you are upgrading from a Siebel Industry Solution or Siebel Industry application.</p> <p>Choose the application you are upgrading from.</p> <p>If you have upgraded to the base Siebel Business application as part of upgrading to the new Siebel Industry application release, choose Siebel Business application (HOR).</p>
Current Siebel Version	Choose the application version you are upgrading from.
Database Encoding	Indicate whether your database uses an ASCII or EBCDIC code page.
Language Selection	Upgrades with more than one language pack installed only. Choose the primary (base) language that is installed. The primary language is the language in which the data is stored in the Siebel database that is being upgraded.
Host/LPAR name where Target database resides	<ul style="list-style-type: none"> ■ Host/LPAR name where Target database resides. The name of the host or LPAR where the target database is located. You can either specify the same or different Host/LPAR names for the target and staging databases.
DB2 Subsystem name of Target database.	<ul style="list-style-type: none"> ■ DB2 Subsystem name of Target database. The DB2 subsystem name where the target database is located. You must specify a <i>different</i> DB2 subsystem name for the target and staging databases.
Schema/Tableowner qualifier name on Target database	Enter the up to eight-character identifier that designates the Siebel schema for your target database. This is also an authorization ID. The schema qualifier must start with a letter, cannot contain special characters, and must be entered in uppercase. The target database tableowner name can be the same or different to the staging database tableowner name.
ODBC Data Source Name of Target Database	<p>Verify the ODBC name for connecting to the target Siebel database.</p> <p>The default value of the target database ODBC DSN is the DB2 subsystem name. When you set up the ODBC connection in DB2 Connect, you can use the actual subsystem name for the database alias.</p> <p>NOTE: To find the name of your ODBC data source, navigate to Start > Settings > Control Panel > ODBC data source. Click the System DSN tab and you will find the name of your ODBC data source.</p> <p>To find the name of your ODBC data source on UNIX, type: vi \$ODBCINI.</p>

Table 10. Information Required for the Database Configuration Wizard—Upgrade Database

Field Name or Menu	Required Information
Valid/Authorized Target database user name or group name	<p>Enter target database user name and password for the Siebel administrator of the target database.</p> <p>For further information on the database user name, see “Granting a Siebel User Upgrade Authorization” on page 100.</p>
<p>Host/LPAR name where staging database resides</p> <p>DB2 Subsystem name where staging database resides</p>	<ul style="list-style-type: none"> ■ Host/LPAR name where Staging database resides. The host or LPAR name of the staging or target database. The staging and target Host/LPAR names can be the same or different. ■ DB2 Subsystem name where Staging database resides. The DB2 subsystem name of the database. You must specify <i>different</i> DB2 subsystem names for the target and staging databases.
Schema/Tableowner qualifier name for Staging database	<p>Enter the up to eight-character identifier that designates the Siebel schema for your staging database. This is also an authorization ID. The schema qualifier must start with a letter, cannot contain special characters, and must be entered in uppercase. The staging database tableowner name can be the same or different to the target database tableowner name.</p>
Authorized TSO account ID used to connect and FTP files to Enterprise Server(s)	<p>Enter your TSO account ID. This account ID must have the authorization to allocate and create datasets on the z/OS host.</p>
Dataset High-level Qualifier for all Host (Staging and Target) dataset names	<p>Specify the high-level qualifier you want to use for the z/OS upgrade datasets. Follow your organization's naming standards.</p>
Security Group ID / Grantee.	<p>Enter the user ID of the group to whom schema access is granted, for example, SSEROLE.</p>

Table 10. Information Required for the Database Configuration Wizard—Upgrade Database

Field Name or Menu	Required Information
Storage Group for Temporary Indexes	Storage Group for Temporary Indexes. Enter the name of the storage group provided by the database administrator (the default value is SYSDEFLT). The staging database storage group name can be the same or different to the target database storage group name.
Storage Control File	<p>Storage Control File. Enter the path and name of the storage control file you want to use as follows:</p> <ul style="list-style-type: none"> ■ zSeries Staging of Files for Upgrade: When you select this upgrade process, specify the storage control file of the target database. ■ zSeries Seed/Repository Upgrade: When you select this upgrade process, specify the storage control file that contains the previously customized database storage layout (this is the file you prepared in “Preparing the Storage Layout of the Schema” on page 88):
Primary Quantity for Temporary Index Space	Enter the primary and secondary quantities for temporary index space.
DDL Commit Frequency	Enter the DDL commit frequency for your upgrade.
Output Directory	<p>Accept the default or enter the directory name.</p> <p>NOTE: When this process is complete, this directory contains all of the files necessary to create the staging database or run the upgrade. These files must be manually applied by the database administrator.</p>

Additional Information Required for Production Upgrades

When you perform a production environment upgrade, you are prompted to enter the additional information shown in [Table 11 on page 126](#) when you run the Database Configuration Wizard in Prepare for Production Upgrade mode.

Note that several screens request information about the Siebel database in the development environment, not the production environment.

Table 11. Additional Information Required for Production Upgrades

Screen Name	Required Information
ODBC Data Source Name for Development Database	Windows only. The ODBC name for connecting to the development environment Siebel database. If you are upgrading without a development environment, this is the ODBC of the reference database.
Database User Name for Development Database	Account name and password of the Siebel administrator of the Siebel database in the development environment.
Siebel Schema Qualifier for Development Database	Enter the up to eight-character identifier that designates the Siebel schema for your development database. This is also an authorization ID. The schema qualifier must start with a letter, cannot contain special characters, and must be entered in uppercase.
Repository Name for Development Database	Enter the name of the upgraded Siebel Tools repository in the development environment database. Typically, this is <i>Siebel Repository</i> .

About Running the Database Configuration Wizard Under Windows

Upgrades: All upgrades.

Environments: All environments.

Platforms: Windows only.

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

Run the Database Configuration Wizard to upgrade the Siebel database. The wizard collects information, populates a master configuration file, and calls the SQL generator to create SQL commands. The Siebel Upgrade Wizard then uses the configuration file and SQL commands to generate upgrade files that are applied on the mainframe, and to make upgrade changes to the Siebel database.

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

Requirements:

- Collect the information that the Database Configuration Wizard requires. See ["Information Required By the Database Configuration Wizard" on page 121](#).

- Install the new release's languages packs for all deployed languages. For further information on multilingual upgrades, see the topic on upgrade planning for multilingual Siebel deployments in the *Siebel Database Upgrade Guide*. For information on multilingual upgrades from Release 6.2.1, also see ["About Multilingual Deployments" on page 85](#).

This topic describes two different procedures for starting the Database Configuration Wizard: one is the standard method of running the utility, the other method is used if you deploy languages that were not shipped by Oracle. The procedures are explained as follows:

- ["Running the Database Configuration Wizard Under Windows" on page 127](#).

This is the standard method of running the Database Configuration Wizard.

If you have deployed languages that are not shipped with the Siebel 8.0 product, or have languages within your system that need to be removed, this procedure will not complete successfully. You receive an error message stating that your present installation is incomplete, and a list of the languages that caused the error is displayed. If this error occurs, you must run the Database Configuration Wizard using the `dbsrvr_lang.scm` model file, as described in ["Running the Database Configuration Wizard With Unshipped Languages \(Windows\)" on page 128](#).

For a list of languages shipped by Oracle, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

NOTE: If your system only contains shipped languages, and you still receive an installer warning, review the records in the `S_LST_OF_VAL` database table using Siebel Tools. For more information, see the topic on preparing for a multilingual upgrade in the *Siebel Database Upgrade Guide*.

- ["Running the Database Configuration Wizard With Unshipped Languages \(Windows\)" on page 128](#).

Use this procedure only if your configuration operation fails because there are unshipped languages deployed within your system. This procedure allows you to run the Database Configuration Wizard, which will validate all data in shipped languages only. You can manually verify the data in unshipped languages following the successful completion of the Database Configuration Wizard.

Running the Database Configuration Wizard Under Windows

This topic describes the standard procedure to follow to run the Database Configuration Wizard under Windows.

To run the Database Configuration Wizard under Windows

- 1 Stop all Siebel Servers by navigating to Start > Settings > Control Panel > Services.

NOTE: The Database Configuration Wizard runs in live mode only so you must be connected to the Gateway Name Server to run it. For further information on Siebel Configuration Wizard running modes, see *Siebel Installation Guide* for the operating system you are using.

- 2 Select Start > Programs > Siebel Enterprise Server Configuration 8.0 > Database Server configuration.

The first screen of the Database Configuration Wizard appears.

- 3 Enter the information you are prompted for in each screen, and click Next to continue.
- 4 When the Configuration is Complete screen appears, select one of the following options, and click Next:
 - **Yes apply configuration changes now.** The configuration information you entered is saved and you can choose to launch the Siebel Upgrade Wizard in [Step 7](#).
 - **No I will apply configuration changes later.** The configuration information is saved but you can *not* choose to launch the Siebel Upgrade Wizard in [Step 7](#).
- 5 On the Configuration Parameter Review screen, review the configuration values you entered on the previous screens. To change any of the values, click Back to return to the screen with the parameter you need to change. If the values are correct, click Next to continue.
- 6 You are prompted as to whether you want to execute the configuration:
 - Click No if you decide you do not want to continue with the upgrade process. The configuration information you have entered is *not* saved. You must enter the database configuration parameters again.
 - Click Yes to continue. The configuration information you have entered *is* saved.
- 7 Depending on the option you selected in [Step 4](#), do one of the following:
 - If you selected the **No I will apply configuration changes later** option, click OK to finish. The configuration information is saved in a master file located in `SIEBEL_ROOT\bin` but the Upgrade Wizard is not launched. You can restart the configuration and run the Upgrade Wizard later. See [“Starting the Siebel Upgrade Wizard” on page 132](#).
 - If you selected the **Yes apply configuration changes now** option in [Step 4 on page 128](#), the configuration information you entered is saved. Click OK and the Siebel Upgrade Wizard is launched; it calls the SQL generator to create or populate SQL scripts.

Running the Database Configuration Wizard With Unshipped Languages (Windows)

This topic describes the procedure to follow to run the Database Configuration Wizard under Windows if your system deploys languages that were not shipped by Oracle.

To run the Database Configuration Wizard using the `dbsrvr_lang.scm` model file

- 1 Verify that no server tasks are running in the background.
If necessary, stop all Siebel Servers by navigating to Start > Settings > Control Panel > Services.
- 2 Navigate to the `SIEBEL_ROOT\admin` directory and do the following:
 - a Back up the `dbsrvr.scm` file, then rename the file to `dbsrvr_orig.scm`.
 - b Rename `dbsrvr_lang.scm` to `dbsrvr.scm`.
- 3 Select Start > Programs > Siebel Enterprise Server Configuration 8.0 > Database Server configuration to start the Database Configuration Wizard.
You can now use the Database Configuration Wizard as described in [Step 3](#) through [Step 7 on page 128](#).

About Running the Database Configuration Wizard Under UNIX

Upgrades: All upgrades.

Environments: All environments.

Platforms: UNIX only.

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

Run the Database Configuration Wizard to upgrade the Siebel database. The wizard collects information, populates a master configuration file, and calls the SQL generator to create SQL commands. The Siebel Upgrade Wizard then uses the configuration file and SQL commands to generate upgrade files that are applied on the mainframe, and to make upgrade changes to the Siebel database.

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

Requirements:

- Collect the information that the Database Configuration Wizard requires. See [“Information Required By the Database Configuration Wizard” on page 121.](#)
- Install the new release’s languages packs for all deployed languages. For further information on multilingual upgrades, see the topic on upgrade planning for multilingual Siebel deployments in the *Siebel Database Upgrade Guide*. For information on multilingual upgrades from Release 6.2.1, also see [“About Multilingual Deployments” on page 85.](#)

This topic describes two different procedures for starting the Database Configuration Wizard: one is the standard method of running the utility, the other method is used if you deploy languages that were not shipped by Oracle. The procedures are explained as follows:

- [“Running the Database Configuration Wizard Under UNIX” on page 130](#)

This is the standard method of running the Database Configuration Wizard.

If you have deployed languages that are not shipped with the Siebel product, or have languages within your system that need to be removed, this procedure will not complete successfully. You receive an error message stating that your present installation is incomplete, and a list of the languages that caused the error is displayed. If this error occurs, you must run the Database Configuration Wizard using the `dbsrvr_lang.scm` model file, as described in [“Running the Database Configuration Wizard With Unshipped Languages \(UNIX\)” on page 131.](#)

For a list of languages shipped by Oracle, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

NOTE: If your system only contains shipped languages, and you still receive an installer warning, review the records in the `S_LST_OF_VAL` database table using Siebel Tools. For more information, see the topic on preparing for a multilingual upgrade in the *Siebel Database Upgrade Guide*.

- [“Running the Database Configuration Wizard With Unshipped Languages \(UNIX\)” on page 131](#)

Use this procedure only if your configuration operation fails because there are unshipped languages deployed within your system. This procedure allows you to run the Database Configuration Wizard, which will validate all data in shipped languages only. You can manually verify the data in unshipped languages following the successful completion of the Database Configuration Wizard.

Running the Database Configuration Wizard Under UNIX

This topic describes the standard procedure to follow to run the Database Configuration Wizard under UNIX.

To run the Database Configuration Wizard under UNIX

- 1 Verify that the Siebel Server is stopped.

NOTE: The Database Configuration Wizard runs in live mode only so you must be connected to the Gateway Name Server to run it. For further information on Siebel Configuration Wizard running modes, see *Siebel Installation Guide* for the operating system you are using.

- 2 Make \$SIEBEL_ROOT the current directory.

- 3 Source environment variables from the \$SIEBEL_ROOT directory:

```
Korn: . siebenv.sh
```

```
C shell: source siebenv.csh
```

- 4 Review the values of the following environment variables and confirm the settings are correct:

- SIEBEL_ROOT. This path must end in siebsrvr. For example, /usr/siebel/siebsrvr.
- LANGUAGE. This is the language in which the Database Configuration Wizard runs. The value of this variable is a language identifier string. For example, enu is the identifier string for English.

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

- 5 Start the Database Configuration Wizard by running the following command:

```
$SIEBEL_ROOT/bin/ssincfgw -args MODEL_FILE=$SIEBEL_ROOT/admin/dbsrvr.scm  
MODE=LIVE
```

The first Database Configuration Wizard screen appears. Enter the information you are prompted for in this screen, and click next to continue.

- 6 Enter the information you are prompted for in all subsequent screens. Use the Next and Back buttons to navigate between screens.
- 7 After you have entered all the requested information, the wizard displays the following message:

Configuration is complete: configuration parameters will be saved to `<$Masterfile>` file when the wizard completes. Please run the following command line after you exit from this configuration wizard. This command will deploy the process you configured to the database.

```
$SiebelRoot/siebsvr/bin/srvrupgwi z /m $SiebelRoot/siebsvr/bin/$Masterfile>
```

- 8 Click Next to continue. The utility displays the Parameter Review screen listing all the values you have entered.
- 9 To amend any of the configuration values, click Back to return to the appropriate screen and make changes. Otherwise, click Next.
- 10 You are prompted as to whether or not you want to execute the configuration:
 - Click Yes, and the configuration information is saved in a master file located in `$SIEBEL_ROOT/bin` but the Upgrade Wizard is not launched. To start the Upgrade Wizard, see [“Starting the Siebel Upgrade Wizard” on page 132](#).
 - Click No, and the configuration information you entered is not saved.

Running the Database Configuration Wizard With Unshipped Languages (UNIX)

This topic describes the procedure to follow to run the Database Configuration Wizard under UNIX if your system deploys languages that were not shipped by Oracle.

To run the Database Configuration Wizard using the `dbsvr_lang.scm` model file

- 1 Verify that no server tasks are running in the background.
- 2 Make `$SIEBEL_ROOT` the current directory.
- 3 Source the environment variables from the `$SIEBEL_ROOT` directory:

```
Korn shell: siebenv.sh  
C shell: source siebenv.csh
```
- 4 Review the values of the following environment variables and confirm the settings are correct:
 - `SIEBEL_ROOT`. This path must end in `/siebsvr`. For example, `/usr/siebel/siebsvr`.
 - `LANGUAGE`. This is the language in which the Database Configuration Wizard runs. The value of this variable is a language identifier string. For example, `enu` is the identifier string for English.
- 5 Navigate to the `SIEBEL_ROOT/admin` directory and do the following:
 - a Back up the `dbsvr.scm` file, then rename the file to `dbsvr_orig.scm`.
 - b Rename `dbsvr_lang.scm` to `dbsvr.scm`.
- 6 Start the Database Configuration Wizard by running the following command:

```
$SIEBEL_ROOT/bin/ssincfgw -args MODE=LIVE MODEL_FILE=$SIEBEL_ROOT/admin/dbsvr.scm
```

- 7 You can now use the Database Configuration Wizard as described in [Step 6 on page 130](#) through [Step 10 on page 131](#).

Starting the Siebel Upgrade Wizard

Upgrades: All upgrades.

Environments: All environments.

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

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

As the Upgrade Wizard performs the steps in the driver file, it lists the steps in a state log. The state log is located in `\si ebsrvr\LOG\process\state` where *process* is the upgrade process, for example `upgprep_dev_752` (upgrade from 7.5.2, upgprep process, development environment).

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

When you run the Database Configuration Wizard under Windows, you are prompted to indicate whether or not you want to start the Upgrade Wizard. When you run the Database Configuration Wizard under UNIX, you must start the Upgrade Wizard manually. For more information on the Siebel Upgrade Wizard, see [“About the Siebel Upgrade Wizard and Driver Files” on page 36](#).

About Starting the Siebel Upgrade Wizard After Errors

If the Siebel Upgrade Wizard stops due to errors, 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.
- If you are continuing a previous and incomplete schema upgrade, do not change the Log Output Directory that you previously selected.
- If problems with your environment prevent the upgrade from restarting, you must restore the database from the prior base version (the version from which you are upgrading).

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

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

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

Starting the Siebel Upgrade Wizard

Use this procedure to start the Upgrade Wizard. See [“Restarting the Siebel Upgrade Wizard After Pauses” on page 134](#) for the procedure to restart the Upgrade Wizard after it has paused and [“Stopping the Siebel Upgrade Wizard” on page 134](#) for the procedure to stop the Upgrade Wizard.

To start the Upgrade Wizard

- 1 Navigate to the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Enter the following command from the command line:

- Windows: `siebug /m master_UPGRADEOPTION_ENVIRONMENT_VERSION_MasterFileType.ucf`

- UNIX: `srvrupgwi z /m master_UPGRADEOPTION_ENVIRONMENT_VERSION_MasterFileType.ucf`

where:

`UPGRADEOPTION_ENVIRONMENT_VERSION_MasterFileType` is the portion of the upgrade configuration file name that lists upgrade process, upgrade environment, the Siebel release from which you are upgrading, and the master file type. The file is located in `SIEBEL_ROOT\bin` (UNIX: `$SIEBEL_ROOT/bin`). See [“About the Database Configuration Wizard” on page 32](#) for further information.

NOTE: Specify either `mf_m` or `mf` for the `MasterFileType` value depending on whether you want to start the manual or automatic portion of an Upgrade Wizard process.

The following table lists examples of some of the file names for an upgrade from Siebel 7.5.x.

Upgrade Mode	File Name
Development env. upgrep	<code>master_upgrep_dev_752_mf_m.ucf</code>
	<code>master_upgrep_dev_752_mf.ucf</code>
Development env. upgphys	<code>master_upgphys_dev_752_mf.ucf</code>
Prepare for Production	<code>master_prepare_for_production_upgrade.ucf</code>
Production env. upgrep and upgphys	<code>master_upgrep_prod_752_mf_m.ucf</code>
	<code>master_upgrep_prod_752_mf.ucf</code>

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

The Upgrade Wizard resumes from the point at which it stopped.

Restarting the Siebel Upgrade Wizard After Pauses

Use this procedure to restart the Upgrade Wizard after it has paused to allow you to apply files generated on the midtier to the z/OS host.

To restart the Upgrade Wizard

- 1 Navigate to the `SIEBEL_ROOT\bin` directory (Windows) or the `$SIEBEL_ROOT/bin` directory (UNIX).
- 2 Do the following:
 - Windows: Double-click on the `upg_restart.bat` file
 - UNIX: Run `upg_restart.ksh`

Stopping the Siebel Upgrade Wizard

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

If you are unsure whether or not the Upgrade Wizard has stopped responding, create a service request (SR) on My Oracle Support, or phone Global Customer Support directly to create a service request.

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

The following procedure describes how to stop the Upgrade Wizard in a Windows environment.

To stop the Upgrade Wizard under Windows

- Do one of the following:
 - If the Upgrade Wizard has launched a separate command window in which a utility is running, close the command window. This terminates the utility and stops the upgrade.
 - In the Upgrade Wizard dialog box, click Cancel.

The Upgrade Wizard will exit when the current upgrade step is completed. There might be a delay while the step completes.

The following procedure describes how to stop the Upgrade Wizard in a UNIX environment.

To stop the Upgrade Wizard under UNIX

- 1 If the Upgrade Wizard has started a utility in a child process, stop the child process.
- 2 Exit the shell in which the Upgrade Wizard is running.
- 3 Locate and stop any orphaned child processes started by the Upgrade Wizard.

After the processes terminate, there may be a delay while the RDBMS executes already-issued SQL commands.

Upgrading the Repository and Importing Seed Data

Upgrades: All upgrades.

Environments: All environments.

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

After you have created the staging database, generated the upgrade files, and upgraded the target database, you must upgrade the repository and import seed data.

Run the Database Configuration Wizard to upgrade the repository and import seed data by following the procedure below.

To upgrade the repository and import seed data

- 1 Run the Database Configuration Wizard. See [“About Running the Database Configuration Wizard Under Windows” on page 126](#) and [“About Running the Database Configuration Wizard Under UNIX” on page 129](#).
- 2 Select the following options:
 - **Upgrade Options:**
 - Development environment: Upgrade Siebel Database Schema (upgrep)
 - Production environment: Upgrade Siebel Database Schema (upgrep + upgphys)
 - **Upgrade Process:** zSeries Seed/Repository Upgrade
 - Make sure you specify values for the target database when prompted for the names of the Host, DB2 subsystem, schema qualifier, ODBC data source, and database user name and password.
- 3 Enter configuration information for your environment as described in [“Information Required by the Database Configuration Wizard” on page 122](#). The master UCF file is updated with the environment configuration information.
- 4 Start the Siebel Upgrade Wizard. See [“Starting the Siebel Upgrade Wizard” on page 132](#).

NOTE: The process of importing seed data and upgrading the repository may take several hours.

Restarting the Seed Data Import and Repository Upgrade

If the Upgrade Wizard stops responding during the repository upgrade process, restart the process using the following procedure.

To restart the repository upgrade and seed data import process

1 From the command line, navigate to the *SI EBSRVR/bin* directory (Windows) or *SI EBSRVR/bin* directory (UNIX).

2 Execute the following commands:

```
RSTAT390 /u UserId /p Password /c DatabaseName /d TableownerName /a Y /l rstat.log  
/T S_COLUMN
```

```
RSTAT390 /u UserId /p Password /c DatabaseName /d TableownerName /a Y /l rstat.log  
/T S_UK_ATTJOIN
```

3 Restart the Upgrade Wizard using one of the methods described in [“Starting the Siebel Upgrade Wizard” on page 132](#).

NOTE: To improve the repository upgrade performance, execute the REORG utility on the tables listed in [“About Reorganizing Tables Before the Repository Merge” on page 186](#) before performing the repository merge.

11 Creating the Siebel Staging Database

This chapter describes how to create the staging database used in the development and production database upgrades. It includes the following topics:

- [Process of Creating the Staging Database on page 137](#)
- [Required Tasks before Creating the Staging Database on page 137](#)
- [Creating the Staging Database Schema DDL Files on page 138](#)
- [Transferring the Staging DDL to the z/OS Host on page 139](#)
- [Preparing the z/OS Host Upgrade Environment and Creating the Staging Database on page 140](#)
- [Removing Interface Tables and Triggers on page 145](#)

Process of Creating the Staging Database

Upgrades: All upgrades.

Environments: All environments.

The process of creating the staging database involves performing the following tasks:

- 1 Perform the tasks described in [“Required Tasks before Creating the Staging Database”](#).
- 2 Create the DDL schema files that will be used to build the staging database by running the Database Configuration Wizard. See [“Creating the Staging Database Schema DDL Files” on page 138](#).
- 3 Transfer the staging DDL files generated in [Step 2](#) to the z/OS host where they are applied. See [“Transferring the Staging DDL to the z/OS Host” on page 139](#).
- 4 Set up the z/OS host environment for the staging and target database upgrades and create the staging database. See [“Preparing the z/OS Host Upgrade Environment and Creating the Staging Database” on page 140](#).
- 5 Drop interface tables and triggers. See [“Removing Interface Tables and Triggers” on page 145](#)

Required Tasks before Creating the Staging Database

Upgrades: All upgrades.

Environments: All environments.

Before you create the staging database, complete the following tasks:

- 1 Allocate an empty tableowner for the staging database.
- 2 Make sure you have a valid storage control file for the database to be upgraded. This contains the storage definitions of the existing database.

If you do not have a valid storage control file, create one by running the Database Configuration Wizard and selecting the Extract from Catalog option. Specify database details for your existing Siebel database (the database to be upgraded). For further information on this task, see *Implementing Siebel Business Applications on DB2 for z/OS*.

The Extract from Catalog option extracts the storage layout of the database you specify from the DB2 catalog and creates a new storage control file that contains a copy of the storage definitions of the database. This storage control file is used to generate the DDL to build the staging database schema.

- 3 Make sure the logical repository definitions and the physical schema definitions of the database to be upgraded match. You can do this by running the Database Configuration Wizard, and specifying the following options:
 - **Siebel Database Operation.** Run Database Utilities.
 - **Database Utility Selection.** Synchronize Schema Definition.
 - **Repository Synchronization Mechanism.** Generate DDL into Files.
 - **Storage Control File.** Specify the file you extracted in [Step 2 on page 138](#) above.

NOTE: It is recommended that you drop EIM tables before synchronizing the Siebel logical and physical schema to ensure that the synchronization process does not fail when processing EIM tables.

When the synchronize process completes, if the logical and physical schema database definitions match, no files are generated from the Synchronize Schema Definition process and no further action is required.

If the logical and physical schema database definitions do not match, a message is displayed stating that files have been generated. Click Yes to exit from the Wizard, then do the following:

- a Navigate to the directory you assigned as the Output Directory.
- b Transfer the schema.sql file to the z/OS host and ask your DBA to apply the DDL contained in the file.

For detailed information on running the Synchronize Schema Definition process, see *Implementing Siebel Business Applications on DB2 for z/OS*.

Creating the Staging Database Schema DDL Files

To create the staging database, you must generate schema DDL files from the target database. You then use these files and the storage control file you extracted from the target database to create the staging database. The following procedure describes how to generate the DDL files.

To generate the DDL files used to create the staging database

- 1 Run the Database Configuration Wizard. See [“About Running the Database Configuration Wizard Under Windows” on page 126](#) and [“About Running the Database Configuration Wizard Under UNIX” on page 129](#).
- 2 Specify the following options:
 - a **Upgrade Options:**
 - ❑ Development environment: Upgrade Siebel Database Schema (upgrep)
 - ❑ Production environment: Upgrade Siebel Database Schema (upgrep + upgphys)
 - b **Upgrade Process:** zSeries Staging of Files for Upgrade
 - c **Storage Control File:** Specify the name of the storage control file you extracted in [Step 2 on page 138](#)

NOTE: The DB2 subsystem you specify for the staging database must be different to the DB2 subsystem of the database to be upgraded (target database).

See [“Information Required by the Database Configuration Wizard” on page 122](#).

- 3 Launch the Siebel Upgrade Wizard. See [“Starting the Siebel Upgrade Wizard” on page 132](#).
 SQL commands are executed on your existing database to generate the staging database files. The Upgrade Wizard then stops (Pause #0).

 For a list of the files generated on the midtier by the Siebel Upgrade Wizard, see [Appendix C, “Siebel Upgrade Files.”](#)

Transferring the Staging DDL to the z/OS Host

When the Upgrade Wizard stops at Pause # 0, you must transfer the SQL and JCL templates and other staging DDL that has been generated on the midtier to the z/OS host where they can be executed. To do this, perform the following steps:

- 1 Navigate to the \DB2390\dbsrvr\dboutput\upgrade directory (Windows) or the /DB2390/dbsrvr/dboutput/upgrade directory (UNIX) and double-click the ftp_stg.bat file (Windows) or issue the following command (UNIX):


```
Ftp - i > ftp_stg.bat
```
- 2 Enter your TSO ID and password and press Enter.
 All the files required to create the staging database are transferred from the midtier to the z/OS host.
- 3 Review the ftp_stg.log file which is created in the upgrade directory and verify that all the files listed in the ftp_stg.txt file transferred successfully to z/OS staging datasets.

Preparing the z/OS Host Upgrade Environment and Creating the Staging Database

You must transfer the staging database and other upgrade files generated on the midtier by the Upgrade Wizard into datasets on the z/OS host. These files must then be processed and prepared for use. These files include the REXX code that contains all the panels and programs for running the upgrade processes, and files for creating the staging database.

To prepare your z/OS upgrade environment and create the staging database, complete the following procedures:

- [“Requirements for Upgrade Procedures on the z/OS Host” on page 140](#)
- [“Customizing the JCL UNIT Parameter Value” on page 140](#)
- [“Creating and Allocating the z/OS Setup Data Sets” on page 141](#)
- [“Preparing the Upgrade Environment and Building the Staging Database” on page 142](#)
- [“Removing Interface Tables and Triggers” on page 145.](#)

Requirements for Upgrade Procedures on the z/OS Host

The user who performs upgrade procedures on the z/OS host requires the following authorities and access:

- A thorough understanding of zSeries architecture, JCL, and TSO functions and navigation.
- A TSO account with the authorization to allocate/create datasets on the z/OS host using the high-level qualifier specified in the FTP script.
- Access to the DB2 target system.
- DB2 authorities to create DB2 objects and create DB2 VSAM datasets.
- Grant and Bind authority.
- DB2 Workload Manager refresh authority.

The default JOBCLASS=Q. Make sure you use the correct job class in the generated jobcards.

Customizing the JCL UNIT Parameter Value

If appropriate for your environment, you can amend the UNI T=SYSDA parameter setting for all of the JCL generated for the Siebel upgrade before you run any jobs on the mainframe. The UNI T=SYSDA parameter can be amended in the following data sets:

- JOB0
- Any of the VSTG000*n* or VSTG00*nn* data sets

To amend the UNIT parameter setting

- 1 Navigate to the data set you want to amend. Make a backup copy of the data set.
- 2 Go to Edit mode on the data set.
- 3 To change the UNIT=SYSDA parameter, enter the following command on the command line:

```
c sysda sgunit all
```

where *sgunit* is the unit name you want to specify.

- 4 Press PF3 to save your changes.

All the JCL generated by the data set will use the new value you specified.

If you want to revert to the default UNIT parameter settings, either restore the backup copy of the data set you made in [Step 1](#) or transfer the staging data sets from the midtier again.

NOTE: Any amendments you make to the UNIT parameter are not applied to data sets that are allocated using REXX routines.

Creating and Allocating the z/OS Setup Data Sets

The following procedure is used to create and allocate the z/OS setup data sets.

To define and allocate z/OS setup datasets

- 1 After you have successfully transferred the staging files generated by the Upgrade Wizard up to Pause #0, log on to the host/LPAR where the staging database is to be created. Verify that the Procedure parameter on the logon panel is set to DB2REL8.
- 2 To create the REXX, CNTL and JCLLIB datasets, navigate to the *DSNHLO.SIEBEL.JOB0* dataset.
- 3 Go to Edit mode on the dataset and submit the job using the JCL in the *DSNHLO.SIEBEL.JOB0* dataset.
- 4 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 5 Verify that the job ran successfully (RC=0) and that the following datasets were created:
 - *DSNHLO.SIEBEL.EXEC*
 - *DSNHLO.SIEBEL.JCLLIB*
 - *DSNHLO.SIEBEL.SP.SPCNTL*
- 6 Verify that the following PDS members contain information that you defined in the Database Configuration Wizard on the midtier:
 - *DSNHLO.SIEBEL.EXEC(@TBOSTG)* contains the value for the staging schema qualifier.
 - *DSNHLO.SIEBEL.EXEC(@BOTAR)* contains the value for the target schema qualifier.
- 7 To allocate the *DSNHLO.SIEBEL.EXEC* REXX exec library, enter 6 on the TSO command line to go to the TSO Command Line Processor panel.

- 8 Enter the following command and press Enter to allocate the CLIST/EXEC library:

```
al l oc f(sysexec) da(' DSNHLQ.SIEBEL.EXEC' ) shr reuse
```

NOTE: To make sure that only Siebel executables are referenced, this command disables all CLIST libraries allocated to the logged-in user.

No message appears when the library is allocated. To verify that the exec library is listed in the file allocations by name, enter the TSO command I SRDDN.

The EXEC library *DSNHLQ.SIEBEL.EXEC* is used to execute all REXX execs. The panel and programs are located in this library. Execute the SBLUPG80 member from the *DSNHLQ.SIEBEL.EXEC* REXX library to display the Siebel Upgrade Menu; you can use this menu to navigate through the steps in the upgrade process.

TIP: The TSO command allocation for the EXEC library must be reissued with each TSO logon. Add this allocation to your personal logon CLIST to avoid having to reallocate the *DSNHLQ.SIEBEL.EXEC* library with each new TSO logon.

Preparing the Upgrade Environment and Building the Staging Database

Once the files transferred from the midtier have been defined and allocated, you must set up the z/OS system environment variables, receive the files (uncompress the files), create JCL libraries, create the staging database, and assign jobname prefixes. The following procedure describes how to perform these tasks.

To set up the z/OS upgrade environment and build the staging database

- 1 Enter the following command and press Enter:

```
TSO SBLUPG80
```

The Siebel 8.0 Upgrade Main Menu appears. The panel ID is SBLUPG8P. You can find the panel ID in the bottom right corner of the screen.

TIP: To access Help on any screen, press PF1. To exit Help and return to the prior screen, press PF3.

- 2 On the Siebel Upgrade 8.0 Main Menu, select option 0: Environment Setup, and press Enter.
The Staging Database Environment Setup Menu appears. The panel ID is SBLSSETP. The options on this panel allow you to prepare files and set environment variables to create the staging database.
- 3 On the Staging Database Environment Setup Menu, select option 0: Set System Environment Variables.
The Staging System Variable Definitions panel appears. The panel ID is SBLSETVP.
- 4 Enter the following information and then press Enter:
 - **WLM Name.** Enter the DB2 WLM name, for example, DB28WLM.

- **WLM Load Library.** Enter the DB2 WLM load library name.
- **Extract DB2 Libraries:** Enter the DB2 load / runlib library names for the target subsystem. (These are the libraries where the DSN, DSNTEP2, and DSNTIAUL programs are located.)
- **Staging DB2 Libraries.** Enter the DB2 load / runlib library names for the staging subsystem.

NOTE: The libraries you enter must exist (that is, they must be cataloged).

5 Press Enter.

Messages appear indicating that the *DSNHLO* and DB2 load library information was written to individual PDS members in the *DSNHLO.SIEBEL.JCLLIB* library.

6 Press PF3 to return to the Staging Database Environment Setup menu.

7 Select option 1: Receive XMIT Datasets, and press Enter.

8 Submit the JCL in dataset *DSNHLO.SIEBEL.install.jcl* (SPXMITR)

9 After submitting the job, enter cancel on the command line or press PF3 to save changes.

This job receives XMIT format files. Three PDS datasets are allocated and populated with members. The three PDS dataset names are:

- *DSNHLO.SIEBEL.LOAD*
- *DSNHLO.SIEBEL.SP.SPDDL*
- *DSNHLO.SIEBEL.DBRMLIB*

10 Verify that the job ran successfully.

- a Review the output in SDSF or another job output facility. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- b Review the output to verify that all three datasets (and members) were received properly. You will get a message for each member successfully received for each of the three datasets.

From this point on, all jobs contain one of the following JCL INCLUDE members to check job step return codes. If condition codes are not met, the job abends (ends abnormally) with User 99. Acceptable return codes for each step of each job is controlled by the following three JCL test condition checks:

- JCLTEST requires RC<=4
- JCLTEST0 requires that RC=0
- JCLTEST8 requires that RC<=8

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see ["Restarting Upgrade Jobs That Fail" on page 183](#).

11 Press PF3 to return to the Staging Database Environment Setup menu.

- 12** Select option 2: Create JCL Libraries, and press Enter. This option builds and allocates the install JCL libraries.

You are placed in edit mode for dataset *DSNHLQ.SI EBEL.VSTG00nn* (*nn* varies according to your upgrade path).

NOTE: If you want to change the job card, do so at this time.

- 13** Run the job using the JCL in dataset *DSNHLQ.SI EBEL.VSTG00nn*. The install JCL libraries are built and allocated and path-specific panels are added to the *DSNHLQ.SI EBEL.EXEC* library.

- 14** Verify that the job ran successfully.

- a** Review the output in SDSF or another job output facility. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- b** Verify that this job allocated and populated dataset *DSNHLQ.SI EBEL.I NSTALL.JCL*. This job also adds path-specific panels to the *DSNHLQ.SI EBEL.EXEC* library.

- 15** After submitting the job, enter cancel on the command line or press PF3 to save changes.

- 16** On the Staging Database Environment Setup menu, select option 3: Create Siebel Logging Table (1). This option allows you to create and load the logging table for the staging jobs and to create the staging TMP_ADDR table.

You are placed in edit mode for dataset *DSNHLQ.SI EBEL.I NSTALL.JCL(LOADLOG1)*.

NOTE: If you want to change the job card, do so at this time.

- 17** Run the job using the JCL in dataset *DSNHLQ.SI EBEL.I NSTALL.JCL(LOADLOG1)*.

This job runs the DDL to create the TMP_SBLLOG_STG table and the TMP_ADDR table in the staging environment and loads an initial set of log entries for logging batch job execution.

NOTE: The TMPTBL_ADDR table is only built if you are upgrading from FINS 6.2.1 or FINS 7.0.4.

- 18** Verify that the job ran successfully:

- a** Review the output in SDSF or another job output facility. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- b** Verify that this job created the TMP_SBLLOG_STG table on the staging database and loaded an initial set of log entries for logging batch job execution.

- 19** After submitting the job, enter cancel on the command line or press PF3 to save changes.

- 20** On the Staging Database Environment Setup menu, select option 4: Create Staging Database (DDL), and press Enter to execute the DDL necessary to create the staging database.

You are in edit mode for dataset *DSNHLQ.SI EBEL.I NSTALL.JCL(STGDDL)*.

NOTE: If you want to change the job card, do so at this time.

- 21** Run the job using the JCL in dataset *DSNHLQ.SI EBEL.I NSTALL.JCL(STGDDL)*.

- 22** Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.

This job executes the staging database DDL to create an exact schema version of the target database (database to be upgraded) in a staging environment using the Host/LPAR, DB2 subsystem and tableowner values you specified when you ran the Database Configuration Wizard.

Only the schema exists; it does not need to be populated with data. All objects should contain the DDL syntax DEFINE NO. Index definitions include the DEFER YES syntax.

- 23** Press PF3 to return to the Environment Setup menu.
- 24** Select option 5: Assign Jobname Prefix/Parms, and press Enter to assign unique jobname prefixes to JCL upgrade jobs by job type.
- The Staging Jobname Prefix/Parm Definitions Jobcard parameters panel is displayed. The panel ID is SBLJXP.
- 25** You must enter a three-character job name prefix for all upgrade job types for items 1 through 20.
- The three-character prefix should be unique— this makes it easier to find your jobs in the queue— but it is only required for the non-unique index and obsolete index job prefixes (which cannot be the same). The remaining five characters of the job name (which do not appear and cannot be modified) are defined by Oracle and are unique across all upgrade jobs.
- 26** You can change the NOTIFY value from &SYSUI D to your TSO ID or leave it as &SYSUI D.
- NOTE:** If you want to remove the notify parameter from the job card, replace the symbolic parameter &sysui d with spaces.
- 27** Review the job card parameters and make any necessary changes. Verify that you are using the correct accounting, job class, and message class.
- 28** Press Enter after entering the job name prefix and parameter definitions. The JCL template files are updated. Messages indicate when each step is completed.

Removing Interface Tables and Triggers

Once you have build the staging database and prepared the upgrade environment on the z/OS host, the final task to complete before you generate the upgrade files is to remove triggers and EIM/Interface tables from the staging database. The following procedure describes this task.

To remove EIM/Interface tables and triggers from the staging database

- 1** On the Staging Database Environment Setup Menu, select option 6: PREGEN Processes, and press Enter.
- The Staging Database Pre_Gen Maint. Menu is displayed. The panel ID is SBLPREGP. Options on this panel allow you to generate EIM DROP statements, identify tables with CLOB columns, and generate trigger drops.

- 2 Select option 0: Generate EIM/Interface Table/Tablespace Drops, and press Enter.
- 3 Submit the JCL in the data set *DSNHLQ.SI EBEL.pregen.jcl* (PREGEN1). This job generates the EIM, Interface table, and table space DROP statements.
- 4 Verify that the job ran successfully.
Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 5 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 6 On the Staging Database Pre_Gen Maint. Menu, select option 1. Generate: - List of Tables with CLOB columns - Trigger Drops, and press Enter.
- 7 Submit the JCL in the data set *DSNHLQ.SI EBEL.pregen.jcl* (PREGEN2). A list of trigger drops and tables with CLOB columns is generated.
- 8 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 9 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 10 On the Staging Database Pre_Gen Maint. Menu, select option 2. Generate: Report of Address Duplicate Rowids, and press Enter.
NOTE: You need to select this option to run jobs to generate a report of duplicate address row IDs only if you are upgrading from SIA 6.2.1 or FINS 7.0.4.
- 11 Submit the JCL in the data set *DSNHLQ.SI EBEL.pregen.jcl* (ADROWIDS) to generate a report of duplicate address row IDs.
For further information on upgrading address data, see the chapter on preparing application data for upgrade in *Siebel Database Upgrade Guide*.
- 12 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 13 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 14 On the Staging Database Pre_Gen Maint. Menu, select option 3: Update Address Report generated log entry, and press Enter.
NOTE: You need to select this option to update address report log entries only if you are upgrading from SIA 6.2.1 or FINS 7.0.4.
- 15 Submit the JCL in the dataset *DSNHLQ.SI EBEL.pregen.jcl* (ADRLOGS) to update address report log entries.
- 16 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.

- 17 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 18 Press PF3 to return to the Staging Database Environment Setup Menu.
- 19 Select option 7: Drop Interface Tables, and press Enter.
- 20 Submit the JCL in the dataset *DSNHLQ.SIEBEL.install.jcl* (PRETIDRP) to drop EIM and Interface tables from the staging database.
- 21 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 22 After submitting the job, enter cancel on the command line or press PF3 to save changes.

This completes the process of creating a staging database. You can now generate the upgrade files.

12 Generating the Siebel Upgrade Files

This chapter describes how to generate the Siebel 8.0 upgrade files using the staging database. It includes the following topics:

- [About Generating the Upgrade Files on page 149](#)
- [Process of Generating the Upgrade Files on page 149](#)
- [Prerequisite Tasks for Generating the Upgrade Files on page 150](#)
- [Preparing the Additive Schema and JCL Files on the z/OS Host on page 151](#)
- [Applying the Additive Schema Changes to the Production Staging Database on page 155](#)
- [Preparing for Table Creation \(PRET\) on the Staging Database on page 156](#)
- [Preparing and Executing the Non-Additive Schema and JCL Files on the z/OS Host on page 158](#)
- [Processing the Index Schema File on page 162](#)
- [Building JCL Templates for the Target Database on page 164](#)

About Generating the Upgrade Files

Upgrades: All upgrades.

Environments: All environments.

After you have created the staging database, you generate the upgrade files by running the Upgrade Wizard against the staging database. The upgrade files are then transferred to the z/OS host, where the JCL preparation process is done. The JCL preparation process constructs all the JCL to run the preupgrade and upgrade (in-place) processes for both the staging and target databases.

The mid-tier file generation process can be performed ahead of the in-place target database upgrade provided no further changes are made to the target database schema.

NOTE: The procedures in this chapter use Siebel-Scheduled job execution, but you can also choose to use a third party job scheduler. For information on choosing a scheduler, see [“About Executing Jobs Using Siebel-Scheduled Mode or Vendor-Scheduled Mode” on page 77](#).

Process of Generating the Upgrade Files

Upgrades: All upgrades.

Environments: All environments.

When the staging database has been created, you run the Database Configuration Wizard to generate the upgrade files. This process involves the following tasks:

- 1 Complete the tasks described in [“Prerequisite Tasks for Generating the Upgrade Files”](#) on page 150.
- 2 Restart the upgrade process from the midtier to generate upgrade files. See [“Restarting the Siebel Upgrade Wizard After Pauses”](#) on page 134 for further information. The Upgrade Wizard stops at Pause #1.
- 3 Transfer the files generated up to Pause #1 to the z/OS host and apply them. See [“Preparing the Additive Schema and JCL Files on the z/OS Host”](#) on page 151.
- 4 (Production environment only) If you are upgrading a production database, you can optionally choose to apply the additive schema changes to the staging database. See [“Applying the Additive Schema Changes to the Production Staging Database”](#) on page 155 for further information.
- 5 Restart the Upgrade wizard as described in [“Restarting the Siebel Upgrade Wizard After Pauses”](#) on page 134. After generating further files, the Upgrade Wizard stops at Pause #2.
- 6 Transfer the files generated up to Pause #2 to the z/OS host and run the jobs against the staging database. See [“Preparing and Executing the Non-Additive Schema and JCL Files on the z/OS Host”](#) on page 158.
- 7 Return to the midtier and restart the Upgrade wizard as described in [“Restarting the Siebel Upgrade Wizard After Pauses”](#) on page 134. The Upgrade Wizard generates the SCINDEX.SQL file on the midtier and stops at Pause #3.
- 8 Transfer the SCINDEX.SQL file to the z/OS host and apply it to drop old schema indexes and create new 8.0 schema indexes. See [“Processing the Index Schema File”](#) on page 162.
- 9 Build the JCL Templates that will be used to upgrade the target database. See [“Building JCL Templates for the Target Database”](#) on page 164 for further information.

This completes the File Generation Process.

Prerequisite Tasks for Generating the Upgrade Files

Upgrades: All upgrades.

Environments: All environments.

Complete the following tasks before starting to generate the upgrade files:

- 1 Extract the storage layout of the Siebel staging database from the DB2 catalog and merge it with a Siebel 8.0 storage control file so as to preserve any customizations you have made to the database layout in the upgraded database. To do this, run the Database Configuration Wizard and select the Extract from Catalog and Merge with Template option.

For further information on this task, see [“Extracting the Storage Control File”](#) on page 91.

- 2 Validate the merged storage control file generated in [Step 1](#) against the staging database. For information on this task, see [“Validating the Extracted Storage Control File”](#) on page 95. If there are validation errors, you must correct them before proceeding with the upgrade.

Preparing the Additive Schema and JCL Files on the z/OS Host

Upgrades: All upgrades.

Environments: All environments.

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

When you restart the Upgrade Wizard after creating the staging database, it generates SQL and JCL templates for the file generation process. Specifically, the following files are generated:

- Additive upgrade files
- Pret files
- Data migration files

When the Upgrade Wizard stops at Pause # 1, you must transfer these files to datasets on the z/OS host. These datasets are then used to create the SQL and JCL templates used to perform the upgrade. These tasks are described in this topic.

Perform the following procedures:

- [“Transferring the Additive Schema, PRET, and Data Migration Files to the z/OS Host” on page 151](#)
- [“Preparing the z/OS Data Sets” on page 152](#)
- [“Specifying the Scheduling Mode” on page 153](#)
- [“Preparing the Additive Schema SQL and JCL Templates” on page 153](#)

Transferring the Additive Schema, PRET, and Data Migration Files to the z/OS Host

Use the following procedure to transfer the schema DDL files and the PRET and data migration files generated by the Upgrade Wizard on the midtier up to Pause #1 to the z/OS host.

NOTE: Before transferring the generated files to the z/OS host, edit them as required by relevant publications such as *Siebel Alerts*, *Technical Notes*, and *Siebel Release Notes on My Oracle Support*.

To transfer the files generated on the midtier

- 1 Navigate to the \DB2390\dbsrvr\dboutput\upgrade directory (Windows) or the /DB2390/dbsrvr/dboutput/upgrade directory (UNIX) and double-click the ftp_pause1.bat file (Windows) or issue the following command (UNIX):

```
Ftp - i > ftp_pause1.bat
```

- 2 Enter your TSO ID and password and press Enter.

All the Pause #1 files are transferred from the midtier to the z/OS host.

- 3 Review the ftp_pause1.log file which is created in the upgrade directory and verify that all the files listed in the ftp_pause1.txt file transferred successfully to z/OS staging data sets.

Preparing the z/OS Data Sets

When you have transferred the files generated by the Siebel Upgrade Wizard up to Pause #1 from the midtier to the z/OS host, allocate and populate the data sets used to execute the upgrade jobs.

To prepare the z/OS data sets

- 1 Go to the Siebel 8.0 Upgrade Main Menu by entering the following command and pressing Enter:

```
TSO SBLUPG80
```

The panel ID is SBLUPG8P. You can find the panel ID in the bottom right corner of the screen.

- 2 On the Siebel Upgrade 8.0 Main Menu, select option 1: File Generation, and press Enter. The Staging Database File Generation Menu panel is displayed. The panel ID is SBLSFGP.
- 3 Select option 0: Allocate PDS/SEQ Datasets, and press Enter.
- 4 Run the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(CREATEDS)* to allocate datasets.
- 5 Verify that the job ran successfully:
 - a Review the output in SDSF or another job output facility. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
 - b Verify that this job allocated datasets.
- 6 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 7 On the Staging Database File Generation Menu, select option 1: Translate SQL Scripts, and press Enter.

This option is used to run the JCL to translate SQL scripts. Language-specific data migration SQL types are translated for PRESCHM, PRODCFG, UPGIDSS and Gen Primary jobs.
- 8 Run the job using the JCL in data set *DSNHLQ.SIEBEL.SP.CNTL(RECxxxxx)* where *xxxxx* varies according to your upgrade path and language. If your primary language is ENU, the job runs in foreground mode. If your primary language is non-enu, the job runs in batch mode.
- 9 Verify that the job ran successfully:
 - a Review the output in SDSF or another job output facility. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
 - b Check the Bookmark field to verify that the job translated the SQL scripts.
- 10 On the Staging File Generation Menu, select option 2: Populate PDS/SEQ Datasets, and press Enter.
- 11 Run the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(UNPACK01)*.

12 Verify that the job ran successfully:

- a Review the output in SDSF or another job output facility. Verify that the RC=0. A RC=4 is acceptable if the dataset is empty.
- b Verify that the JCLTEST return code is FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF2061.
- c Verify that this job populates (unpacks) all PDS members into corresponding PDS datasets and sequential files. It is acceptable to have some empty datasets.

NOTE: PDS datasets are populated using IEBUPDTE; sequential files are populated using IEBCGENER.

Specifying the Scheduling Mode

Perform the following procedure to choose your preferred job scheduling method (Siebel scheduled or vendor scheduled).

To specify the scheduling mode

- 1 On the Staging Database File Generation Menu, select option 4: PRET Processes, and press Enter.
- 2 On the Staging Database PRET menu, select option 0: Add Jobcards & Siebel Logging, and press Enter.

If this is the first time you have selected this option, you are prompted to enter the type of scheduling for the PRET processes.

- 3 Enter either 1: Siebel Scheduled mode or 2: Vendor Scheduled mode. For information about choosing a scheduler, see [“About Executing Jobs Using Siebel-Scheduled Mode or Vendor-Scheduled Mode” on page 77](#).

The following procedures assume that you chose Siebel scheduled mode.

CAUTION: Choose your scheduling mode carefully. Once you select a scheduling mode for PRET, you cannot change it.

Preparing the Additive Schema SQL and JCL Templates

Perform the following procedure to prepare the additive schema files.

To prepare additive schema SQL and JCL templates

- 1 On the Staging Database File Generation Menu, select option 3: Process Additive Components, and press Enter.

The Staging Database Additive Schema Menu appears. The panel ID is SBLADMP.

- 2 Select option 0: Build Additive JCL/Schema, and press Enter,
Status messages are displayed as the additive components are built.

- 3** On the Staging Database Additive Schema Menu, select option 1: Populate JCL/Schema PDS Datasets, and press Enter.
- 4** Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(UNPKADD)*.
- 5** Verify that the job ran successfully:
 - a** Review the output in SDSF or another job output facility. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
 - b** Verify that this job populated the JCL and schema PDS data sets.
- 6** After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 7** On the Staging Database Additive Schema Menu, select option 2: Add Jobcards & Siebel Logging, and press Enter.
- 8** Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBJCL12)*.
- 9** Verify that the job ran successfully. The JCL jobcard counts are displayed by job type. The number of jobs that are build for the additive components varies according to your upgrade path. You must review the output very carefully in SDSF or another job output facility very carefully. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 10** After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 11** On the Staging Database Additive Schema Menu, select option 3: Load Siebel Log Entries (3), and press Enter. This option runs the job to load the Siebel logging table with additive entries for staging jobs.
- 12** Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(LOADLOG3)*.
- 13** Verify that the job ran successfully:
 - a** Review the output in SDSF or another job output facility very carefully. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
 - b** Verify that additional log entries have been loaded to the TMP_SBLLOG_TAR table on the staging database.
- 14** After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 15** On the Staging Database Additive Schema Menu, select option 4: Additive Schema Status, and press Enter.

The Additive Schema Status panel is displayed. The panel ID is SBLSADLP. A list of the additive schema members is displayed and their staging status, either PENDING or COMPLETED.
- 16** To view a member in browse mode, select the member by entering any non-blank character in the Opt column for the member (you can select more than one member to view on each panel).
- 17** Press Enter. If you selected more than one member to view, press PF3 to move to the next member.

Applying the Additive Schema Changes to the Production Staging Database

Upgrades: All upgrades.

Environments: Production test, production.

NOTE: For development environment upgrades, all schema changes are processed as non-additive, therefore this step is not required.

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

When you are upgrading the production staging database, you can optionally choose to apply all or a subset of the additive schema upgrade files ahead of the in-place upgrade. Only the additive upgrade files that you choose to apply to the staging database can be applied as additive changes to the target database before the in-place upgrade. Any additive changes you do not apply become part of the non-additive changes that are applied later. See [“Applying the Non-Additive Schema Changes” on page 160](#) for further information.

To apply additive schema changes

- 1 On the Staging Database Additive Schema Menu, select option 5: Select / Run Additive Changes, and press Enter.

The Staging Pending ADDITIVE Job Submission Menu is displayed. The panel ID is SBL5ADDP.

- 2 Choose whether you want to apply all, none, or a subset of the additive schema changes:
 - If you decide not to apply any of the additive schema changes, enter N for the Apply All (Y/N) prompt. Press PF3 until you return to the Staging Database File Generation menu.
 - To apply all of the additive schema changes, do the following:
 - Enter Y for the Apply All (Y/N) prompt, and press Enter.
 - Run the job using the JCL in data set *DSNHLQ.SIEBEL.INSTALL.JCL(ADDVSCHS)*.
 - To apply a subset of the additive schema changes, enter either J, S, or SUB in the Opt column of the appropriate member, and press Enter:
 - Typing SUB in the Opt column for a member automatically submits the JCL to apply the additive change in the member: *DSNHLQ.SIEBEL.ADD.JCL(member)*
 - Typing J in the Opt column for a member places you in edit mode in the JCL for the member: *DSNHLQ.SIEBEL.ADD.JCL(member)*
 - Typing S in the Opt column for a member places you in edit mode in the SQL for the member: *DSNHLQ.SIEBEL.ADD.SQL(member)*

NOTE: If you choose to selectively apply additive schema changes, bear in mind that some schema changes may require that other additive database, table space or table changes are applied first. In general, submit additive schema changes in database, table space, table hierarchical order.

- 3 Verify that the job ran successfully:

- a** Review the output in SDSF or another job output facility very carefully. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF2061.
- b** You can view the Siebel staging database job log to check whether an additive job completed successfully or not by navigating to the In-Place Upgrade Main Menu and selecting option 5: Staging Database Joblog. See [“Viewing the Siebel Job Log Status” on page 242](#) for further information.

Do not proceed until the additive schema changes you chose to apply complete successfully. Any additive changes you do not apply become part of the non-additive changes that are applied during the in-place database upgrade.

Preparing for Table Creation (PRET) on the Staging Database

Upgrades: All upgrades.

Environments: All environments.

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

The PRET (pre-table) jobs are run against the staging database tables at the beginning of the upgrade; a small number of tables are altered by the PRET step to prepare the database for upgrading. There are two types of PRET jobs: jobs that you submit manually and jobs that run automatically. The manual PRET jobs perform the following tasks (this list varies according to your upgrade path):

- Determine whether there are CLOBs in the schema
- Query catalog tables for clustering index information

The automatic jobs drop some indexes, rename table(s) and add columns to tables. If you are using Siebel Scheduling, the automatic jobs use the unique job name prefixes you specify in the following procedure.

Complete the following procedures to prepare the staging database to generate unload, load, and schema files:

To run the PRET jobs

- 1** On the In-Place Upgrade Main Menu, select option 1: File Generation, and press Enter.
- 2** The Staging Database File Generation Menu appears. The panel ID is SBLSFGP.
- 3** Select option 3: PRET Processes. The Staging Database PRET Menu appears. The panel ID is SBLPRETP.

- 4 To build job cards for the PRET and pretfins job types, on the Staging Database PRET menu, select option 0: Add Jobcards & Siebel Logging, and press Enter.

The pretfins jobcards are only built if you are upgrading a Siebel Industry Application. If you executed Household scripts on the midtier and transferred those files to the z/OS host, household jobcards are also built.

Messages are displayed indicating the job type (PRET, pretfins, or Household), the number of jobs built and the jobcards added.
- 5 To load Siebel logging tables for the staging database upgrade jobs, on the Staging Database PRET Menu, select option 1: Load Siebel Log Entries (2), and press Enter.
- 6 Submit the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(LOADLOG2)*.
- 7 After submitting the job, review the output in SDSF or another job output facility to verify that the job ran successfully:
 - a Verify that RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
 - b Verify that additional log entries have been loaded to the TMP_SBLLOG_TAR table on the staging database.
- 8 To run the staging PRET processes, on the Staging Database PRET menu, select option 2: PRET - PRET Jobs, and press Enter.
- 9 Submit the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(STGPRET)*.

This triggers the first PRET job, which then automatically submits the next PRET job in sequence. The number of PRET jobs that are automatically submitted varies according to your upgrade path.
- 10 After submitting the job, review the output in SDSF or another job output facility to verify that the job ran successfully.

Verify that RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I. All jobs must complete successfully before you proceed to the next step.
- 11 To run the staging pretfins jobs, on the Staging Database PRET menu, select option 3: PRETF - PRET-FINS Jobs, and press Enter.

NOTE: You only have to perform this step if you are performing an Siebel Industry Applications (SIA) upgrade.
- 12 Submit the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(STGPRETF)*. This triggers the first PRET-FINS job, which then automatically submits the next PRET job in sequence. The number of PRET jobs that are automatically submitted varies according to your upgrade path.
- 13 After submitting the job, review the output in SDSF or another job output facility to verify that the job ran successfully.

Verify that RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I. All jobs must complete successfully before you proceed to the next step.

- 14 You have now completed the additive schema staging file generation process. You must now return to the midtier and restart the Siebel Upgrade Wizard to generate the non-additive schema files, Temp table DDL files, and the load and unload files.

Preparing and Executing the Non-Additive Schema and JCL Files on the z/OS Host

Upgrades: All upgrades.

Environments: All environments.

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

When the Upgrade Wizard stops at Pause # 2, you must transfer the non-additive schema files, Temp table DDL files, and the load and unload files that were generated on the midtier to data sets on the z/OS host. You then prepare the files, and apply them to the staging database. Perform the following tasks:

- [“Transferring the Non-Additive Schema, Temp table, and Load and Unload files to the z/OS Host” on page 158](#)
- [“Preparing the Non-Additive Schema SQL and JCL Templates and Executing the DDL” on page 159](#)
- [“Applying the Non-Additive Schema Changes” on page 160](#)
- [“Creating the Data Migration Indexes” on page 161](#)

Transferring the Non-Additive Schema, Temp table, and Load and Unload files to the z/OS Host

Use the following procedure to transfer the non-additive schema DDL files, the Temp table DDL files, and the Load and Unload control cards generated by the Upgrade Wizard on the midtier up to Pause #2 to the z/OS host.

NOTE: Edit the generated files as required by Siebel Technical Notes, Siebel Alerts and *Siebel Release Notes* on My Oracle Support, or other publications before transferring them to the z/OS host.

To transfer the files generated on the midtier

- 1 Navigate to the \DB2390\dbsrvr\dboutput\upgrade directory (Windows) or the /DB2390/dbsrvr/dboutput/upgrade directory (UNIX) and double-click the ftp_pause2.bat file (Windows) or issue the following command (UNIX):

```
Ftp - i > ftp_pause2. bat
```

- 2 Enter your TSO ID and password and press Enter.

All the Pause #2 files are transferred from the midtier to the z/OS host.

- 3 Review the ftp_pause2.log file which is created in the upgrade directory and verify that all the files listed in the ftp_pause2.txt file transferred successfully to z/OS staging datasets.

Preparing the Non-Additive Schema SQL and JCL Templates and Executing the DDL

Perform the following procedure to prepare the non-additive schema files.

To prepare non-additive schema SQL and JCL templates

- 1 Go to the Siebel 8.0 In-Place Upgrade Main Menu by entering the following command and pressing Enter:

```
TSO SBLUPG80
```

The panel ID is SBLUPG8P.

- 2 On the Siebel In-Place Upgrade 8.0 Main Menu, select option 1: FILE GENERATION and press Enter. The Staging Database File Generation Menu is displayed. The panel ID is SBLSFGP.
- 3 On the Staging Database File Generation Menu, select option 5: Process Non-Additive Components, and press Enter:
 - If you are performing a development environment upgrade, the Staging Database Non-Additive Schema Menu is displayed. The panel ID is SBLSADNP.
 - If you are performing a production environment upgrade, you are first prompted to confirm that you have applied and completed all planned additive changes. Enter Y and press Enter and the Staging Database Non-Additive Schema Menu is displayed.
- 4 On the Staging Database Non-Additive Schema Menu, select option 0: Apply JCL/Schema Template Parm, and press Enter.

Messages are displayed indicating that changes are being applied to the Unload and Load datasets.
- 5 On the Staging Database Non-Additive Schema Menu, select option 1: Populate JCL/Schema PDS Datasets, and press Enter.
- 6 Submit the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(UNPACK02)*.

This job unpacks the JCL and schema files into members in their corresponding PDS data set.
- 7 Verify that the job ran successfully.

Review the output in SDSF or another job output facility. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 8 On the Staging Database Non-Additive Schema Menu, select option 2: Customize CLOB Unload/Load cards, and press Enter.

If the target database does not have CLOB columns on any tables, one of two message types appears:

- A message box that indicates that this step is not required.
- UTL0033: CLOB CUSTOMIZATION NOT REQUIRED FOR *upgtype* UPGRADE PATH.

If CLOBs exist on the target system, a series of messages appear. Read them carefully. This option will run in foreground and then place you in edit mode in the following dataset:

```
DSNHLO.SIEBEL.INSTALL.JCL(CLOBCOPY)
```

- 9 Submit the job and review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 10 On the Staging Database Non-Additive Schema Menu, select option 3: Add Jobcards & Siebel Logging to Unloads/Loads and Data Migration Jobs, and press Enter.

A message appears, asking you to confirm that the *DSNHLO.SIEBEL.PROC(ISPBAT)* (ISPF batch proc) is correctly configured.

You can choose to modify the Unload/Load and data migration jobs in either TSO foreground mode or batch mode. It is recommended that you perform the procedure in batch mode.

In foreground mode, messages are displayed on the screen as the JCL-Prep progresses. This mode will lock up your session until the option is complete. Each option can take an extended period of time (more than thirty minutes), depending on the user's dispatching priority.

NOTE: It is recommended that you perform the procedure in batch mode. Before doing so, make sure you modify the ISPF batch procedure, *dsnhlg.siebel.proc(ispbat)*, to your installation standards. If you perform the procedure in foreground mode, make sure your logon region size is at least 7092.

- 11 Enter Y to confirm that the ISPF batch proc is correctly configured, and press Enter.
- 12 Press Enter again and you are placed in edit mode in the following dataset:

```
DSNHLO.SIEBEL.INSTALL.JCL(SBLJCL23)
```

- 13 Submit the job and review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.

NOTE: If you chose to perform the upgrade using Siebel-Scheduled mode, the data migration JCL includes a jobstep that auto-submits dependent jobs in the data migration flow until all jobs are completed for that data migration job type.

Applying the Non-Additive Schema Changes

Perform the following procedure to apply the non-additive schema changes to the staging database.

If you are performing a development database upgrade, or if you are performing a production database upgrade but chose not to apply any additive changes in advance, all the Siebel 8.0 schema upgrade changes are now applied to the staging database.

To apply non-additive schema changes

- 1 On the Staging Database Non-Additive Schema Menu, select option 4: Apply Non-Additive Changes, and press Enter.
- 2 Run the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(SCHEMAS)*.
- 3 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 4 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Creating the Data Migration Indexes

Perform the following procedure to create temporary tables and indexes for the data migration scripts and to rebuild indexes.

To create data migration indexes

- 1 On the Staging Database Additive Schema Menu, select option 5: Build Data Migration Index Statements, and press Enter.

Messages are displayed on screen indicating that data migration CREATE INDEX and DROP INDEX statements are being generated. The messages you receive depends on your upgrade path.
- 2 On the Staging Database Non-Additive Schema Menu, select option 6: Create Data Migration Temp Table/Indexes, and press Enter.

This job creates COMMON temp tables and indexes used by upgrade data migration SQL scripts.
- 3 Submit the JCL in one of the following datasets:
 - For Siebel Industry Application (SIA) upgrades, use *DSNHLQ.SIEBEL.INSTALL.JCL(DMXSIA)*
 - For Siebel Business Application upgrades, use *DSNHLQ.SIEBEL.INSTALL.JCL(DMXHOR)*.
- 4 Verify that the job ran successfully:
 - a Review the output in SDSF or another job output facility very carefully. Verify that the RC=0, RC=4, or RC=8 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
 - b Verify that additional log entries have been loaded to the TMP_SBLLOG_TAR table on the staging database.
- 5 After submitting the job, enter cancel on the command line or press PF3 to save changes.

- 6 On the Staging Database Non-Additive Schema Menu, select option 7: Generate Index Rebuilds, and press Enter.

A message appears, asking you to confirm that the *DSNHLQ.SIEBEL.PROC(ISPBAT)* (ISPF batch proc) is correctly configured.

You can choose to generate index rebuilds in either TSO foreground mode or batch mode. It is recommended that you perform the procedure in batch mode.

In foreground mode, messages are displayed on the screen as the JCL-Prep progresses. This mode will lock up your session until the option is complete. Each option can take an extended period of time (more than thirty minutes), depending on the user's dispatching priority.

NOTE: It is recommended that you perform the procedure in batch mode. Before doing so, make sure you modify the ISPF batch procedure, *dsnhlq.siebel.proc(ispmat)*, to your installation standards. If you perform the procedure in foreground mode, make sure your logon region size is at least 7092.

- 7 Enter Y to confirm that the ISPF batch proc is correctly configured, and press Enter.
- 8 When the following message appears, specify the number of indexes to be included in each rebuild job, and press Enter:

NUMBER OF INDEXES PER REBUILD JOB.

The maximum number of indexes that can be included in a job is 10. It is recommended that you specify 3.

CAUTION: Consider your objective before choosing a maximum number of indexes for each job. Increasing this number results in fewer jobs but requires more memory and sort work. Reducing this number results in more jobs, which reduces resource requirements but causes fewer indexes to be built in parallel.

- 9 Press Enter, and you are placed in Edit mode in the *DSNHLQ.SIEBEL.INSTALL(SBLJCL24)* dataset.
- 10 Submit the job and verify that the job ran successfully.

Review the output in SDSF or another job output facility very carefully. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF2061.

- 11 Press PF3 twice to return to the Staging Database File Generation Menu.

Processing the Index Schema File

Upgrades: All upgrades.

Environments: All environments.

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

When the Upgrade Wizard stops at Pause # 3, you must transfer the SCINDEX.SQL file to the z/OS host and apply it to prepare the 8.0 index DDL and to build DROP statements for old schema indexes.

Perform the following procedures:

- [“Transferring the SCINDEX.SQL File to the z/OS Host” on page 163](#)
- [“Restructuring the Index DDL” on page 163](#)

Transferring the SCINDEX.SQL File to the z/OS Host

Use the following procedure to transfer the SCINDEX.SQL file generated by the Upgrade Wizard on the midtier up to Pause #3 to the z/OS host.

NOTE: Edit the generated files as required by Siebel Technical Notes, Siebel Alerts and *Siebel Release Notes* on My Oracle Support, or other publications before transferring them to the z/OS host.

To transfer the file generated on the midtier

- 1 Navigate to the \DB2390\dbsrvr\dboutput\upgrade directory (Windows) or the /DB2390/dbsrvr/dboutput/upgrade directory (UNIX) and double-click the ftp_pause3.bat file (Windows) or issue the following command (UNIX):

```
Ftp - i > ftp_pause3. bat
```

- 2 Enter your TSO ID and password and press Enter.
The SCINDEX.SQL file is transferred from the midtier to the z/OS host.
- 3 Review the ftp_pause3.log file which is created in the upgrade directory and verify that the SCINDEX file transferred successfully to z/OS staging datasets.

Restructuring the Index DDL

Perform the following procedure to prepare the Index JCL.

To restructure the index DDL

- 1 Go to the Siebel In-Place Upgrade 8.0 Main Menu by entering the following command, and pressing Enter:

```
TSO SBLUPG80
```

- 2 On the Siebel In-Place Upgrade 8.0 Main Menu, select option 1: File Generation, and press Enter. The Staging Database File Generation Menu is displayed.
- 3 On the Staging Database File Generation Menu, select option 6: Process SCINDEX Index Components - Restructure Indexes, and press Enter.

A message appears, asking you to confirm that the *DSNHLQ.SIEBEL.PROC(ISPBAT)* (ISPF batch proc) is correctly configured.

You can choose to generate index rebuilds in either TSO foreground mode or batch mode. It is recommended that you perform the procedure in batch mode.

- 4 Enter Y to confirm that the ISPF batch proc is correctly configured, and press Enter.

- 5 When the following message appears, specify the number of indexes to be included in each rebuild job, and press Enter:

```
NUMBER OF INDEXES PER REBUILD JOB.
```

The maximum number of indexes that can be included in a job is 10. It is recommended that you specify 3.

CAUTION: Consider your objective before choosing a maximum number of indexes for each job. Increasing this number results in fewer jobs but requires more memory and sort work. Reducing this number results in more jobs—reducing resource requirements but building fewer indexes in parallel.

- 6 Press Enter, and you are placed in Edit mode in the *DSNHLQ.SIEBEL.INSTALL(SBLJCL31)* dataset.
- 7 Submit the job and verify that the job ran successfully.
Review the output in SDSF or another job output facility very carefully. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF2061.
- 8 Press PF3 twice to return to the Staging Database File Generation Menu.

Building JCL Templates for the Target Database

Perform the following procedure to build the JCL templates for the target database.

To build JCL templates for the target database

- 1 Go to the Siebel In-Place Upgrade 8.0 Main Menu by entering the following command, and pressing Enter:

```
TSO SBLUPG80
```
- 2 Select option 1: File Generation, and press Enter. The Staging Database File Generation Menu is displayed.
- 3 Select option 7: Build Target Database JCL Templates, and press Enter.
This option applies target database LPAR, Tableowner and STORGROUP values to the JCL templates that were created to run the preupgrade and upgrade (in-place) processes for the staging database so they are appropriate for the target database upgrade.
- 4 Submit the JCL in one of the following data sets:
 - For Siebel Industry Application (SIA) upgrades, use *DSNHLQ.SIEBEL.INSTALL.JCL(ALLSIAS)*
 - For Siebel Business Application upgrades, use *DSNHLQ.SIEBEL.INSTALL.JCL(ALLHORS)*
- 5 Verify that the job ran successfully. Review the output in SDSF or another job output facility very carefully. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF2061.

- 6 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 7 Press PF3 twice to return to the Staging Database File Generation Menu.

When you successfully complete the target database file generation process, the File Generation option (1) on the Siebel In-Place Upgrade 8.0 main Menu is no longer available. You are now ready to start the target database pre-upgrade and upgrade processes.

13 Upgrading the Target Database

This chapter describes how to upgrade your database to Siebel 8.0 using the upgrade files you generated against the staging database. It includes the following topics:

- [Process of Upgrading the Target Database on page 167](#)
- [Creating and Loading Siebel Log Tables on page 168](#)
- [Applying Additive Upgrade Changes to the Target Database on page 168](#)
- [Performing the In-Place Target Database Upgrade on page 170](#)
- [Restarting Upgrade Jobs That Fail on page 183](#)

Process of Upgrading the Target Database

Upgrades: All upgrades.

Environments: All environments.

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

After you have generated upgrade files against the staging database and prepared the JCL used to run the upgrade processes, you are ready to perform the target database upgrade. This involves the following steps:

- 1 [“Creating and Loading Siebel Log Tables” on page 168](#)
- 2 (Production upgrades only) [“Applying Additive Upgrade Changes to the Target Database” on page 168](#)
- 3 [“Performing the In-Place Target Database Upgrade” on page 170](#)

For production database upgrades, the additive changes applied to the staging database are applied to the production target database as part of the preupgrade process. The remaining non-additive, schema, unload, load and data migration processes are then applied to the target database as part of the critical path in-place upgrade.

For development database upgrades, upgrade changes are applied in one step to the target database during the in-place upgrade.

NOTE: The procedures in this chapter use Siebel-Scheduled job execution but you can also choose to use a third party job scheduler. For information on choosing a scheduler, see [“About Executing Jobs Using Siebel-Scheduled Mode or Vendor-Scheduled Mode” on page 77.](#)

Creating and Loading Siebel Log Tables

Upgrades: All upgrades.

Environments: All environments.

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

Perform the following procedure to create Siebel log tables and load them on the target database before you begin the target database upgrade.

To load the target log table

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:

```
TSO SBLUPG80
```

- 2 Select option 2: Target Database Processes - Pre-Upgrade, and press Enter.
- 3 Select option 0: Create & Load Target Siebel Log, and press Enter.
- 4 Run the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(LoadTar)*.
This loads the target Siebel log table using the *DSNHLQ.SIEBEL.JOBLOG.LoadFile*.
- 5 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 6 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Applying Additive Upgrade Changes to the Target Database

Upgrades: All upgrades.

Environments: Production test, production.

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

NOTE: For development database upgrades, all schema changes are processed as non-additive so this step does not apply.

Before you can perform the in-place target database upgrade, you must first apply all of the additive schema changes that you previously applied to the production staging database to the production target database. See [“Applying the Additive Schema Changes to the Production Staging Database” on page 155](#) for further information.

When applying additive changes, you can either:

- Apply one, or a few of the additive changes to the target database during one or more sessions.
- Apply all of the additive changes as one job, *provided that you applied all of the additive changes to the staging database.*

Both methods of applying additive changes are described in this section.

Applying Additive Changes Individually

You can apply individual additive changes to the target database using the following procedure.

To apply selected additive changes to the target production database

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:

```
TSO SBLUPG80
```

- 2 Select option 2: Target Database Processes - Pre-Upgrade, and press Enter.
- 3 Select option 1: Schedule/Run PENDING Jobs (Target), and press Enter.

The Target Additive PENDING Job Status Menu is displayed. The panel ID is SBL SADTP.

This panel lists all the pending additive jobs, that is, all the jobs that were applied as additive jobs to the staging database and which have not been applied to the target database.

- 4 To apply or view the additive schema changes, enter either J, S, or SUB in the Opt column of the appropriate member, and press Enter:

- Typing SUB in the Opt column for a member automatically submits the JCL to apply the additive change in the member: *DSNHLQ. SI EBEL. ADDTAR. JCL(member)*

To apply all the additive changes together, follow the procedure in [“Applying the Additive Changes in One Job” on page 170](#).

- Typing J in the Opt column for a member places you in edit mode in the JCL for the member: *DSNHLQ. SI EBEL. ADDTAR. JCL(member)*

- Typing S in the Opt column for a member places you in edit mode in the SQL for the member: *DSNHLQ. SI EBEL. ADDTAR. SQL(member)*

NOTE: You can selectively submit additive schema changes according to the amount of time you have available. You must bear in mind, however, that some schema changes may require that other additive database, table space or table changes are applied first. Review the additive changes before submitting them and, in general, submit additive schema changes in database, table space, table hierarchical order.

- 5 After submitting the JCL to apply the additive change in a member, verify that the job ran successfully:
 - a Review the output in SDSF or another job output facility very carefully. Verify that the RC=0, RC=4, or RC=8 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF2061.
 - b You can view the Siebel target database job log to check whether an additive job completed successfully or not by navigating to the In-Place Upgrade Main Menu and selecting option 6: Target Database Joblog. See [“Viewing the Siebel Job Log Status” on page 242](#) for further information.

Do not proceed until all the pending additive schema jobs are run successfully.

Applying the Additive Changes in One Job

You can apply all the additive schema upgrade changes to the target database using the JCL in the data set *DSNHLQ.SIEBEL.INSTALL.JCL(ADDVSCHT)* provided you also applied all of the additive changes to the staging database using the JCL in the *DSNHLQ.SIEBEL.INSTALL.JCL(ADDVSCHS)* data set.

CAUTION: If you applied only a subset of the additive schema upgrade changes to the staging database, you must not apply the additive schema upgrade changes to the target database using the JCL in the data set *DSNHLQ.SIEBEL.INSTALL.JCL(ADDVSCHS)*. If you do, you will corrupt your upgrade.

To apply all additive changes to the target production database

- 1 Navigate to the ISPF Primary Option Menu, and select option 2: Edit.
- 2 Specify *DSNHLQ.SIEBEL.INSTALL.JCL(ADDVSCHT)* as the name of the data set member you want to edit on the Edit Entry Panel.
- 3 Submit the job in the *ADDVSCHT* member.

All of the pending additive schema changes are automatically submitted and applied to the target database.

Performing the In-Place Target Database Upgrade

Upgrades: All upgrades.

Environments: All environments.

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

Before performing the in-place target database upgrade, make sure you have completed the appropriate pre-upgrade tasks. See [“Creating and Loading Siebel Log Tables” on page 168](#) and [“Applying Additive Upgrade Changes to the Target Database” on page 168](#) for further information.

To execute the in-place target database upgrade, perform the following tasks in the sequence shown:

- [“Preparing the Target Database for the Upgrade” on page 171](#)
- [“Running the PRET Jobs For the Target Database” on page 172](#)
- [“Applying Non-Additive Schema Upgrade Changes to the Target Database” on page 172](#)
- [“Creating and Deploying Stored Procedures on the Target Database” on page 176](#)
- [“Migrating Data On the Target Database” on page 178](#)
- [“Creating 8.0 Schema Indexes” on page 180](#)
- [“Migrating the Gen Primary 8.0 SQL to Update Data in Target Tables” on page 181](#)
- [“Generating RUNSTATS Jobs” on page 182](#)

Accessing the Target Database In-Place Upgrade Menu

The Target Database In-Place Upgrade Menu provides options that allow you to perform all of these tasks. Perform the following procedure to access this menu.

To access the Target Database In-Place Upgrade Menu

- 1 Access the Siebel In-Place Upgrade 8.0 Main Menu by entering the following command, and pressing Enter:

```
TSO SBLUPG80
```

- 2 Select option 3: Target Database Processes - Upgrade, and press Enter.

The Target Database In-Place Upgrade Menu is displayed. The panel ID is SBLINPP.

Preparing the Target Database for the Upgrade

Perform the following procedure to drop interface tables, triggers, and stored procedures from the target database to prepare for the upgrade.

To drop interface tables, triggers and stored procedures

- 1 On the Target Database In-Place Upgrade Menu, select option 0: Drop Interface Tables, and press Enter.

This option runs the job to drop interface tables from the target database.

- 2 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(INFDRPJT)*.
- 3 Verify that the job ran successfully. Review the output in SDSF or another job output facility very carefully. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 4 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 5 On the Target Database In-Place Upgrade Menu, select option 1: Drop Triggers, and press Enter.
- 6 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(TRGDRPJT)* to drop triggers from the target database.
- 7 Verify that the job ran successfully. Review the output in SDSF or another job output facility very carefully. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 8 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 9 On the Target Database In-Place Upgrade Menu, select option 2: Drop SPs and Functions, and press Enter.
- 10 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SPFDRPT)*.

- 11 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0, RC=4, or RC=8 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 12 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Running the PRET Jobs For the Target Database

Perform the following procedure to run the PRET jobs to prepare the target database for table creation during the in-place upgrade.

To run the PRET jobs

- 1 On the Target Database In-Place Upgrade Menu, select option 3: Pret, and press Enter.
- 2 Run the job using the JCL in dataset *DSNHLQ.SI EBEL. I NSTALL. JCL(SUBPRET)*.
This triggers the first PRET job, which then automatically submits the next PRET job in sequence. The number of PRET jobs that are automatically submitted varies according to your upgrade path.
- 3 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 4 After submitting the job, enter cancel on the command line or press PF3 to save changes.
All jobs must complete successfully before you proceed to the next step.
- 5 On the Target Database In-Place Upgrade Menu, select option 4: PretFINS, and press Enter.
- 6 Run the job using the JCL in dataset *DSNHLQ.SI EBEL. I NSTALL. JCL(SUBPRETF)*.
This triggers the first PRET job, which then automatically submits the next PRET job in sequence. The number of PRET jobs that are automatically submitted varies according to your upgrade path.
NOTE: You only have to perform this step if you are performing an Siebel Industry Applications upgrade.
- 7 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 8 After submitting the job, enter cancel on the command line or press PF3 to save changes.
All PretFINS jobs must complete successfully before you proceed to the next step.

Applying Non-Additive Schema Upgrade Changes to the Target Database

Perform the following procedures to apply the non-additive schema upgrade changes to the target database. Perform the procedures in the sequence in which they are listed.

- “Removing Target Database Views” on page 173
- “Running Unload Jobs on the Target Database” on page 173
- “Creating the Schema on the Target Database” on page 174
- “Loading the Schema on the Target Database” on page 174
- “Executing 8.0 Index DDL and Rebuilding Indexes” on page 175
- “Creating and Rebuilding Obsolete Indexes” on page 176

Removing Target Database Views

Perform the following procedure to remove views from the target database.

To drop views from the target database

- 1 On the Target Database In-Place Upgrade Menu, select option 5: Non-Additive Processes, and press Enter.

The following message appears:

Before proceeding, make sure ALL “Pret-FINS” jobs completed. Re-enter option 5 to continue.

- 2 Reselect option 5: Non-Additive Processes, and press Enter.
The Target Database “Non-Additive” Processes Menu appears. The panel ID is SBLNONP.
- 3 Select option 0: Drop Views, and press Enter.
- 4 Run the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(SUBDRPV)*.
- 5 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 6 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Running Unload Jobs on the Target Database

Perform the following procedure to run unload jobs on the target database.

To run Unload jobs on the target database

- 1 On the Target Database “Non-Additive” Processes Menu, select option 1: Unload Schema, and press Enter.
- 2 Run the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(SUBUNLD)*.

This job submits all the Unload jobs to run in parallel. The number of Unload jobs run varies according to your upgrade path.

- 3 Verify that each Unload job ran successfully.

For each job, review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.

A RC=4 may be returned if a table is empty.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Restarting Upgrade Jobs That Fail” on page 183](#).

- 4 After submitting the job, enter cancel on the command line or press PF3 to save changes.

- 5 View the job status log.

You can view the job status log by completing the procedure described in [“Running SQL in Siebel Logs” on page 243](#).

Creating the Schema on the Target Database

Perform the following procedure to create the Siebel 8.0 schema on the target database.

To create the 8.0 schema on the target database

- 1 On the Target Database “Non-Additive” Processes Menu, select option 2: Create Schema, and press Enter.

The following message appears:

Before proceeding, make sure ALL “UNLOAD” jobs completed. Re-enter option 2 to continue.

- 2 Reselect option 2: Create Schema, and press Enter.
- 3 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SCHEMAT)*.
- 4 Verify that each Unload job ran successfully.

For each job, review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.

- 5 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Loading the Schema on the Target Database

Perform the following procedure to run Load jobs on the target database.

To run the Load jobs on the target database

- 1 On the Target Database “Non-Additive” Processes Menu, select option 3: Load/Re-Load Schema, and press Enter.

- 2 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBLOAD)*.

This submits all the Load jobs to run in parallel. The number of upgrade Load jobs varies by upgrade path.

NOTE: If your database layout allows multiple tables in each table space, Loads for the same table space are stacked in the input job queue using the same job name to guarantee serialized loading.

- 3 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 4 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Executing 8.0 Index DDL and Rebuilding Indexes

Perform the following procedure to execute the Siebel 8.0 index DDL and rebuild indexes on the target database.

To execute the index DDL and rebuild 8.0 indexes on the target database

- 1 On the Target Database “Non-Additive” Processes Menu, select option 4: Create 8.0 Restructured Indexes (DDL), and press Enter.

The following message appears:

Before proceeding, make sure ALL “LOAD” jobs completed. Re-enter option 4 to continue.

- 2 Reselect option 4: Create 8.0 Restructured Indexes (DDL), and press Enter.
- 3 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SCHEMAT2)*.
This job runs DDL Create Index statements to build non-unique indexes for the old schema.
- 4 Verify that the job ran successful. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 5 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 6 On the Target Database “Non-Additive” Processes Menu, select option 5: Rebuild 8.0 Indexes, and press Enter.
- 7 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBNONIX)*.
This job automatically submits all non-unique 8.0 index rebuild jobs.
- 8 Verify that the job ran successful. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Restarting Upgrade Jobs That Fail” on page 183](#).

Creating and Rebuilding Obsolete Indexes

If you choose, you can create and rebuild the old schema obsolete indexes. This step is optional.

Perform the following procedure to create and rebuild obsolete indexes on the target database.

To create and rebuild obsolete indexes

- 1 On the Target Database “Non-Additive” Processes Menu, select option 6: Create Obsolete Indexes, and press Enter.
- 2 Run the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(SCHEMATO)*.
This job automatically builds the old-schema obsolete indexes.
- 3 Verify that the job ran successful. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 4 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 5 On the Target Database “Non-Additive” Processes Menu, select option 7:Rebuild Obsolete Indexes, and press Enter.
- 6 Run the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(SUB0BSIX)*.
This job automatically rebuilds the old-schema obsolete indexes.
- 7 Verify that the job ran successful. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 8 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Creating and Deploying Stored Procedures on the Target Database

Perform the following task to install stored procedures and functions on the target database and to verify that they deployed correctly.

To install and verify stored procedures

- 1 On the Target Database In-Place Upgrade Menu, select option 6: Create SPs & Functions (Bind), and press Enter.
A message appears informing you that you should use the instructions in the @README member to install the stored procedures and functions.
- 2 Enter Y to continue, and press Enter.
You are placed in edit mode for PDS dataset *DSNHLQ.SIEBEL.SP.CNTL*.
- 3 Select the *DSNHLQ.SIEBEL.SP.CNTL(@README)* PDS member.

- 4 Review the instructions in the @README file.
- 5 Select the IEBCOPY PDS member and submit the JCL in the dataset *DSNHLO.SIEBEL.SP.CNTL(IEBCOPY)*.

This job moves the stored procedure load modules into the WLMSPAS (this is the WLM load library you specified in “Preparing the Upgrade Environment and Building the Staging Database” on page 142).

- 6 Verify that the job ran successfully, RC=0.

The JCLTEST return code should be FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF2061.

- 7 After submitting the job, enter cancel on the command line or press PF3 to save changes.

- 8 Select the PDS member for your upgrade path:

- Use BINDHOR if you are upgrading a Siebel 7.0.4, 7.5.2, or 7.7 application.
- Use BINDSIA if you are upgrading a SIA 6.2.1, SIA/FINS/SIS 7.0.4, SIA 7.5.2, or SIA 7.7 application.

This places you in edit mode for PDS dataset member *DSNHLO.SIEBEL.SP.CNTL(BINDSIA)* or *DSNHLO.SIEBEL.SP.CNTL(BINDHOR)*.

This job binds the stored procedure packages.

NOTE: A DROP procedure job is included (*DSNHLO.SIEBEL.SP.CNTL(DRPSIA)* or *DSNHLO.SIEBEL.SP.CNTL(DRPHOR)*). If the procedure already exists, you must drop it before running the bind job.

- 9 Verify that the job ran successfully, RC=0 or RC=4.

The JCLTEST return code should be FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF2061.

- 10 After submitting the job, enter cancel on the command line or press PF3 to save changes.

- 11 Select the WLMREFSH PDS member and submit the JCL in the dataset *DSNHLO.SIEBEL.SP.CNTL(WLMREFSH)*.

This job refreshes the DB2 WLM environment.

- 12 Verify that the job ran successfully, RC=0.

The JCLTEST return code should be FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF2061.

- 13 Press PF3 to return to the Target Database In-Place Upgrade Menu.

- 14 Select option 7: Verify SP and Function Deployment, and press Enter.

This places you in edit mode for one of the following PDS datasets and members:

- For Siebel Business Application upgrades: *DSNHLO.SIEBEL.SP.CNTL(SPVHOR)*
- For Siebel Industry Application upgrades: *DSNHLO.SIEBEL.SP.CNTL(SPVSIA)*

- 15 Run the JCL in the appropriate dataset for your upgrade. The JCL in the SPVSIA and SPVHOR members executes each stored procedure against the target database after the new schema has been created and data has been loaded onto the target. This process verifies that the stored procedures have been installed and can be executed.
- 16 Verify that the job ran successfully, RC=0.
The JCLTEST return code should be FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF2061.
- 17 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Migrating Data On the Target Database

Perform the tasks described in this topic to migrate pre 8.0 Siebel data to version 8.0 data. Generally, this involves inserting or updating values in the target tables but new indexes may also be created and rebuilt.

There are optional data migration scripts for eChannel, Household and Product Configurator data and for Siebel Financial Services (FINS) applications. The scripts you need to apply depends on the applications you have implemented and your upgrade path. Review [Table 12](#) to determine the scripts that apply for your upgrade.

Table 12. Data Migration Scripts to Run

Data Migration Script	Applicable Upgrade Path
eChannel	Upgrades from Release SIA 6.2.1 if eChannel is implemented
Household	All Siebel Industry application (SIA) upgrades
Household - FINS	All Siebel Industry application (SIA) upgrades
Preschm	All upgrade paths
Address	Upgrades from Release SIA 6.2.1 and FINS 7.0.4
Preschm - FINS	All Siebel Industry application (SIA) upgrades
Product Config	Upgrades from Release SIA 6.2.1
Product Config - FINS	Upgrades from Release SIA 6.2.1 and FINS 7.0.4
UpGISS	All upgrade paths

Follow the procedure below to run each of the data migration scripts.

To run the data migration scripts

- 1 On the Target Database In-Place Upgrade Menu, select option 8: Data Migration Processes, and press Enter.
The Target Database Data Migration Processes Menu appears. The panel ID is SBLDMP.

- 2 Select the appropriate option for the data migration script you want to run, for example, select option 3: Preschm, and press Enter, to run the preschm scripts.
- 3 You are placed in edit mode on one of the following datasets, depending on the option you select:
 - eChannel: `DSNHLO.SI EBEL. I NSTALL. JCL(ECHANNEL)`
 - Household: `DSNHLO.SI EBEL. I NSTALL. JCL(HHMI G)`
 - Household - FINS: `DSNHLO.SI EBEL. I NSTALL. JCL(HHMI GF I N)`
 - Preschm: `DSNHLO.SI EBEL. I NSTALL. JCL(SUBPSH)`
 - Address: `DSNHLO.SI EBEL. I NSTALL. JCL(ADMI GF I N)`
 - Preschm - FINS: `DSNHLO.SI EBEL. I NSTALL. JCL(SUBPSHF)`
 - Product Config: `DSNHLO.SI EBEL. I NSTALL. JCL(SUBPC)`
 - Product Config - FINS: `DSNHLO.SI EBEL. I NSTALL. JCL(SUBPCF)`
 - UpgISS: `DSNHLO.SI EBEL. I NSTALL. JCL(SUBUPGI S)`
- 4 Run the data migration job using the JCL in the dataset in which you have been placed. For example, if you selected option 3:Preschm, submit the JCL in the `DSNHLO.SI EBEL. I NSTALL. JCL(SUBPSH)` dataset.

This job automatically submits the *first* job in the job stream.

If the first Household, Preschm, and Product Configurator data migration job completes successfully, it automatically submits all subsequent jobs in the that data migration process flow. The number of jobs that are run varies according to your upgrade path.

If one of the automatically submitted job fails, the succeeding dependent job is not submitted and the automatic job submission sequence terminates. Correct the problem that caused the job failure and resubmit the individual failed job. When the job completes successfully, it then submits the next job in sequence.
- 5 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Restarting Upgrade Jobs That Fail” on page 183](#).
- 6 Press PF3 to return to the Target Database Data Migration Processes Menu and run the next data migration job in sequence.

About Migrating Preschm Data

Some of the PRESCHM jobs run independently but others are submitted in a defined order and cannot run until previous jobs have completed successfully; you can run the stand-alone jobs in parallel with those with dependencies. To see the serial flow of the PRESCHM jobs and the PRESCHM job dependencies, look at the PDS member `DSNHLO.SI EBEL. PRESCHM. JCL(@DEPFLOW)`.

About Migrating Address Data

The Address Migration job corresponds to scripts run previously on the midtier against the target system. This means that you *only* need to run this job if you ran the midtier Address Migration script (rpt_dup_addr_rowids.sql) and applied the midtier procedures previously noted for rowids identified by this script. Otherwise, you do not need to run this job. For further information, see [“Identifying and Resolving Duplicate Row IDs” on page 103](#).

Creating 8.0 Schema Indexes

Perform the following procedure to drop old schema indexes, create 8.0 schema and EIM indexes, and submit the rebuild jobs for the 8.0 Gen Primary indexes for the target database.

To run index jobs for the target database

- 1 On the Target Database In-Place Upgrade Menu, select option 9: SCINDEX Index Processes, and press Enter.

The following message appears:

Before proceeding, make sure ALL Data Migration “UPGI SS” jobs completed. Re-enter option 9 to continue.

- 2 Reselect option 9: SCINDEX Index Processes, and press Enter.

The Target SCINDEX Index Processes Menu appears. The panel ID is SBLSIXP.

- 3 Select option 0: Drop Old Schema Indexes, and press Enter.

You are placed in edit mode on dataset *DSNHLQ.SIEBEL.INSTALL.JCL(GPDRPIX)*.

- 4 Submit the JCL in the dataset to drop the old schema indexes.

- 5 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.

- 6 After submitting the job, enter cancel on the command line or press PF3 to save changes.

- 7 On the Target SCINDEX Index Processes Menu, select option 1: Create Pre-Gen Primary Indexes (DDL), and press Enter.

You are placed in edit mode on dataset *DSNHLQ.SIEBEL.INSTALL.JCL(GPRI X)*.

- 8 Submit the JCL in the dataset to run the DDL to create the Pre-Gen Primary Indexes.

- 9 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.

- 10 After submitting the job, enter cancel on the command line or press PF3 to save changes.

- 11 On the Target SCINDEX Index Processes Menu, select option 2: Rebuild Pre-Gen Primary Indexes (DDL), and press Enter. This job rebuilds the Pre-Gen primary indexes to create ROW_IDs for the data in the existing row.
You are placed in edit mode on dataset *DSNHLQ.SIEBEL.INSTALL.JCL(SUBGPPIX)*.
- 12 Submit the JCL in the dataset.
This job runs all the Pre-Gen Primary Index rebuild jobs in parallel. The number of index rebuild jobs that are run varies according to your upgrade path and the number of indexes you specified to be included in each rebuild job. See [“Restructuring the Index DDL” on page 163](#) for further information.
- 13 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 14 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 15 On the Target SCINDEX Index Processes Menu, select option 3: Create EIM Indexes (DDL), and press Enter.
You are placed in edit mode on dataset *DSNHLQ.SIEBEL.INSTALL.JCL(EIMIX)*.
- 16 Submit the JCL in the dataset to run the DDL to create new EIM indexes.
- 17 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 18 After submitting the job, press PF3 to return to the Target Database In-Place Upgrade Menu.

Migrating the Gen Primary 8.0 SQL to Update Data in Target Tables

Perform the following procedure to run the data migration scripts for Gen Primary:

- 1 On the Target Database In-Place Upgrade Menu, select option 10: Gen_Primary Data Migration, and press Enter.
You are placed in edit mode on dataset *DSNHLQ.SIEBEL.INSTALL.JCL(SUBGENP)*.
- 2 Submit the JCL in the dataset to update data in the target tables.
This submits all the Gen-Primary data migration job streams (the number of job streams varies by upgrade type). All of the jobs are run in parallel—no dependencies exist.
- 3 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 4 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Generating RUNSTATS Jobs

Upgrades: All upgrades.

Environments: All environments.

The following procedure generates RUNSTATS jobs for all Siebel table spaces. This process excludes all interface tables (EIM and tables with an _IF suffix).

NOTE: If you are performing a development environment upgrade, run the RUNSTATS jobs before starting the repository merge process.

To generate RUNSTATS jobs

- 1 On the Target Database In-Place Upgrade Menu, select option 11: Generate/Run Runstats, and press Enter.

The following message appears:

Before proceeding, make sure ALL "Gen-Primary" jobs completed. Re-enter option 11 to continue.

- 2 Reselect option 11: Generate/Run Runstats, and press Enter.
The Upgrade Runstats panel appears. The panel ID is SBLRSP.
- 3 Read the information relating to the RUNSTATS jobs on the Upgrade Runstats panel, then press Enter to start the RUNSTATS job generation process.
- 4 Messages are displayed as the jobs are generated. When the process is completed, press Enter.
You are placed in edit mode on the PDS dataset that contains the RUNSTATS jobs, *DSNHLO.SIEBEL.RUNST*.
- 5 Select the *DSNHLO.SIEBEL.RUNST(@RSTXREF)* PDS member.
This file contains information relating to each RUNSTATS job, for example, the table for which the job collects statistics and when statistics were last collected on the table.
- 6 Select the RUNSTATS jobs you want to run and run each one individually using the JCL in the dataset *DSNHLO.SIEBEL.RUNST(@RSTXREF)*.
Alternatively, go to the *DSNHLO.SIEBEL.INSTALL.JCL(SUBRUNST)* dataset and submit the JCL in the dataset to run all the RUNSTATS jobs for table spaces that did *not* have statistics collected during any of the previous upgrade processes.
- 7 Verify that the job(s) ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 8 Press PF3 to save changes.

Restarting Upgrade Jobs That Fail

Upgrades: All upgrades.

Environments: All environments.

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

This topic describes how to restart mainframe upgrade jobs that fail.

To restart a mainframe upgrade job that fails

- 1 Identify the job that failed under the SDSF exit (job status).

You can find the name of the job that failed using one of the following options on the In-Place Upgrade 8.0 Main Menu:

- Option 5: Staging Database Joblog
- Option 6: Target Database Joblog

See ["Running SQL in Siebel Logs" on page 243](#) for further information.

- 2 Determine the reason the job failed. You can determine the reason for the job failure by selecting the job on the SDSF output queue panel using the s action character.
- 3 Correct the problem.

If a job fails because of an SQL error, fix the problem in the appropriate SQL PDS member. For example, for PRESCHM jobs, you can locate the relevant SQL PDS member in the DSNHLQ.SIEBEL.PRESCHM.SQL data set.

NOTE: If you need help in performing these tasks or if you require confirmation that the tasks that you are about to run are correct, create a service request (SR) on My Oracle Support.

- 4 Once the problem has been identified and corrected, restart the job. You can do this by selecting the job that failed on the SDSF output queue using the SJ action character. This will automatically call up the next scheduled job allowing the upgrade process, for example PRESCHM, to continue.

14 Performing the Siebel Repository Merge

This chapter lists the steps involved in performing a repository merge during a development environment upgrade that are specific to DB2 for z/OS upgrades. You also need to perform the relevant tasks in the chapter of the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge. This chapter includes the following topics:

- [About Backing Up the New Customer Repository or Database Schema on page 185](#)
- [About Reorganizing Tables Before the Repository Merge on page 186](#)
- [Running the Repository Preparation Wizard on a Release 6.2.1 Repository on page 187](#)
- [Performing a Siebel Repository Merge on page 189](#)
- [Regenerating the Siebel Repository Definition Files on page 198](#)
- [Creating a New SRF File on page 200](#)

About Backing Up the New Customer Repository or Database Schema

Upgrades: All upgrades.

Environments: Development environment only.

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

The process of merging repositories to create the final customized repository used in the upgrade is time-intensive and resource-intensive. As a result, a merge might sometimes fail because of environmental factors, for example, space constraints. When this happens, the merge process continues, even if there is a fatal database error, and the errors might not be detected for some time.

If the merge fails, you must restore the database environment to its pre-merge state and run the merge again. There are two methods you can use to preserve the pre-merge environment so that you can restart the merge again if you need to. Before beginning the merge you can either:

- **Backup the entire database schema** then, if the merge fails, you can restore the schema and rerun the merge. The disadvantage of this option is that it is time consuming and disk space intensive.
- **Export the New Customer Repository** to create a backup copy. This is a simpler option. If the merge fails, import the New Customer Repository. See *Using Siebel Tools* for information on exporting and importing repositories using the Database Configuration Wizard.

About Reorganizing Tables Before the Repository Merge

Upgrades: All upgrades.

Environments: Development environment only.

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

During the repository merge process, objects from four separate repositories are read and compared. Because this is a memory-intensive process, it is recommended that you execute the REORG utility on certain tables before performing the repository merge to improve performance. The following tables receive a large number of inserts during each repository import; running REORGs on each of the following table's ROW_ID column will significantly increase the performance of the merge:

- S_APPLET
- S_APPLET_INTL
- S_APPLET_METH_MI
- S_APPL_WEB_TMPL
- S_APPL_WTMPL_IT
- S_BOCOMP
- S_BUSCOMP_UPROP
- S_COLUMN
- S_CONTROL
- S_CONTROL_INTL
- S_CONTROL_UPROP
- S_DDOWN_OBJECT
- S_EIM_FK_MAPCOL
- S_FIELD
- S_INDEX
- S_INDEX_COLUMN
- S_INTFLD_UPROP
- S_INT_CKEY_FLD
- S_INT_COMP
- S_INT_FIELD
- S_JOIN
- S_JOIN_SPEC
- S_LIST
- S_LIST_COL_INTL

- S_LIST_COLUMN
- S_PICKMAP
- S_SCREEN_VIEW
- S_UK_ATTJOIN
- S_USER_KEY_ATT
- S_VIEW_WTMPL_IT
- S_EIM_ATT_MAP

Running the Repository Preparation Wizard on a Release 6.2.1 Repository

6.x upgrades only. This topic applies to upgrades from Release 6.2.1.

Environments: Development environment only.

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

Before you run the repository merge on an ASCII database, you must run the Repository Preparation Wizard on the Prior Customer Repository. This utility is invoked from Siebel Tools and is used to perform the following procedures:

- **Migrate strings.** Release 8.0 supports locale strings in object-specific tables. In order to move the S_MSG data in your environment to the new locale table structure in Release 8.0, you need to run the Migrate Strings utility.

Migrate strings for each language that was supported in your prior repository.

- **Merge labels and fields.** Release 8.0 merges labels with controls so that the label is now recognized as the caption property of the control. Before you run the repository merge, you need to run the Repository Preparation Wizard to merge labels, controls and fields from the previous version based upon form applet layout.

NOTE: If you are upgrading to a Siebel Financial Services 8.0 application, do not use the default input file to merge labels and fields. See the Caution note in [Step 3 on page 188](#).

- **Merge Web templates.** Release 8.0 merges Web templates in Siebel Tools to only use Base and Edit/Query/New for the majority of the preconfigured applets. Previous releases of Siebel used four separate templates to deliver the same functionality. Before you run the repository merge, you need to run the Merge Templates utility to merge your applet web templates.

The Repository Preparation Wizard prompts you to confirm that you completed the following steps before it proceeds with the repository merge.

NOTE: If your upgrade fails while you are preparing the prior customer repository for merge, you need to restore your database to the most recent backup and complete repository preparation procedures. If your upgrade fails after you successfully completed the “Merge labels and fields” part of repository preparation, you must restore your database to the point after which you merged labels and fields.

To migrate strings, merge labels and fields, and merge applet Web templates

- 1 In Siebel Tools, navigate to the Tools menu, and then choose Tools > Upgrade > Prepare Repository.

Choose Prior Customer Repository as the repository that you wish to prepare.

The String Migration window appears.

- 2 In the String Migration window:

- a Select the language for which you wish to migrate strings.

- b To log migrated strings, click in the check box beside Log migrated strings, then click Browse to specify the log file.

If you have a single-language implementation and your base language is not English (ENU), you need to perform extra steps to upgrade labels of custom applets. See 476642.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Alert 426.

To continue, click Next. The Merge Labels and Fields window appears.

- 3 In the Merge Labels and Fields window:

- a Specify the input file by clicking the Browse button.

The utility operates on every form applet except those specified in the input file. The default input file is applets.txt. Modify the input file only if you have additional applets that do not need to be merged.

CAUTION: If you are upgrading to a Siebel Financial Services 8.0 application, do not use the default input file, applets.txt, because the repository preparation utility operates on every form applet except those specified by the input file, and Siebel Financial Services applications contain additional applets that should not be merged. Instead, browse to *SIEBEL_ROOT\Tools\Bin* directory and select the input file named *fi ns_applets.txt*.

- b Specify the location of Web templates.

To continue, click Next. The Merge Applet Web Templates window appears.

- 4 In the Merge Applet Web Templates window, click the Browse button to specify the same input file that you used in [Step 3 on page 188](#), then click Next.

NOTE: The utility operates on every form applet except those specified in the input file. The default input file is `applets.txt`. Modify the input file only if you have additional applets that you do not want merged.

The Merge Applet Web Templates window appears. Click OK to confirm that you want to proceed.

The wizard prepares your Prior Customer Repository for the merge.

The Repository Preparation Wizard records the results of the labels and fields merge and the template merge into the `labelmerge.txt` file and the `templatemerge.txt` file, respectively. If you want to view information about how the Repository Preparation wizard prepared your repository for merge, you can check these files, located in the `Siebel_Tools\temp` directory. More information about these log files is provided in the table below.

File Name	Description
<code>labelmerge.txt</code>	Generated by label and field merge. Lists all applets that are being modified. For each applet modified, displays whether the label and field merge was successful. If a control already has a caption, the old caption is logged for that control.
<code>templatemerge.txt</code>	Generated by the template merge. Lists all applets that are being modified and displays whether the merge was successful. For each applet, displays the following: <ul style="list-style-type: none"> ■ The template being used as the source (other templates are merged into this template). ■ The Applet Web Templates being inactivated. ■ Controls whose types are changed to make sure that they only appear in the correct modes.

The following errors in the `templatemerge.txt` file are acceptable if they are against standard applets, because there were no Edit or New applet web templates associated with standard applets in the previous version:

- Upgrading APPLET_NAME.
- Using Edit as source template.
- Could not find source template.

Performing a Siebel Repository Merge

Upgrades: All upgrades.

Environments: Development environment only.

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

During the repository merge, objects from the Prior Siebel Repository, Prior Customer Repository, and New Siebel Repository are compared by name to identify the total set of object differences. The process also determines how conflicts between repository changes are resolved as they are merged into the New Customer Repository.

There are three basic categories of object differences:

- New
- Deleted
- Modified

The repository merge executes the following processing steps to identify object differences:

- **New or deleted objects.** Identify objects that the customer has added by comparing their names in the Prior Customer Repository with the Prior Siebel Repository.

All new customer objects are carried over from the Prior Customer Repository to the New Customer Repository. The repository merge typically avoids deletion of objects. Most of the objects that are deleted in the Prior Customer Repository reappear after the merge. The merge does this to avoid accidental deletion of objects which may be required. It does, however, allow deletion of specific types of objects. Such objects are deleted from the New Customer Repository during the merge.

Objects of the following types are deleted from the New Customer Repository:

- Control
 - Chart
 - List Column
 - Applet Web Template Item
 - Page Tab
 - View Web Template Item
- **Objects with altered attributes.** Identifies objects that exist in both the Prior Customer Repository and the New Siebel Repository, and compares the attributes of each object to determine if they have been modified. Attribute comparisons are of interest only for those attributes which were changed by the customer.

If an object attribute was altered in the Prior Customer Repository, but not in the New Siebel Repository, the customer's attribute value is merged into the New Customer Repository.

A conflict occurs, however, if an object attribute was altered in both the Prior Customer Repository and the New Siebel Repository, in which case the values in all three repositories would be different. In this event, the repository merge process uses the setting of the object attribute's StandardWins flag to determine how to resolve the conflict. If this is set to Y, the attribute value from the New Siebel Repository is used; if this is set to N, the attribute value from the Prior Customer Repository is used. Conflict resolutions can be overridden for each object attribute in the New Customer Repository. See the chapter of the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge for further information on conflict resolution.

About the Repository Merge

The configuration utility that you ran while upgrading your development environment loaded two version 8.0 standard repositories. You must now use Siebel Tools to merge your existing custom configuration into one of these new repositories, creating a version 8.0 custom configuration that includes all of your previous configuration changes.

The four repositories that currently exist in your development database are listed in [Table 13](#).

Table 13. Development Environment Repositories

Repository Name	Description
Prior V6.2.1 Siebel Repository or Prior V7.x Siebel Repository	Standard version 6.2.1 or 7.x repository, depending on the version from which you are upgrading.
Prior Customer Repository	Customized version 6.2.1 or 7.x repository, depending on the version from which you are upgrading.
New Siebel Repository	Newly loaded version 8.0 standard repository.
New Customer Repository	Newly loaded version 8.0 repository into which your custom configuration is merged.

Follow the guidelines provided in [“Optimizing Performance of the Repository Merge” on page 191](#) to improve performance of the repository merge.

The repository merge is a memory-intensive process that fails if insufficient memory is available on the Siebel Tools workstation. Before beginning a repository merge, make sure that the following preparations have been completed on the developer workstation. Make sure that the developer workstation on which Siebel Tools is running has been upgraded to the newest available version.

The method you use to perform a repository merge depends on whether your database uses an ASCII or EBCDIC encoding scheme:

- For ASCII databases, perform the procedure, [“Merging the Repositories for an ASCII Database” on page 193](#).
- For EBCDIC databases, perform the procedure, [“Merging Repositories for an EBCDIC Database” on page 196](#).

Optimizing Performance of the Repository Merge

There are several ways in which you can reduce the time required to complete the repository merge.

- 1 Optimize the machine on which you are running the repository merge as follows:
 - Use a workstation with a minimum of 512 megabytes (MB) of RAM.
 - Allocate at least 2 GB of virtual memory, and a 2 GB page file. If the amount of virtual memory on the system is too low, performance degrades significantly.

- If necessary, increase the swap space, using the Control Panel System applet, and then restart the development workstation before proceeding.
- Allocate plenty of 32K sortwork space
- Make sure you have a high-performing network connection.

NOTE: A slow network connection significantly increases the time required for the repository merge.

- Close all other applications.
- Close all Siebel services.
- Defragment the disk. Fragmentation significantly affects system performance.
- On the workstation, check that the environment variable `SI EBEL_LOG_EVENTS` is set to zero. To check, enter the following command at the MS DOS prompt: `echo %SI EBEL_LOG_EVENTS%` If this variable is not set, no action is required. However, if `SI EBEL_LOG_EVENTS` is returned with a value other than zero, you must set it to zero by performing the following steps:
 - Close Siebel Tools and any other Siebel client applications.
 - Navigate to Start > Settings > Control Panel > System > Environment.
 - In the Environment dialog box, in the System Variables box, select `SI EBEL_LOG_EVENTS`. Enter 0 in the Value box, and click Set. Click OK.
 - Relaunch Siebel Tools. The new setting becomes active.

NOTE: The steps you need to take to set this variable may vary depending on the operating system you are using.

- 2 Optimize your database, because database performance can cause the repository merge to slow down considerably. Check the following:
 - Make sure that temporary table space has enough space allocated.
 - Make sure the database has enough space allocated.
 - Make sure that the top-most logging applet in tools has no extra rows (from previous repository merge runs) when starting the repository merge.
 - Make sure that the database is not loaded with users when repository merge is run (no other users should be connected).
 - Delete extra repositories from the database using Siebel Tools. Running the repository merge on a database with more than the four repositories which are needed for the repository merge degrades repository merge performance. Before deleting extra repositories, make backups.

NOTE: Deletion of extra repositories can take a few hours.

To increase the foreground performance of the repository merge

- 1 Navigate to Start > Control Panel > System.
- 2 Select the Advanced tab.

- 3 Select the Performance Options button.
 - 4 In the Application Response box, click the Applications radio button and click OK.
 - 5 While the repository merge process is running, click on the title bar of the Siebel Tools application to verify that the Siebel Tools application is the foreground application on the machine.
- NOTE:** After the repository merge process has finished, set the Performance setting back to its former value.

Merging the Repositories for an ASCII Database

Perform the following task to merge the development repositories for an ASCII database.

CAUTION: This procedure does not support EBCDIC databases. If you are upgrading a DB2 database that uses an EBCDIC encoding scheme, see “Merging Repositories for an EBCDIC Database” on page 196.

To merge the repository for an ASCII database

- 1 Log in to Siebel Tools and navigate to View > Options > Language Settings.
- 2 Verify that the language mode setting is set as desired.

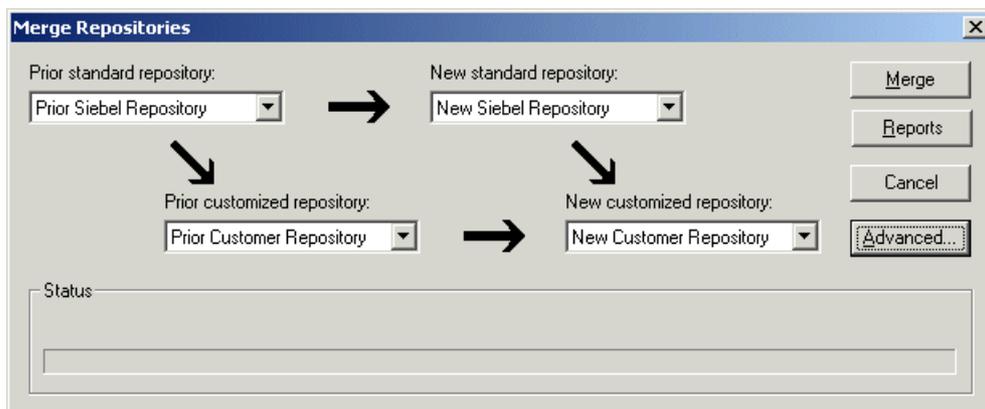
This is the user interface language for SRF files compiled from the New Customer Repository. It will also be the language used by the postmerge utilities.

- 3 Use the File > Open Repository command to open the Prior Customer Repository.

CAUTION: Open the Prior Customer Repository, not another repository. Later steps in the repository merge process fail if you open the wrong repository.

- 4 Choose Tools > Upgrade > Upgrade Application.

The Merge Repositories dialog box appears.



The Merge Repositories dialog box provides four options:

- **Merge.** This button merges the repositories you specify to produce a New Customer Repository.

- **Reports (for upgrades from Release 6.2.1 only)**. This button manually launches the postmerge reports for user interface configuration that are described in *Siebel Database Upgrade Guide*. For upgrades from Release 7.x, these reports are automatically generated after the repository merge.

CAUTION: Do not select Reports unless you are upgrading from Release 6.2.1 and have already completed your repository merge.

- **Cancel**. This button cancels the repository merge and exits the Merge Repositories dialog box.
- **Advanced**. This button opens the Merge Options dialog box described in [Step 7](#).

- 5 In the Merge Repositories dialog box, choose the appropriate repository name from each picklist, using the repository names listed in the following table.

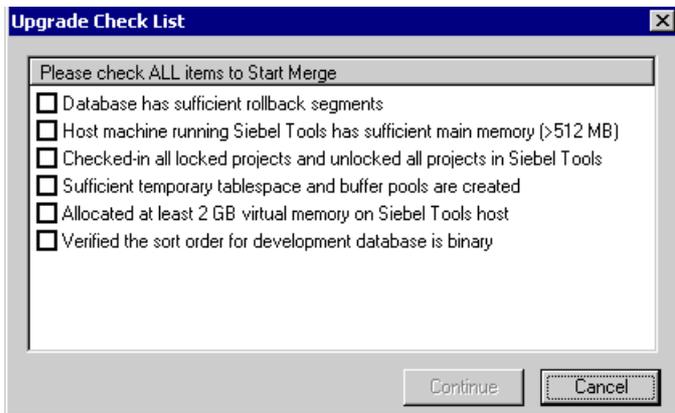
Drop-Down List Item	Value to Choose
Prior Standard Repository	Prior 6.2.1 Siebel Repository or Prior 7.x Siebel Repository, as appropriate for the version from which you are upgrading
Prior Customized Repository	Prior Customer Repository
New Standard Repository	New Siebel Repository
New Customized Repository	New Customer Repository

- 6 Review the settings in the Merge Repositories dialog box, and then click Advanced.
The Merge Options dialog box appears.
- 7 In the Merge Options dialog box, click *Abort merge if more than x errors occur*, as desired. This option aborts the repository merge automatically if more than a designated number of errors occur.

CAUTION: The typical repository merge generates many benign errors. If you select this option, set the number of errors to a large value. This will help prevent the repository merge from aborting due to benign errors.

- 8 To continue, click OK.
- 9 Click Merge on the Merge Repositories dialog box.
The Upgrade Check List dialog box appears.

- 10 In the Upgrade Check List dialog box, you must confirm that your environment meets the requirements for a successful repository merge. Review each requirement and select the check box if your configuration meets or exceeds the requirement.



- 11 To continue, click Continue.

A warning screen prompts you to prepare the Prior Customer Repository for the merge if you are upgrading from Release 6.2.1.

- 12 In the Warning screen, choose the option for your upgrade path:

- For upgrades from Release 6.2.1, click Yes to execute Prior Customer Repository preparation steps.

- For upgrades from Release 7.x, click No to skip the Prior Customer Repository preparation steps. You completed these steps during a prior upgrade to Siebel 7.



The merge process begins. The repository merge process can take, on average, five to seven hours to complete. Timings can vary greatly depending on the kind of machine, the hardware configuration, virtual memory allocation, the use of the upgrade inheritance feature, and level of customizations in the customer repository (such as new records or changed attributes). In addition to merging the base repository, all locales are merged. Plan additional time for each language, including the base language.

Customizations are moved to the New Customer Repository, which results in a large number of database operations (inserts and updates). For each of these operations, logging records are created, and these log records also affect performance. If the repository is large, or the database setup is not optimal, this may take much longer.

- 13 After the merge completes, a dialog displays requesting that you make a backup of the New Customer Repository. Back up the New Customer Repository and click OK in the dialog box.

This launches the postmerge utilities. The postmerge utilities resolve common, merge-related user interface issues. For more information about the postmerge utilities, see [“Creating a New SRF File” on page 200](#).

- 14 After the postmerge utilities complete, do the following:
 - Click OK in the Postmerge Utilities dialog box.
 - Close the Merge Options dialog box.

CAUTION: Do not click Merge in the Merge Options dialog box. This will restart the merge.

When the postmerge utilities start, the Postmerge Utilities dialog box displays the log entries from the utilities. The steps for reviewing the log are part of evaluating the UI and are covered in a later process. The postmerge utilities can require several hours to complete.

Merging Repositories for an EBCDIC Database

Perform the following task to complete a repository merge for an EBCDIC database.

To perform a repository merge for an EBCDIC database

1 Use the Import/Export Repository option in the Database Configuration Wizard to export the following repositories from your prior EBCDIC database:

- Prior Standard Repository
- Prior Customized Repository
- New Standard Repository
- New Customer Repository

See *Using Siebel Tools* for information on exporting repositories using the Database Configuration Wizard.

2 Prepare a new Siebel 8.0 ASCII database on which you will perform the repository merge. In the storage control file, the following tables must be defined with CLOBS = Yes:

- S_SCHMST_DBSCPT
- S_BI TMAP_DATA
- S_SERVICE_SCRIPT

3 Use the Import/Export Repository option in the Database Configuration Wizard to import the repositories exported from the EBCDIC database in [Step 1](#) into the ASCII database prepared in [Step 2](#).

NOTE: Make sure you select the **Import Custom Repository** option to import all languages used.

See *Using Siebel Tools* for information on importing repositories using the Database Configuration Wizard.

4 Launch Siebel Tools against the ASCII database.

5 Run the repository merge using the procedure, [“Merging the Repositories for an ASCII Database” on page 193](#).

6 Generate Siebel EIM temporary columns. This task is described in the chapter of the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge.

7 Review the repository merge results to determine if the merge was successful. This task is described in the chapter of the *Siebel Database Upgrade Guide* that describes how to perform the Siebel Repository merge. Verify that the repository merge was successful, and that all reported validation messages are either acceptable or fixed.

8 Use the Database Configuration Wizard to export the merged repository (the New Customer Repository) from the ASCII database.

9 Rename the existing New Customer Repository in the EBCDIC database.

10 Use the Database Configuration Wizard to import the merged New Customer Repository back into the EBCDIC database.

11 From the Tools application, pointing to the ASCII database, click on *Repository* in the Object Explorer and copy the value against the Comments column for the New Customer Repository.

- 12 Connect to the EBCDIC database through the DB2 command line and update the Comments column (with the copied value) on the table S_REPOSITORY for name= "New Customer Repository".

For example, if you copied a *Comments* value of:

```
APPLIED_PATCHES: Grid, UI NavUpgrade, MVGUpgPatch77, UI NavUpgrade, PCLWebTempl Swap, WFD, PM8.0; UpgEi mCol ,
```

Then, you need to execute the following command against the EBCDIC database:

```
Update s_repository set comments
```

```
= ' APPLIED_PATCHES: Grid, UI NavUpgrade, MVGUpgPatch77, UI NavUpgrade, PCLWebTempl Swap, WFD, PM8.0; UpgEi mCol '
```

Where name='New Customer Repository'

Regenerating the Siebel Repository Definition Files

Upgrades: All upgrades.

Environments: Development environment only.

Platforms: All platforms.

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

If you have modified repository objects after the development environment upgrade (upgphys) and before upgrading the production test environment, you must regenerate the schema.ddl and custrep.dat files. These files were created during the upgphys:

- **Schema.ddl.** This file contains the logical definition of the Siebel database.
- **Custrep.dat.** This file contains the definition of repository objects.

These files are used as input to the production test and production environment upgrades. If you modify the object definitions or the schema definitions in the repository after these files have been created, you must regenerate the files.

Regenerating the schema.ddl File

Use this procedure to regenerate the schema.ddl file.

To regenerate the schema.ddl file

- 1 On the Siebel Server where the Siebel Database Server files are installed, navigate to the following location:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Run the following command:

```
ddl dict /u DatabaseOwner /p Password /c "ODBCDataSource" /d TableOwner /f  
DBSRVR_ROOT\DatabasePlatform\schema.ddl /e y /a y /l Siebel LogDir\schedict.log /n  
"Siebel Repository" /t dir
```

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 Configuration Utilities installation directory.
 - *DatabasePlatform* is the Siebel Database Configuration Utilities directory name for the database, that is, DB2390. The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
 - *SiebelLogdir* is the path to the directory where you want the output log placed (log output directory). The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
- 3 After the command completes, review the output logs for errors. If the log indicates there are errors, create a service request (SR) on My Oracle Support.

Regenerating the custrep.dat File

Use this procedure to regenerate the custrep.dat file.

To regenerate the custrep.dat file

- 1 On the Siebel Server where the Siebel Database Configuration Utilities files are installed, navigate to the following location:

Windows: *SIEBEL_ROOT*\bin

UNIX: *\$SIEBEL_ROOT*/bin

- 2 Run the following command:

```
repimexp /a e /u DatabaseOwner /p Password /c "ODBCDataSource" /d TableOwner  
/r "New Customer Repository" /f DBSRVR_ROOT\DatabasePlatform\custrep.dat  
/l Siebel LogDir\explog
```

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 Configuration Utilities installation directory. The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
 - *DatabasePlatform* is the Siebel Database Configuration Utilities directory name for the database, that is, DB2390.
 - *SiebelLogdir* is the path to the directory where you want the output log placed (log output directory). The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
- 3 After the command completes, review the output logs for errors. If the log indicates there are errors, create a service request (SR) on My Oracle Support.

Creating a New SRF File

Upgrades: All upgrades.

Environments: Development environment only.

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

You can optionally create an .srf file immediately after the repository merge. This enables you to run an application and evaluate whether you have resolved attribute conflicts correctly.

After the you have completed the upgphys portion of the upgrade, you must create a new .srf file. This file includes the merged repository changes.

If you will review your UI in more than one language, set the Tools Language setting and compile an .srf file for each language. For information about how to compile an .srf file for a specific language, see *Siebel Global Deployment Guide*.

Requirements: The repository merge must have been successful.

Compiling a New SRF File Immediately After the Repository Merge

Perform this optional procedure immediately after the repository merge and before you perform the upgphys upgrade in the development environment. Repeat this procedure as needed to help evaluate how you have resolved attribute conflicts.

Do not use this procedure to compile an .srf file if you have completed the upgphys upgrade. Instead, see ["Compiling a New .srf File After the Upgphys Upgrade"](#) on page 201.

To compile a new .srf file after the repository merge

- 1 Navigate to the following directory on an upgraded Siebel Server you will use to test applications in the development environment. Verify that this is *not* the directory where Siebel Tools is installed:

Windows: `SI EBEL_ROOT\OBJECTS\language`

UNIX: `$SI EBEL_ROOT/OBJECTS/language`

where:

language is the installed language, for example, ENU.

- 2 Rename the siebel.srf file to siebel_release.srf.
- 3 On the Siebel Tools development workstation, connect to the database against which you ran the merge.
- 4 Start Siebel Tools.
- 5 Choose File > Open Repository.
- 6 Select New Customer Repository.
- 7 Choose Tools > Compile Projects.
- 8 Select All Projects.
- 9 Enter the path to the Siebel Server OBJECTS directory where you renamed the siebel.srf file, and specify siebel.srf as the file name.

CAUTION: Do *not* enter the path to the Siebel Tools OBJECTS directory. You will overwrite and destroy the Siebel Tools .srf file.
- 10 Click Compile.
- 11 Copy the contents of the Siebel Tools WEBTEMPL directory to the Siebel Server WEBTEMPL directory.
- 12 Start the application.

Compiling a New .srf File After the Upghys Upgrade

After completing the upghys upgrade in the development environment, you must compile a new .srf file. This updates the file with the merged repository changes. This is the .srf file you will copy to the Siebel Servers in your test environment.

To compile a new .srf file after the upgphys upgrade

- 1 Navigate to the following directory on an upgraded Siebel Server you will use to test applications. Verify that this is not the directory where Siebel Tools is installed:

Windows: `SI EBEL_ROOT\OBJECTS\language`

UNIX: `$SI EBEL_ROOT/OBJECTS/language`

where:

`language` is the installed language, for example, ENU.

- 2 Rename the `si ebel .srf` file to `si ebel _rel ease.srf`.
- 3 On the Siebel Tools development workstation, connect to the database against which you ran the merge.
- 4 Start Siebel Tools.
- 5 Choose File > Open Repository.
- 6 Select Siebel Repository.
- 7 Choose Tools > Compile Projects.
- 8 Select All Projects.
- 9 Enter the path to the Siebel Server OBJECTS directory where you renamed the `siebel.srf` file, and specify `siebel.srf` as the file name.

CAUTION: Do *not* enter the path to the Siebel Tools OBJECTS directory. You will overwrite and destroy the Siebel Tools `.srf` file.

- 10 Click Compile.

Siebel Tools creates a new `.srf` file and places it at the specified location on the Siebel Server.

15 Migrating Siebel 6.2.1 Customizations

If you are upgrading your development environment from Release 6.2.1, there are a number of tasks you must perform to make sure your Siebel customizations are migrated when you upgrade to Release 8.0. These tasks are described in this chapter. This chapter contains the following topics:

- [About Migrating to the Siebel Web Client on page 203](#)
- [About Migrating 6.2.1 Scripts on page 209](#)
- [About Migrating 6.2.1 Client-Side Interfaces on page 210](#)
- [Identifying What Will Be Migrated on page 211](#)
- [Exposing Hidden Properties of Applets and Views on page 212](#)
- [Running the Siebel Web Client Migration Wizard on page 213](#)
- [Reviewing Applet and View Migration on page 216](#)
- [Running the Web Layout Wizard on page 217](#)
- [Migrating Scripts Attached to Controls on page 218](#)
- [Migrating Scripts Attached to Applets on page 220](#)
- [Migrating Business Component, Business Service, and Application Scripts on page 223](#)
- [Migrating Outbound COM Interfaces on page 224](#)
- [Migrating Inbound COM Interfaces on page 225](#)
- [Regenerating the Postmerge Utilities Log on page 228](#)
- [Reviewing Customized Business Components on page 228](#)

About Migrating to the Siebel Web Client

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

In Release 8.0, the Siebel Web Client is used as the deployment framework for all Siebel Business Applications. Siebel Web Client runs in a standard browser on the end user's client computer, and does not require additional persistent software installed on the client. The browser connects through a Web server to the Siebel Server, which executes business logic and accesses data from the Siebel database. Only the user interface layer of the Siebel Business Applications architecture resides on the user computer.

The Siebel Web Client Migration Wizard upgrades customized view and applet definitions used by the Siebel Dedicated Client, Windows Thin Client, and Java Thin Client in previous Siebel releases to the new Siebel Web Client. You run this wizard after the repository has been upgraded.

The Siebel Web Client Migration Wizard performs the following functions:

- Recognize what types of applets are migrated
- Provide a model applet for each applet type
- Select a template automatically based on the model applet
- Populate the model applet automatically
- Migrate applets and views from Siebel 6.2.1 customer applications

The Release 8.0 Siebel Web Client migration results in a grid-based layout for form applets. The grid-based layout allows you to control the layout of a form applet using the Web Layout Editor. You can drag and drop controls without having to modify the Web templates. Siebel Web Client Migration of form applets using grid-based functionality preserves applet models, similar to the Release 6.2.1 dynamic grid.

The Siebel Web Client Migration Wizard does not migrate Siebel VB or Siebel eScript scripts attached to applets. You must either move these scripts to the server or rewrite them in JavaScript, the Siebel Web Client-supported language for user interface scripting.

The Release 8.0 Siebel Web Client supports most of the events and methods on user interface objects supported by previous Siebel releases. In addition, the Siebel Web Client provides comparable inbound and outbound integration capabilities to what was available in the business object interfaces in previous releases of the Siebel Dedicated Client. You may need to perform migration tasks to achieve optimal functionality.

For detailed information about the grid layout, see *Configuring Siebel Business Applications*.

Example of the Siebel Web Client UI

Figure 4 shows a standard Siebel 6.2.1 Windows Client. Figure 5 on page 206 shows a view that has been migrated from the Windows Client to the Siebel Web Client. This view is running in a browser.

The screenshot shows the Siebel Web Sales - My Contacts interface. The main window displays a list of contacts with the following columns: New, Last Name, First Name, Mr/Ms, Call, Work Phone #, Work Fax #, and Home Phone #. The 'Call' column contains checkmarks for several contacts. The 'New' column has a right-pointing arrow for the first contact, 'Alan Wiley'.

New	Last Name	First Name	Mr/Ms	Call	Work Phone #	Work Fax #	Home Phone #
>	Alan	Wiley	Mr.		(425) 810-9988	(425) 810-9706	
	Alder	Kevin	Mr.		(415) 491-2360 x334	(415) 491-2200	
	Allen	Donna	Ms.	✓	(415) 717-2002	(415) 717-9999	(415) 247-3323
	Alterman	David	Mr.		(510) 452-2000	(510) 452-2000	
	Alvarez	Harlan	Mr.		(510) 223-9320	(510) 223-2323	
	Atkins	Charlie	Mr.	✓	(510) 223-2319	(510) 223-2328	
	Bass	Ken	Mr.		(415) 329-1000	(415) 329-1001	
	Basset	Mary	Ms.		(510) 292-0945	(512) 292-6544	
	Bell	Jeff	Mr.		(415) 717-2009	(415) 717-9999	
	Bell	Robert			(704) 992-3487		
	Bingenheimer	David	Mr.		(925) 485-5439	(510) 594-6111	(925) 640-6021
	Bolt	Maria	Ms.		(815) 435-8899	(815) 435-8800	
	Brianic	Byrne	M		(510) 782-1016	(510) 782-1023	
	Brighton	Amy	Mrs.		(415) 654-9872	(415) 654-1111	
	Brown	Scott	Mr.	✓	(510) 567-2300	(510) 567-2222	
	Burson	Robert	Mr.		(404) 493-2400	(404) 493-2401	
	Campos	Miguel	Mr.		(510) 292-1228	(512) 292-6544	
	Choi	Victor					
	Delgado	Abel	Mr.				(415) 234-5847
	Douglas	John	Mr.		(510) 292-1002	(512) 292-6544	
	Finkel	Michelle	Ms.		(617) 254-8255	(617) 254-5002	
	Floyd	Sara	Ms.		(415) 717-2056	(415) 717-9999	
	Franklin	Bill	Mr.	✓	(212) 733-9000	(212) 733-9110	

Figure 4. Contact List View—Windows Client

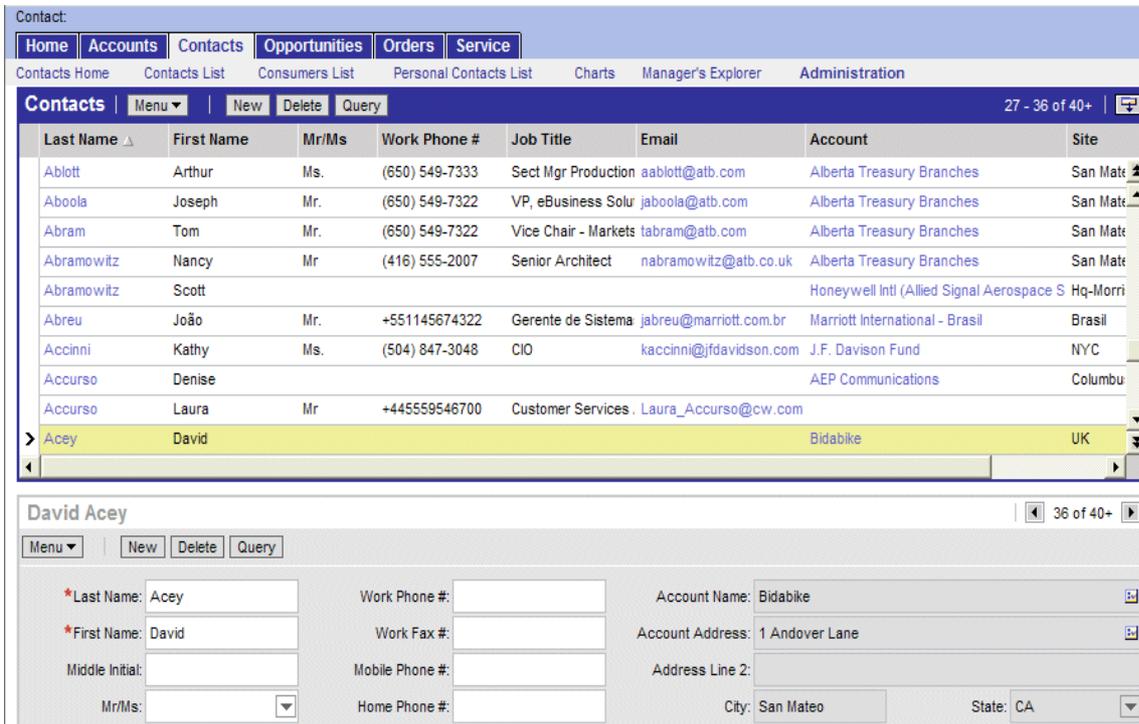


Figure 5. Contact List View—Siebel Web Client

About the Siebel Web Client Migration Wizard

When you upgrade to Release 8.0, standard, uncustomized applets and views are automatically converted to Web-based layouts that can be rendered in a browser. Applets and views that you have created or modified are not converted automatically.

The upgrade does not convert applets or views that you have created or modified. To migrate created or modified applets and views, you must run the Siebel Web Client Migration Wizard. The wizard does the following:

- Identifies created or modified applets and views.
- Puts a comment in the Comments field of each. This identifies them so you can query for them.
- Provides a model applets and model views. These models provide Web templates that determine the layout of the applet or view.
- Migrates applets and views to a Web-based layout using the selected Web templates.

The Siebel Web Client Migration Wizard converts applets and views as a batch.

Web Layout Wizard

The Web Layout Wizard allows you to migrate created or modified applets and views individually. You can select specific applets and views for migration. Then you select the desired model applet or model view to be used for the migration.

Recommended Migration Strategy

The recommended strategy for using the two wizards is as follows:

- Run the Siebel Web Client Migration Wizard in Mark Applets and Views mode to identify which applets and views it will migrate. Review the marked applets and views, and select the desired model applets and model views.
- Expose hidden properties of applets and views. This improves the accuracy of the migration.
- Test the Siebel Web Client Migration Wizard against a few applets and views before running the wizard against the entire repository. Verify all model applets and model views work correctly.
- Migrate all the created and modified applets and views as a batch using the Siebel Web Client Migration Wizard.
- Review the migrated applets and views.
- Migrate again individual applets and views or groups of applets and views as needed using the Web Layout Wizard.

NOTE: Using the Web Layout Wizard as the only way to migrate your applets and views is time-consuming and is not recommended.

Model Applets and Views

The Release 8.0 repository contains model applets and views. The Siebel Web Client Migration Wizard and Web Layout Wizard use these to migrate created and modified applets and views.

Before using model applets, compare the applets you intend to migrate with the layout of the model applets. Modify model applets as needed. For example, you can modify model applets to use different Web templates, controls, or control mappings.

Modifying model applets used for converting to grid-based layout is not recommended.

If you create new model applets, verify that you have defined all the Applet Web Template Items that you will need. Also, verify that you have assigned a unique HTML sequence number to all controls and list columns.

Before using model views, compare the views you intend to migrate with the layout of the model views. Modify the model views to meet your requirements.

Table 14 lists preconfigured model applets and views available in Release 8.0.

Table 14. Model Applets and Views

Model Applets and Views
Model chart applet
Model form applet
Model list applet
Model association list applet
Model MVG applet

Table 14. Model Applets and Views

Model Applets and Views
Model pick applet
Model tree applet
Model more applets view
Model one applet view
Model tree applet view

How the Migration Is Done

The Siebel Web Client Migration Wizard and Web Layout Wizard map items from the dedicated client layout to the Siebel Web Client using the following information from model applets and views:

- **Modes to map applet Web templates.** Modes determine the kind of actions available in applets to users. For example, some applets are read-only. Other applets allow users to edit data. Typical modes include Base, Edit, and Edit List. The mode also determines which buttons appear in Web templates. For example, the Edit button appears in applets set to Edit mode, but does not appear in applets set to Base (Read Only) mode. For more information about applets, see *Configuring Siebel Business Applications*.
- **Web templates to use for each mode.** Templates contain placeholders for applets and are used to render Siebel views. The mode selected determines the Web template to be used. For more information about Web templates, see *Configuring Siebel Business Applications*.

For chart and tree applets, the wizard only uses the Web template information. For this reason, any applet of the appropriate type may be used as a model applet for chart and tree applets.
- **Buttons that appear on migrated applets.** Web applets typically contain additional buttons that were not used by their dedicated client predecessors. For example, the Next and Previous buttons used to scroll through record sets in the Siebel Web Client did not appear in the dedicated client. During migration, the wizards automatically propagate buttons contained in the model applet so that these types of buttons do not require manual work.
- **Grid-based template and applet controls.** The grid layout conversion process converts a grid-based template and a set of applet controls with pixel-based coordinates into a Web-enabled layout that closely matches the original 6.2.1 layout. For created or modified form applets that are converted to grid-based layout, you must manually map the buttons.

Form Applets and Grid-Based Layout

Many form applets are converted to grid-based layout during upgrade. The following summarizes how grid-based applet Web templates differ from standard applet Web templates:

- You can modify the layout of the form using Siebel Tools without having to modify the Web template.
- With grid-based templates, labels and controls behave as separate items in the Web Layout Editor. This allows you to place them independently in the applet layout. However, labels and controls are really a single object in the repository with one set of shared properties.

- Grid-based templates do not automatically compress empty space in a column. There are two applet Web templates that support grid layout, as shown in [Table 15](#).

Table 15. Grid Layout Templates

Web Template	File Name	Comments
Applet Form Grid Layout	CCAppletFormGridLayout.swt	Use with all modes of form applets.
Applet Popup Form Grid Layout	CCAppletPopupFormGridLayout.swt	Use with all modes of popup form applets.

Not all form applets can be converted to a grid layout.

For a particular Applet Web Template mode, if the Web template type is *not* Applet Template—Form or Applet Template, the applet is not converted to grid layout.

If the Web template type is Applet Template—Form or Applet Template, but either the Applet name or the Web template name is specified in the awtcvtcfg.txt file (located in the binary subdirectory of the root Siebel Tools installation directory), the applet is not converted to grid layout.

About Migrating 6.2.1 Scripts

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

You must migrate scripts on UI objects to server scripts or browser scripts when upgrading to Release 8.0.

Server Scripts

Server scripts execute in the Application Object Manager. They existed in prior releases and continue to be supported. These scripts are written in Siebel VB (for Windows platforms) and Siebel eScript (for Windows or UNIX). There are four types of server scripts:

- Business component server script
- Business service server script
- Application server script
- Applet server script

Server scripts enable you to script the behavior of business components, business services, applications, and applets by writing event handlers for events exposed by the scripting model.

Browser Scripts

Browser scripts execute in the browser. Browser scripts are written in JavaScript and interact with the document object model (DOM) as well as the Siebel Object Model available in the high interactivity Siebel Web Client framework. You can script the behavior of Siebel events as well as browser events that are exposed through the DOM.

Siebel Tools allows you to write scripts by selecting the appropriate User Agent. For information about browser script techniques, see 476879.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 386. For more information about events that are exposed through the DOM, see *Siebel Tools Online Help*.

You must migrate Release 6.2.1 scripts written on applets to applet server scripts or business component server scripts, or you must rewrite them as browser scripts.

Applet script migration includes the following types of tasks:

- Moving Release 6.2.1 applet scripts such as Control_Click event handlers to business components
- Rescripting Release 6.2.1 ActiveX controls in browser script
- Rewriting other Release 6.2.1 applet scripts in browser script

To identify the number of 6.2.1 applet scripts to be reviewed or migrated, look at the Object List Editor in Flat mode in Siebel Tools.

You must review Release 6.2.1 scripts on the business components, business services, and application to identify references to methods, such as UI methods, that can not be accessed from server scripts.

To locate server scripts that you may need to modify, run the Siebel Tools Validator on business components, business services, and applications for which the Scripted flag is set to *True*.

You must modify Release 6.2.1 business component and application scripts that contain references to UI methods. For example, you need to replace calls to MsgBox with calls to the RaiseError/RaiseErrorText method. Other modifications may include moving scripts to Browser script. For more information about modifying business components and application scripts, see *Siebel Tools Online Help*.

About Migrating 6.2.1 Client-Side Interfaces

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

Siebel Web Clients support most of the events and methods on user interface objects supported by previous Siebel releases. In addition, the Siebel Web Client provides inbound and outbound client-side integration capabilities comparable to what was available in the business object interfaces in previous releases of the Siebel Dedicated Client. You may need to perform migration tasks to achieve optimal functionality. The following capabilities are provided:

- **Outbound COM interfaces.** As in previous releases, you can invoke desktop applications, such as Microsoft Excel, from the Siebel client.

- **Inbound COM Interfaces.** In Release 6.2.1, desktop applications communicated with the Siebel Client through an Automation interface. In Release 8.0, the high interactivity Siebel Web Client provides similar automation interfaces.

This functionality is only available in a high interactivity framework, and it is applicable for the Siebel Web Client, Siebel Mobile Web Client, and Siebel Developer Web Client. (Unless otherwise specified, references in this section to the Siebel client may refer to any or all of these clients.)

Identifying What Will Be Migrated

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

This topic describes how to use the Siebel Web Client Migration Wizard to identify the created and modified applets and views that will be migrated when upgrading to Release 8.0.

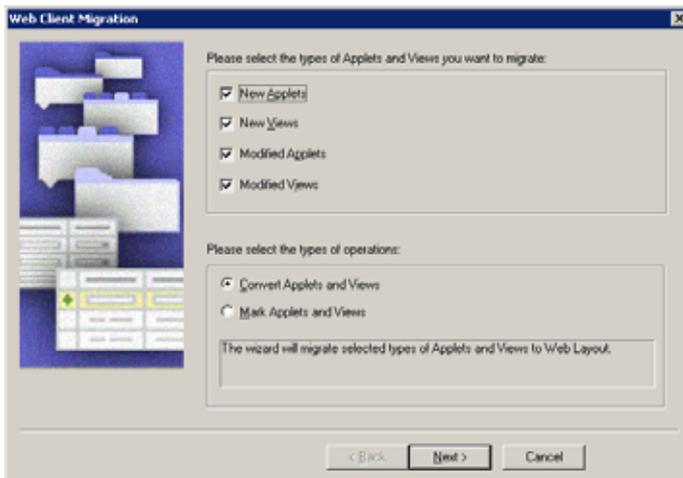
For more information on migrating created or modified applets and views to the Siebel Web Client, see [“About Migrating to the Siebel Web Client” on page 203.](#)

Requirements: The repository merge must have been completed successfully.

To identify which objects will be migrated

- 1 In Siebel Tools, navigate to Screens > Application Upgrader > Application Upgrade Object List.
- 2 In the Application Upgrades list, select the record of the merge.
- 3 Select Tools > Upgrade > Web Client Migration.

This starts the Siebel Web Client Migration Wizard. The model applet selection screen appears.



- 4 Select the types of applets and views that you want to migrate:

- New applets
 - New views
 - Modified applets
 - Modified views
- 5 Select Mark Applets and Views and click Next.
- No migration occurs.
- Instead, the wizard puts TBMN (to be migrated new, for created applets and views) or TBMM (to be migrated modified, for modified applets and views) in the Comment field for applets and views to be migrated.
- 6 Query for objects with TBMN or TBMM in the Comments field to create a list of the objects that the Siebel Web Client Migration Wizard identified.

Exposing Hidden Properties of Applets and Views

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

As of Release 7.0.x, certain properties in Siebel Tools are no longer applicable for the Siebel Web Client. By default, these properties are hidden. Before you begin the migration to the Siebel 8.0 Web Client, expose these properties to make the migration of your customizations to Release 8.0 easier.

For more information on migrating created or modified applets and views to the Siebel Web Client, see [“About Migrating to the Siebel Web Client” on page 203.](#)

To expose hidden properties

- 1 Using a text editor, open the tool s.cfg file located in the following directory:
`Siebel Tool sDir\bin\LANG`
- 2 Under the [Siebel] section in the tool s.cfg file, locate the following parameter:
`ClientConfigurationMode`
The default value for ClientConfigurationMode is Web. This value hides properties that are no longer applicable.
- 3 To expose these attributes for migration, change the value from Web to All.
- 4 Save and close the tool s.cfg file.
- 5 Launch Siebel Tools.
The previously hidden properties are now visible.

- 6 After completing the Siebel Web Client migration to Release 7.x, reset the `ClientConfigurationMode` parameter to `Web` to hide properties that are no longer applicable.

Running the Siebel Web Client Migration Wizard

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

This topic describes how to use the Siebel Web Client Migration Wizard to batch-migrate your new and modified applets and views to Release 8.0.

For more information on migrating created or modified applets and views to the Siebel Web Client, see [“About Migrating to the Siebel Web Client” on page 203.](#)

Requirements

Before running the Siebel Web Client Migration Wizard, complete the following tasks:

- Complete the repository merge successfully.
- Identify model applets and model views. See [“Identifying What Will Be Migrated” on page 211.](#) This task is optional.
- Expose hidden properties on objects. See [“Exposing Hidden Properties of Applets and Views” on page 212.](#)

Related Topic

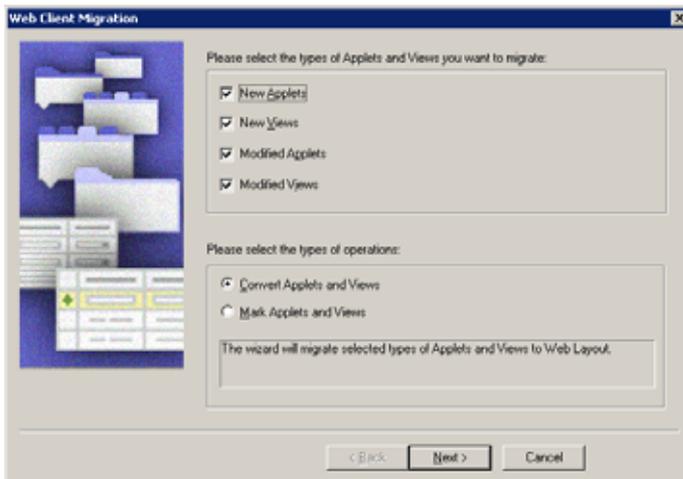
[“Running the Web Layout Wizard” on page 217](#)

To run the Siebel Web Client Migration Wizard

- 1 In Siebel Tools, navigate to Screens > Application Upgrader > Application Upgrade Object List.
- 2 In the Application Upgrades list, select the record of the merge.

- 3 Select Tools > Upgrade > Web Client Migration.

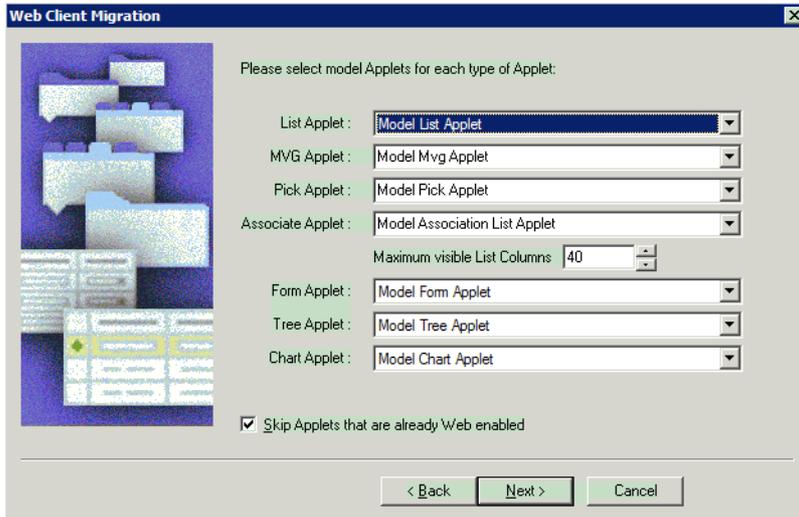
This starts the Siebel Web Client Migration Wizard. The model applet selection screen appears.



- 4 Select the types of applets and views that you want to migrate:
 - New applets
 - New views
 - Modified applets
 - Modified views
- 5 To continue, click Next.

6 Select a model applet for each applet type.

You can accept the recommended default values or select an alternate value.



By default, the maximum number of visible list columns is 40. Also by default, the option to Skip Applets that are already Web enabled is selected. This option preserves prior customizations if you have an existing customer application that uses Web templates.

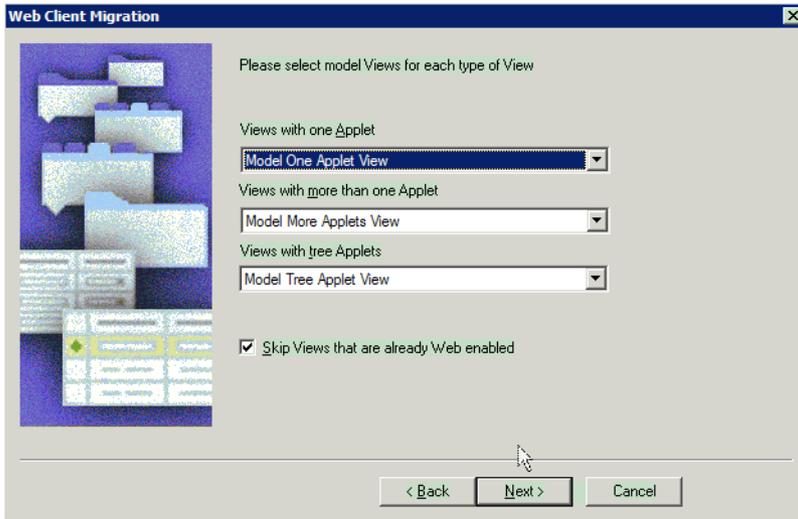
To continue, click Next.

7 Select a model view for each category of view:

- View with one applet
- View with two or more applets

■ Views containing tree applets

You can use any view of the appropriate category as a model view. The information used by the Siebel Web Client Migration Wizard includes the templates to be used and the applet modes to use for the View Web Template Items.



By default, the option to Skip Views that are already Web enabled is selected. This option preserves prior customizations if you have an existing customer application that uses Web templates.

8 To start the migration, click Next.

When the wizard completes, it appends the following to the Comments field of objects that it has migrated:

- **MigN.** Indicates new applets or views that have been migrated.
- **MigM.** Indicates modified applets or views that have been migrated.

Reviewing Applet and View Migration

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

The Siebel Web Client Migration Wizard and Web Layout Wizard migrate new and modified applets and views to the Siebel Web Client. After running either wizard, review the migrated applets and views.

The following issues may require that you remigrate an applet:

- A set of nonfield controls is misplaced or absent in the Web layout.
- The template needs to be changed.

- Field controls are misplaced in the Web layout.

The following issues may require that you remigrate a view:

- The mode of a View Web Template Item is incorrect.
- The template needs to be changed.

Issues other than those above may need to be resolved manually.

For more information on migrating created or modified applets and views to the Siebel Web Client, see [“About Migrating to the Siebel Web Client” on page 203](#).

To review applet and view migration

- 1 In Siebel Tools, set the Object Explorer to Flat view and select Applet or View.
- 2 In the list view Comments field, query for Mi gN to review new applets or views. Query for Mi gM to review modified applets for views.
- 3 For applets, sort the resulting list of applets by Class.
- 4 For each applet or view, right-click on it and select Edit Web Layout.
- 5 Review the applet or view layout:
 - For applets, use the Applet Web Editor to make any necessary changes.
 - For views, use the View Web Editor to make any necessary changes.
- 6 To change the model applet or model view, run the Web Layout Wizard.

Running the Web Layout Wizard

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

If some views or applets did not migrate properly when you ran the Siebel Web Client Migration Wizard to upgrade to Release 8.0, you can remigrate them using the Web Layout Wizard. With the Web Layout Wizard you can migrate applets or views individually or in groups.

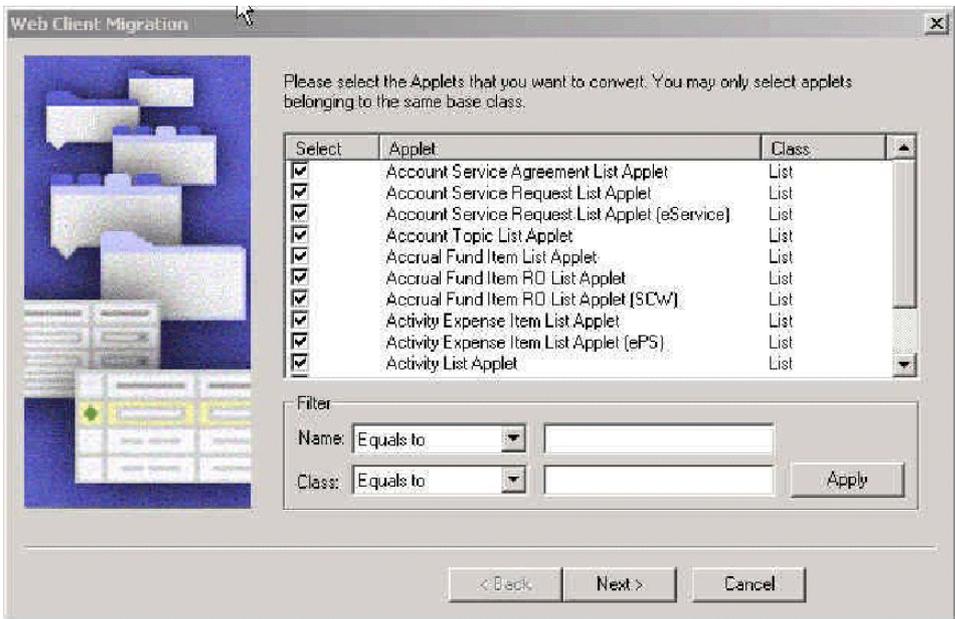
When you select applets for migration, they must be in the same base class. When you select views, they must be of the same type.

You can run the Web Layout Wizard on applets and views multiple times.

For more information on migrating created or modified applets and views to the Siebel Web Client, see [“About Migrating to the Siebel Web Client” on page 203](#).

To run the Web Layout Wizard

- 1 For applets, identify the desired model applet you want to use. For views, identify the desired model view.
You may need to modify a model applet or model view to obtain the desired characteristics.
- 2 In Siebel Tools list view, select the applets or views you want to migrate.
- 3 In the list view, right-click and select Web Layout Wizard.



- 4 In the Web Client Migration dialog box, verify that all the applets or views you want to migrate are selected. Click Next.
- 5 Select the model applet or model view you identified.
- 6 Click Next to migrate the applets or views.
- 7 Review the conversion results. Repeat the procedure if needed.

Migrating Scripts Attached to Controls

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

In Release 7.0.x, the ControlName_Click event handler was replaced by the Method Invoked property on control objects. By specifying the Method Invoked property, when a user clicks on a control (for example, a Button), the client framework checks to see if the method was implemented in browser script associated with the Applet_PreInvokeMethod event.

If the method was not implemented in browser script, the request is sent to the server for execution. After this request is on the server, the Object Manager verifies that the method was implemented in the WebApplet_PreInvokeMethod event, and then the BusinessService_PreInvokeMethod event.

If the method was not implemented in either browser or server script, an error occurs.

Control methods that were formerly available in Siebel VB or Siebel eScript also are no longer available. In Release 8.0, all control methods are available through browser scripting and execute in the high interactivity applications only. [Table 16](#) provides a list of the control methods that are supported by browser script. For more information on control methods, see *Siebel Tools Online Help*.

Table 16. Control Method Syntax Summary

Method	Description	Syntax
Applet	Returns the parent applet for the control.	var oControl; var oApplet; oApplet = oControl.Applet();
Buscomp	Returns the corresponding business component for the control.	var oControl; var busComp; busComp = oControl.Buscomp();
GetValue	Returns the value of a control.	var oControl; var sValue; sValue = oControl.GetValue();
Name	Returns the name of the control.	var oControl; var sName; sName = oControl.Name();
SetValue	Sets the contents of the control to the indicated value.	var oControl; oControl.SetValue(<i>value</i>);

In addition to specifying the Method Invoked property, controls may also be scripted based upon native DOM events supported by the browser.

Figure 6 depicts the DOM events that may be scripted when a MiniButton control is added to an applet. The scripting of control-based DOM events is supported by high and standard interactivity applications and must be implemented in JavaScript.

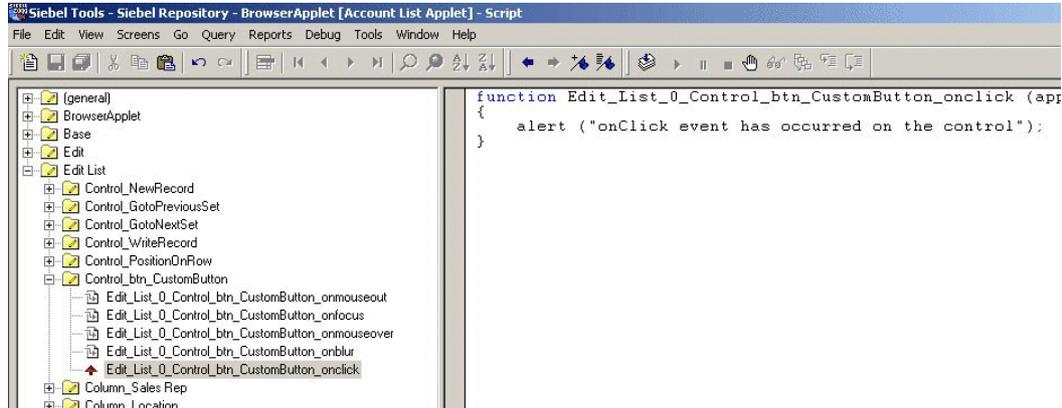


Figure 6. DOM Events When MiniButton Control Added to Applet

For more information on migrating scripts, see [“About Migrating 6.2.1 Scripts”](#) on page 209.

Related Topics

[“Migrating Scripts Attached to Applets”](#)

[“Migrating Business Component, Business Service, and Application Scripts”](#) on page 223

Migrating Scripts Attached to Applets

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

When upgrading to Release 8.0, you must do one of the following:

- Migrate Release 6.2.1 scripts written on applets to applet server scripts or business component server scripts. Migrating applet scripts includes the following types of tasks:
 - Moving Release 6.2.1 applet scripts such as Control_Click event handlers to business components
 - Rescripting Release 6.2.1 ActiveX controls in browser script
 - Rewriting other Release 6.2.1 applet scripts in browser script
- Rewrite Release 6.2.1 scripts as browser scripts.

To identify the number of 6.2.1 applet scripts to be reviewed or migrated, look at the Object List Editor in Flat mode in Siebel Tools.

As shown in Figure 7, applet scripts implemented in Siebel 6.2.1 need to be moved to one or more of the following browser or server script events:

- (Web) Applet Server Script:
 - PreInvokeMethod
 - InvokeMethod
 - ShowControl
 - ShowListColumn
 - PreCanInvoke
 - Load
- Applet Browser Script:
 - PreInvokeMethod
 - InvokeMethod
 - ChangeRecord
 - ChangeFieldValue
- Application Server Script:
 - PreNavigate
 - Navigate

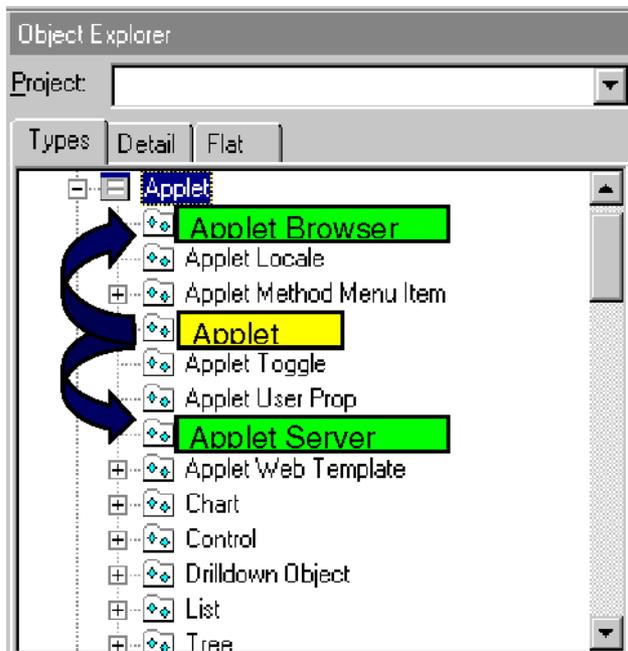


Figure 7. Movement of Applet Scripts

In Release 8.0, the WebApplet_ShowControl and WebApplet_ShowListColumn event handlers are only supported for standard interactivity applications.

In many cases, the migration of applet scripts is straightforward, because several WebApplet events correspond to Siebel 6.2.1 applet events. Corresponding browser or server script event handlers might be available, but the scripts are not automatically migrated. Table 17 depicts Siebel 6.2.1 events and corresponding Release 7.x events available in either browser or server script.

Table 17. Siebel 6.2.1 Applet Events and Release 7.x Equivalents

Siebel 6.2.1 Applet Event	Release 7.x Equivalent
Applet_Load	WebApplet_Load (Server Event)
Applet_GotFocus	WebApplet_Load (Server Event)
Applet_ChangeFieldValue	Applet_ChangeFieldValue (Browser Event)
Applet_ChangeRecord	Applet_ChangeRecord (Browser Event)
Applet_PreInvokeMethod	WebApplet_PreInvokeMethod (Browser or Server Event)
Applet_InvokeMethod	WebApplet_InvokeMethod (Browser or Server Event)

In cases where there is no direct correlation between Siebel 6.2.1 events and Release 8.0 events (for example Applet_LostFocus event), you must evaluate and reimplement the functionality of the Siebel 6.2.1 script. Depending upon the functionality provided by the script, scripts on the Applet_LostFocus event could be reimplemented as a server script and implemented on the TheApplication.Navigate or PreNavigate event.

Some applet level methods are no longer available in Release 8.0. The obsolete applet methods are listed in Table 18. In some cases, the methods were reimplemented as browser script methods, as in the case of FindControl and FindActiveX Control. Other methods, such as PostChanges, were replaced by properties that can be configured in Siebel Tools (Immediate Post Changes property on a Field). A few other methods require modifications to scripts that use these methods.

Table 18. Obsolete Applet Methods

Applet Method
ActiveControl
FindControl
FindActiveXControl
GotoControl
PostChanges
Drilldown

For more information on migrating scripts, see [“About Migrating 6.2.1 Scripts” on page 209](#).

Related Topics

[“Migrating Scripts Attached to Controls” on page 218](#)

[“Migrating Business Component, Business Service, and Application Scripts”](#)

Migrating Business Component, Business Service, and Application Scripts

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

The majority of application, business components, and business services scripts implemented in Siebel 6.2.1 can remain unchanged and do not require any modification after the upgrade. If existing scripts refer to methods that became obsolete or contain references to the Siebel user interface, the scripts need to be edited.

The following application and business component methods became obsolete as of Release 7.x:

- Application.MsgBox
- Application.InputBox
- Application.ActiveBusComp
- Application.ActiveApplet
- Application.GotoApplet
- Application.ShowStatus
- Application.ActiveControl
- Application.FindApplet
- BusComp.AllowNotification
- BusComp.SuppressNotification

In many cases, Release 7.x implementation alternatives offer comparable functionality to the obsolete methods. For example, the ActiveBusComp method can typically be replaced with a combination of ActiveBusObject and GetBusComp. By using these two methods together in a script, you can get access to the active business component. Additionally, MsgBox and InputBox methods can typically be replaced with some core functions available in JavaScript (through browser script) and some methods introduced in Release 7.x.

For example, using browser script, you can use code JavaScript features to provide warnings, dialog boxes, or input boxes through the alert, confirm, and prompt functions. For more information about using JavaScript features available in browser script to replace MsgBox and InputBox, see 476612.1 (Article ID) on My Oracle Support. This document was previously published as Siebel FAQ 1562.

The RaiseError and RaiseErrorText methods were introduced to provide an alert notification from server script. These methods allow an error message to be displayed as an alert in the Siebel Web Client. Lines of code that follow the RaiseError or RaiseErrorText methods are not executed.

For more information about using RaiseError and RaiseErrorText methods, as well as additional information about using browser script and server script in Release 8.0, see *Siebel Object Interfaces Reference* and 476879.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 386.

For more information about migrating scripts, see [“About Migrating 6.2.1 Scripts” on page 209](#).

Related Topics

[“Migrating Scripts Attached to Controls” on page 218](#)

[“Migrating Scripts Attached to Applets” on page 220](#)

Migrating Outbound COM Interfaces

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

Client-side application integration in the Siebel Web Client can be accomplished by using two methods provided by Jscript.Net, Microsoft’s implementation of the ECMA 262 language:

- **ActiveXObject.** Enables and returns a reference to an Automation object
- **GetObject.** Returns a reference to an Automation object from a file

For additional information on the ActiveXObject and GetObject functions, see the Microsoft Web site (<http://www.microsoft.com>) and the Jscript.Net documentation.

The following example illustrates using the ActiveXObject function to access properties and methods of Microsoft Excel, including the Application object and the ActiveSheet.

```
var Excel App;  
  
var Sheet;  
  
Excel App = new ActiveXObject("Excel . Appl i cati on");  
Sheet = new ActiveXObject("Excel . Sheet");  
  
// Make Excel vi si bl e  
Sheet . Appl i cati on . Vi si bl e = true;  
  
// Place some text in the first cell
```

```

Sheet.ActiveSheet.Cells(1,1).Value = "This is row 1 column A";

// Save the sheet.

Sheet.SaveAs("C:\\TEST.XLS");

// Close Excel and quit.

Sheet.Application.Quit();

```

In addition to the functions described above, you can use native functions provided by Siebel VB or Siebel eScript to handle client-side application integration:

- Siebel VB:
 - CreateObject (to access a client-side automation object)
 - Declare (to access an external DLL)
- Siebel eScript:
 - COMCreateObject (to access a client-side automation object)
 - SELib.DynamicLink (to access an external DLL)

For additional information on the Siebel VB and Siebel eScript functions listed above, see *Siebel Tools Online Help*.

For more information about migrating client-side interfaces, see [“About Migrating 6.2.1 Client-Side Interfaces” on page 210](#).

Related Topic

[“Migrating Inbound COM Interfaces”](#)

Migrating Inbound COM Interfaces

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

Release 8.0 external applications can interact with the Siebel client through the Siebel Web Client Automation Server. The Siebel Web Client Automation Server is similar in functionality to the COM Automation Server available in Siebel 6.2.1, but it does not expose applet or control objects.

The Release 8.0 Siebel Web Client Automation Server enables external applications to interact with the Siebel Web Client. The Web Client Automation Server enables external COM-based applications to invoke Siebel business services and manipulate property sets. The Siebel Web Client Automation Server is implemented as a COM object resident within the browser.

To enable the Siebel Web Client Automation Server, in the [SWE] section of your application's configuration file, make sure that the `EnableWebClientAutomation` parameter is set to `TRUE`. When this parameter is set to `TRUE`, when the application is run and a control is required, an ActiveX control is downloaded to the desktop. This process terminates when the Siebel Web Client is terminated normally.

The following example illustrates how a small Microsoft Visual Basic application that uses the Siebel Web Client Automation Server and the EAI Siebel Adapter Business Service queries for a list of Accounts that start with the letter A, and displays the result in a message box.

```
Private Sub Command1_Click()
    Dim siebelApp As SiebelHTMLApplication
    Dim inPs As SiebelPropertySet, siebelMsg As SiebelPropertySet
    Dim listofAcct As SiebelPropertySet, acctPs As SiebelPropertySet
    Dim outputPs As SiebelPropertySet
    Dim svc As SiebelService
    Dim i As Long, j As Long, iPos As Long
    Dim acctList As String
    Dim status As Boolean
    Set siebelApp = GetObject("", "SiebelHTML.SiebelHTMLApplication.1")
    Set inPs = siebelApp.NewPropertySet
    Set siebelMsg = siebelApp.NewPropertySet
    Set listofAcct = siebelApp.NewPropertySet
    Set acctPs = siebelApp.NewPropertySet
    Set outputPs = siebelApp.NewPropertySet
    Set svc = siebelApp.GetService("EAI Siebel Adapter")
```

' The following lines will construct a property set to query on the Sample Account

' Integration Object

```
siebelMsg.SetType ("Siebel Message")
status = siebelMsg.SetProperty("IntObjectName", "Sample Account")
status = siebelMsg.SetProperty("MessageId", "")
status = siebelMsg.SetProperty("MessageType", "")
```

```

ListOfAcct.SetType ("ListOfSample Account")

acctPs.SetType ("Account")
status = acctPs.SetProperty("Name", "A*")

iPos = ListOfAcct.AddChild(acctPs)
iPos = siebelMsg.AddChild(ListOfAcct)
iPos = inPs.AddChild(siebelMsg)

' Now that the Integration Object has been constructed, query for Accounts
starting with A
status = svc.InvokeMethod("Query", inPs, outputPs)

If status then
j = 0
i = outputPs.GetChild(0).GetChild(0).GetChild(0).GetChild(0).GetPropertyCount

Do While j < i
    acctList = acctList & outputPs.GetChild(0).GetChild(0).GetChild(0). _
        GetChild(j).GetProperty("Name") & Chr(10) & Chr(13)
    j = j + 1
Loop
MsgBox (acctList)
End If

End Sub

```

For additional information on the Siebel Mobile Web Client Automation Server or the Siebel Web Client Automation Server, see *Siebel Tools Online Help*.

You might need to modify browser security settings to allow use of the Siebel Web Client Automation Server. For security settings and information about deploying ActiveX controls in the browser, see *Siebel System Administration Guide*.

For more information about migrating client-side interfaces, see [“About Migrating 6.2.1 Client-Side Interfaces” on page 210](#).

Related Topic

[“Migrating Outbound COM Interfaces” on page 224](#)

Regenerating the Postmerge Utilities Log

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

When upgrading from Release 6.2.1 to Release 8.0, you perform the repository merge and then the Siebel Web Client migration. This invalidates part of the postmerge utilities log created after the repository merge. You must regenerate the postmerge utilities log before using it to evaluate the merge.

After the repository merge, the postmerge utilities start automatically. They help resolve common repository merge problems related to configuration and layout.

These utilities write to the postmerge utilities log:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\repreports.log
```

Requirements: You must have run the repository merge and the Siebel Web Client Migration Wizard.

To regenerate the postmerge utilities log

- 1 In Siebel Tools, click Tools in the menu bar.
- 2 In the Upgrade submenu, click Upgrade Application.
- 3 In the Merge Repositories dialog box, set up the upgrade in exactly the same way as the actual upgrade, including the advanced settings.
- 4 Click Reports.
- 5 When the process completes, review the log.

Reviewing Customized Business Components

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

After upgrading the Siebel database to Release 8.0, thoroughly review the postupgrade configuration to make sure that the object level definitions are preserved as expected.

The access control business component migration utility, `upgcust`, is run on the Prior Customer Repository by the Upgrade Wizard during upgrade of the Siebel database schema. This utility reconfigures custom business component configurations on obsolete tables. However, you may need to perform manual steps to business component fields and joins, depending upon the complexity of your business component configuration.

In some cases, not all custom business components are reconfigured by the access control business component migration utility. For example, customer business components that are based on customer extension columns or on Siebel columns in obsolete tables need to be identified and manually fixed after the upgrade.

During the upgrade, the access control business component migration utility, `upgcust`, generates a log file titled `upgcust.log` that contains a list of business component joins and fields that need to be manually rectified. This particular log file, along with others generated by the upgrade process, can be found under `SIEBEL_ROOT\log` (Windows).

Log File Part 1

For extension columns on obsolete tables such as `S_EMPLOYEE` and `S_ORG_INT`, you need to reimplement the extension columns on the replacement tables. After you have done this, review the business component definition to verify proper operation.

The report generated by part 1 of the log file provides a list of the business component fields that are based on custom extension columns in obsolete tables such as `S_EMPLOYEE` and `S_ORG_INT`. This list displays the following properties:

- Business Component Name
- Field Name
- Column Name

Upgrades from Release 7.0.x and 7.5.x

The table `S_EMPLOYEE` is migrated to three tables—`S_CONTACT`, `S_EMP_PER`, and `S_USER`. After you have determined and implemented the approach for previously defined custom extension columns on obsolete tables, manually configure the business component field to reference that database column. Perform any manual reconfiguration in the New Customer Repository after the upgrade has been run.

Log File Part 2

After the repository merge has been run, there may be inconsistencies in the join names and joins set at the field level due to merge behavior. If you do not resolve these discrepancies, the application configuration may result in errors or result in incorrect behavior. The report generated by part 2 of the log file provides a list of joins that were not updated during the merge process. This list displays the business component name and the join name.

With this list, you must go to each of the Business Component definitions and manually change the join name from the current value to the value listed in the report. Make sure that there is consistency between the joins as defined by name and the joins defined for each of the fields. Again, perform any manual configuration in the New Customer Repository after the upgrade has been run. For more information about working with Business Component definitions and joins, see *Siebel Tools Online Help*.

16 Performing Post-Upgrade Tasks On the Target Database

This chapter describes tasks you must perform after upgrading the target database. It includes the following topics:

- [Transferring the Development Environment Upgrade Output Files to the z/OS Host on page 231](#)
- [Synchronizing the Schema on page 232](#)
- [Migrating Custom Business Component Configurations on page 233](#)
- [Performing Intersection Table Maintenance on page 234](#)
- [Installing New License Keys After Upgrade on page 237](#)
- [Deleting Redundant Upgrade Files on page 237](#)

Transferring the Development Environment Upgrade Output Files to the z/OS Host

Upgrades: All upgrades.

Environments: Development environment only.

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

The development upgrade upgphys process generates the following files, which are used to complete the development upgrade:

- **synctab.sql and syncidx.sql:** Synchronize tables and indexes on the development database
- **dedup.jcl and dedup_prod.jcl:** Remove duplicate records from the intersection tables and sets up support for database aggregation

Transfer these files to the z/OS host using the following procedure.

To transfer the development environment upgrade output files to the z/OS host

- 1 Navigate to the \DB2390\dbsrvr\dboutput\upgrade directory (Windows) or the /DB2390/dbsrvr/dboutput/upgrade directory (UNIX) and double-click the ftp_syncdd.bat file (Windows) or issue the following command (UNIX):

```
Ftp - i > ftp_syncdd.bat
```

- 2 Enter your TSO ID and password and press Enter.

All the development environment post-upgrade files are transferred from the midtier to the z/OS host.

- 3 Review the ftp_syncdd.log file which is created in the upgrade directory and verify that all the files listed in the ftp_syncdd.txt file transferred successfully to z/OS staging datasets.

NOTE: If the development and production upgrades are run on different mid-tier machines, then you must copy the files to be transferred to the z/OS host to the production mid-tier machine before running the ftp_syncdd.bat file.

About the Dedup Files

The dedup files and dedup_prod.jcl files generated by the upgphys process are similar to the files that you generate during the Prepare for Production Upgrade process, which is run before starting the production environment upgrade. Since both processes use the same information when generating these files, you can use the dedup files created by either the upgphys process or the Prepare for Production Upgrade process.

Synchronizing the Schema

Upgrades: All upgrades.

Environments: Development environment only.

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

After completing the development environment target database upgrade, you must perform the following procedure to synchronize database tables and indexes.

To synchronize the schema

- 1 Enter the following command and press Enter:

```
TSO SBLUPG80
```

The Siebel In-Place Upgrade 8.0 Main Menu appears. The panel ID is SBLUPG8P.

- 2 Select option 4: Post-Upgrade, and press Enter.

The Target Database Post-Upgrade Menu appears. The panel ID is SBLPSTP.

- 3 Select option 0: Apply/Run Table Synchronization, and press Enter.

You are placed in edit mode for the dataset `&DSNHLQ.SI EBEL. I NSTALL. JCL(SUBSYNCT)`.

- 4 Run the job using the JCL in the dataset.

This job submits the SYNCTAB (synchronize table) job which executes a file containing the DDL for the tables.

- 5 Verify that the job ran successfully.

Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.

- 6 After submitting the job, enter cancel on the command line or press PF3 to save changes.

- 7 On the Target Database Post-Upgrade Menu, select option 1: Apply/Run Index Synchronization, and press Enter.
You are placed in edit mode for the dataset `&DSNHLQ.SIEBEL.INSTALL.JCL(SUBSYNCX)`.
- 8 Run the job using the JCL in the dataset.
This job submits the SYNCIDX (synchronize index) job which executes a file containing the DDL for the indexes.
- 9 Verify that the job ran successfully.
Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 10 Press PF3 to return to the Upgrade In-Place Main Menu.
Upgphys processing is now completed.

Migrating Custom Business Component Configurations

Upgrades: Release 6.2.1 of Siebel Financial Services applications.

Environments: Development environment only.

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

Customized business component configurations (buscomps) which are based on customer extension columns or on Siebel columns in obsolete tables need to be identified and manually fixed after the upgrade. Siebel columns in obsolete tables are reconfigured and migrated during the upgrade. However, customer business components or custom extension columns on obsolete tables need to be migrated manually.

NOTE: Custom extension columns on tables that are upgraded (not obsolete) are retained during the upgrade.

To review a list of the business components that require manual migration, review the upgcust.log file, located in the `SI EBSRVRI` log directory.

Critical obsolete tables are listed in [Table 19](#).

Table 19. Repository Tables That Are Obsolete in Release 7

Previous Table	Suggested New Table
S_EMPLOYEE	S_CONTACT, S_USER, S_EMP_PER
S_EMP_POSTN	S_PARTY_PER
S_ORG_INT	S_ORG_EXT, S_BU
S_POSTN_RPT_REL	S_PARTY_RPT_REL

The access control business component migration utility is run on the Prior Customer Repository by the Siebel Upgrade Wizard during upgrade of the Siebel database schema.

Several fields, business components, and columns may need to be reevaluated and recreated after your upgrade. A sample conversion script is available for migration of data to the new columns during the development upgrade.

Performing Intersection Table Maintenance

Upgrades: All upgrades.

Environments: Production test, production.

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

The intersection table maintenance jobs (ITM) identify and resolve potential duplicate row IDs from intersection tables. Perform the following procedures to prepare and run the ITM jobs.

To prepare the intersection table maintenance jobs

- 1 Enter the following command and press Enter:

```
TSO SBLUPG80
```

The Siebel In-Place Upgrade 8.0 Main Menu appears. The panel ID is SBLUPG8P.

- 2 Select option 4: Post-Upgrade, and press Enter.

The Target Database Post-Upgrade Menu appears. The panel ID is SBLPSTP.

- 3 Select option 2: Copy ITM files from Development HLQ, and press Enter. This process copies the ITM development upgrade files and then converts them for use in the production upgrade.

A message appears prompting you to enter the Dataset Name High-Level Qualifier (DSNHLO) used for the ITM/Dedup datasets produced during the development upgrade.

- 4 Enter the DSNHLO used for the development upgrade ITM/Dedup datasets, and press Enter.

If the DSNHLO you enter is valid, the following message appears:

```
Press Enter and submit JCL to complete required changes. . . .
```

- 5 Run the JCL in dataset *&DSNHLO.SIEBEL.INSTALL.JCL(ITMCONV)*.

The datasets are copied from the development environment and renamed to use the production upgrade DSNHLO.

If the DSNHLO and associated datasets do not exist, or if you enter an invalid development upgrade DSNHLO, the process stops. Correct the problem before running the process again.

- 6 Verify that the job ran successfully.
Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 7 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 8 On the Target Database Post-Upgrade Menu, select option 3: Populate PDS Datasets, and press Enter.
- 9 You are placed in edit mode in the dataset *&DSNHLQ.SIEBEL.INSTALL.JCL(UNPACK04)*. Submit the JCL in the dataset to populate the PDS datasets.
- 10 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 11 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 12 On the Target Database Post-Upgrade Menu, select option 4: Add Jobcards to Unload & Delete/Update Jobs, and press Enter.
- 13 You are prompted to enter a three-character jobname prefix. Enter the prefix you want to use, and press Enter.

Messages are displayed indicating that the JCL generation is in process, then a series of messages indicate the number of jobs generated. The number of ITM Unload jobs that are generated varies according to your upgrade path.

The jobcards are added to the JCL templates.

Perform the following procedure to run the intersection table maintenance jobs.

To run the intersection table maintenance jobs

- 1 On the Target Database Post-Upgrade Menu, select option 5: Intersection Table Maintenance, and press Enter.
The Intersection Table Maint. Process Menu appears. The panel ID is SBLITMP.
- 2 Select option 0:Load Siebel Log Entries (1), and press Enter.
- 3 You are placed in edit mode in the dataset *&DSNHLQ.SIEBEL.INSTALL.JCL(LOADITM)*. Submit the JCL in the dataset.
- 4 Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 5 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 6 On the Intersection Table Maint. Process Menu, select option 1: Run Unload Jobs to Identify Candidate Tables, and press Enter.

- 7** You are placed in edit mode in the dataset `&DSNHLQ.SIEBEL.INSTALL.JCL(SUBITMUL)`. Submit the JCL in the dataset.

This job submits all the ITM Unload jobs to run in parallel. The number of jobs that are run varies according to your upgrade path.
- 8** Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 9** After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 10** On the Intersection Table Maint. Process Menu, select option 2: Generate Delete/Update Jobs for Candidates Identified, and press Enter:
 - If no duplicate row IDs are identified in the intersection tables, a message appears indicating that the ITM process is completed. You can proceed to the next step in the post-upgrade process.
 - If tables are identified that contain duplicate row IDs, you are prompted to enter the COMMIT frequency.
- 11** Enter the COMMIT frequency you want to use or press Enter to accept the default value of 100.

Messages appear indicating that the input Unload PDS dataset is being read and the update and delete jobs are being built.
- 12** On the Intersection Table Maint. Process Menu, select option 3: Load Siebel Log Entries (2), and press Enter.
- 13** You are placed in edit mode in the dataset `&DSNHLQ.SIEBEL.INSTALL.JCL(LOADITM2)`. Submit the JCL in the dataset.
- 14** Verify that the job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 15** After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 16** On the Intersection Table Maint. Process Menu, select option 4: Run Delete/Update Jobs for Candidates Identified, and press Enter.
- 17** You are placed in edit mode in the dataset `&DSNHLQ.SIEBEL.ITMJOB.SJCL`.

The PDS member list is displayed.
- 18** Submit each member in the dataset by selecting the PDS member, entering submit on the command line, and pressing Enter. Repeat for each member.
- 19** Verify that each job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 or RC=4 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.
- 20** After submitting the job, press PF3 to return to the In-Place Upgrade 8.0 Main Menu.

Installing New License Keys After Upgrade

Upgrades: All upgrades.

Environments: All environments.

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

With the new release of Siebel Business Applications, you receive one or more license keys. You must add all of the new license keys to enable the new release of Siebel Business Applications.

To add new license keys

- 1 Start the new-release version of Siebel Tools
- 2 Connect to the database server as the Siebel administrator.
Use SADMIN as the login name and SADMIN as the password, and choose Server in the Connect to drop-down list.
- 3 Click Add Key.
- 4 In the License Key dialog box, enter the Siebel Business base application license key or the Siebel Tools license key, and then click OK.
- 5 Close Siebel Tools.

For more information on installing license keys, see *Siebel Installation Guide* for the operating system you are using and *Siebel Applications Administration Guide*.

Deleting Redundant Upgrade Files

Upgrades: All upgrades.

Environments: All environments.

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

After you complete the database upgrade process, delete any upgrade files that are no longer required. The procedures in this topic describe how to drop all staging database objects, and how to delete Unload datasets and stored procedures.

Dropping the Staging Database

After you have successfully completed your database upgrade, you must drop the staging database objects before you can reuse the same database prefix and tableowner name.

To drop the staging database

- 1 Log on to the z/OS host.

- 2 Submit the JCL in the `&DSNHLQ.SIEBEL.INSTALL.JCL(STGDROPJ)` dataset.
The staging database objects are dropped.
- 3 Press PF3 to complete the process.

Deleting Unload Datasets

When your database upgrade is completed, you no longer need the Unload datasets. Perform the following procedure to generate a list of the Unload datasets and then delete them.

NOTE: Before performing this procedure, you must have the `*.siebel.exec` library allocated.

To delete Unload datasets

- 1 Log on to the z/OS host.
- 2 Enter the following command and press Enter:

```
TSO SBLDELDS
```

A list of the unload datasets to be deleted is generated.
- 3 Navigate to the `*.siebel.install.jcl` library.
- 4 Submit the JCL in the dataset `&DSNHLQ.SIEBEL.INSTALL.JCL(DELULDS)` to delete the Unload datasets.
- 5 Verify that each job ran successfully. Review the output in SDSF or another job output facility. Verify that the RC=0 and that the JCLTEST return code=FLUSH. If you do not see the FLUSH return code, you can verify the condition codes by searching for the condition code IEF206I.

Deleting Stored Procedures

A number of stored procedures are installed during the database upgrade on the z/OS host to facilitate upgrade processing. These stored procedures are not required after the upgrade has been completed successfully and can be deleted. The following procedure describes how to drop the upgrade stored procedures.

For information on installing the stored procedures, see [“Creating and Deploying Stored Procedures on the Target Database” on page 176](#).

To drop the upgrade stored procedures

- 1 Log on to the z/OS host.
- 2 Submit the JCL in the dataset member `DSNHLQ.SIEBEL.SP.CNTL(DRPSIA)` or `DSNHLQ.SIEBEL.SP.CNTL(DRPHOR)`, depending on your upgrade path.
The Stored Procedures are deleted.
- 3 Press PF3 to complete the process.

17 Reviewing the Siebel Upgrade Log Files

This chapter describes the upgrade log files that the Siebel Upgrade Wizard produces on the midtier during the upgrade file generation process. It also describes how to check the status of staging and target upgrade jobs on the z/OS host using the z/OS Siebel job logs. This chapter includes the following topics:

- [About the Siebel Upgrade Log Files on page 239](#)
- [Reviewing Siebel Upgrade Log Files for Errors on page 241](#)
- [Manually Archiving Upgrade Log Files on page 242](#)
- [Viewing the Siebel Job Log Status on page 242](#)
- [Running SQL in Siebel Logs on page 243](#)

About the Siebel Upgrade Log Files

The Upgrade Wizard writes logs that provide detailed information on the upgrade processes and they also list all errors. The Upgrade Wizard writes the logs for a process to the following directory by default:

Windows: `SIEBEL_ROOT\LOG\process`

UNIX: `$SIEBEL_ROOT/log/process`

where *process* is the name of the upgrade process you have run, for example, `upgrep_dev_704` or `prepare_for_production_upgrade`.

The *process* directory contains the following subdirectories:

- **Output.** Directory containing the Upgrade Wizard log files
- **State.** Directory containing the state.log file

The output and state 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.

NOTE: You can select a different log directory from the Log Output Directory screen on the Database Configuration Wizard.

About the State Log File

Each upgrade process consists of a series of steps, each of which must complete successfully. If the Upgrade Wizard cannot complete a step, it marks the step as incomplete in the state.log file and exits. The state.log file is located in `SIEBEL_ROOT\LOG\process\state` (Windows) or `SIEBEL_ROOT/LOG/process/state` (UNIX).

You must correct the error and then 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.

About Process Log Files

You can identify errors you encounter during an upgrade by reviewing the log file named UpgWiz.log (Windows) or srvrupgwiz1.log (UNIX) in the *SIEBEL_ROOT\LOG\process\output* directory (Windows) or the *SIEBEL_ROOT/LOG/process/output* directory (UNIX).

The name of the log file increments for subsequent log files that are created if the Siebel Upgrade Wizard encounters a problem and you run the Siebel Upgrade Wizard again.

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

How to Determine if the Upgrade Process Completed Successfully

If the status of all the steps in the state.log is Complete, the upgrade process completed successfully.

If the status of any step is Incomplete, the upgrade process did not complete successfully. You must identify the error and correct it before resuming the upgrade.

NOTE: 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, even those with a status of Complete.

Use the following process to identify errors:

- 1 Resolve errors for steps identified with a Status of Incomplete in the state.log file.
- 2 Review all the steps with a status of Complete in the state.log file. If any contain unacceptable errors, resolve these errors. See [“Reviewing Siebel Upgrade Log Files for Errors” on page 241](#) for information on identifying unacceptable upgrade 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, create a service request (SR) on My Oracle Support.

Log Files That Can Be Ignored

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

- Windows: sw_cfg_xxx.log and siebel.log
- UNIX: srvrupgwiz_*.log and siebel_*.log. For example, srvrupgwiz_001.log, and srvrupgwiz1_02.log
- Any other log file that existed before the start of the upgrade

CAUTION: UNIX Only: The log file *srvrupgwiz_001.log* is a different file than *srvrupgwiz1.log*. Do not ignore log files named *srvrupgwiz1.log*, *srvrupgwiz1_01.log* and so on.

Reviewing Siebel Upgrade Log Files for Errors

Upgrades: All upgrades.

Environments: All environments.

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

Review the logs created when you run the Siebel Upgrade Wizard to verify that the upgrade process completed correctly and to identify errors that must be resolved. The log files may include errors that are expected and benign. You must compare any error messages found in the log files to a list of acceptable error messages, and correct any non-benign errors.

Complete the following procedure to manually review log files for unacceptable errors.

To manually review the log files for unacceptable errors

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

Windows: `SI EBEL_ROOT\LOG\process\state`

UNIX: `$SI EBEL_ROOT/LOG/process/state`

- 2 Print the errors file. The errors file lists the benign and expected errors you might find in the log files; you can ignore these errors. The errors file is located in the installation subdirectory:

Windows: `DBSRVR_ROOT\DB2390\errors.rtf` or `errors.htm`

UNIX: `DBSRVR_ROOT/DB2390/errors.txt`

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

Windows: `SI EBEL_ROOT\LOG\process\output`

UNIX: `$SI EBEL_ROOT/LOG/process/output`

- 4 Open each log file, starting with the earliest, and search for errors. Starting with the earliest log file can shorten your research time.

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

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

The log files generated by the Siebel Upgrade Wizard (for example `svrupgwiz1.log`) appear in the errors file as `upgwiz1.log`, `upgwiz2.log`, incrementing for additional log files. Identify errors as follows:

- If you find the error in the errors file, it is acceptable and no action is required. Continue to review the errors found in the log file.

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

To help resolve unacceptable errors, view 477324.1 (Article ID) on My Oracle Support. This document was formerly published as Siebel Troubleshooting Steps 21. Common errors and how to resolve them are listed in this document. If the error is not listed or you cannot resolve it, create a service request (SR) on My Oracle Support. *Do not proceed with the upgrade.*

6 Repeat [Step 5](#) for each log file.

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

Related Topic

[“About the Siebel Upgrade Log Files” on page 239](#)

Manually Archiving Upgrade Log Files

Upgrades: All upgrades.

Environments: All environments.

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

After a successful installation and upgrade, you must manually save and archive the log files located in the *SIEBEL_ROOT/LOG/process* (Windows) directory.

By default, only nine (9) upgrade log files are retained for subsequent retries of the Siebel Upgrade Wizard. After nine log files have been created, when the Siebel Upgrade Wizard is rerun, it overwrites log files beginning with the earliest one created and recycles the rest as necessary. (This does not apply to the state.log file.)

The number of log files retained can be increased by resetting the *siebel_log_archive* environment variable to 20—for example, to retain twenty (20) log files.

Viewing the Siebel Job Log Status

Upgrades: All upgrades.

Environments: All environments.

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

Whether you are using Siebel-scheduling or vendor-scheduling to run your upgrade jobs, you can query the Siebel job log for the staging and target upgrade processes by completing the following procedure.

NOTE: To view job status, you must have installed DSN REXX.

To view the Siebel job log status

- 1 If you are not on the Siebel In-Place Upgrade 8.0 Main Menu, enter the following command:
 TSO SBLUPG80
- 2 Select one of the following options, and press Enter:
 - Option 5: Staging Database Joblog (SBLLOG S).
 The Staging Joblog Query panel displays.
 - Option 6: Target Database Joblog (SBLLOG T).
 The Target Joblog Query panel displays.

The panel ID of both the staging and target joblog panels is SBLLOGP.
 A list of successful and failed jobs appears.
- 3 Next to the label List By Job Status: , enter 1 to list failed jobs and enter 2 to list jobs that have not yet been run.

The list displays 250 lines only. The Unload, Load and Index Rebuild jobs have more than 250 jobs, so you must query using another option or by specific or partial job name.
- 4 Press PF3 when you are finished viewing the log.

The Siebel In-Place Upgrade 8.0 Main Menu for your upgrade path appears.

Running SQL in Siebel Logs

By using SPUFI or the command line, you can construct SQL queries to run against the staging or the target log tables.

The following statements report the status of the load jobs for the staging or target databases:

- `SELECT JOB_DESC, JOB_NAME, JOB_STATUS FROM CQ10A901.TMP_SBLLOG_STG WHERE JOB_NAME LIKE ' LKC%';`
- `SELECT JOB_DESC, JOB_NAME, JOB_STATUS FROM CQ10A901.TMP_SBLLOG_TAR WHERE JOB_NAME LIKE ' LKC%';`

You can alter the preceding statements to report the status of any jobs by changing the JOB_NAME LIKE statement to another prefix.

This following statement checks for failed unload jobs, but can check for any other job by changing the JOB_NAME LIKE statement to use the appropriate prefix.

```
SELECT JOB_DESC, JOB_NAME, JOB_STATUS FROM CQ10K034.TMP_SBLLOG_STG WHERE JOB_STATUS  
!= 'COMPLETED SUCCESSFUL' AND JOB_NAME LIKE 'LKB%';
```

18 Postupgrade Tasks for a Siebel Development Environment

This chapter describes the tasks you need to perform after upgrading your DB2 for z/OS development environment. Also perform applicable tasks described in the chapter of the *Siebel Database Upgrade Guide* that describes the Siebel postmerge development tasks. This chapter includes the following topics:

- [Reapplying Schema Customizations to the Siebel Database on page 245](#)
- [Deleting Duplicate EIM Mappings on page 247](#)
- [Regenerating the Database Template File on page 248](#)

Reapplying Schema Customizations to the Siebel Database

Upgrades: All upgrades.

Environments: Development environment only.

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

In the current release, several tables are obsolete or have been replaced by new tables. If you added extension columns or foreign key (FK) columns to tables that are obsolete in the current release, you may want to reapply these changes to new tables that have replaced the obsolete tables.

Reviewing Obsolete Tables

The upgrade process generates a report that you can review for information about obsolete tables that will help you decide whether or not you need to reapply schema customizations. This report, `xtndobtbl.txt`, lists the following:

- Obsolete tables in the current release
- Custom columns in obsolete tables
- Custom foreign key columns pointing to obsolete tables
- EIM mappings for custom foreign key columns to obsolete tables
- Workflow columns by custom foreign key to obsolete tables
- Custom denormalized columns to Siebel base tables that might be obsolete

Each obsolete table is listed with one of three codes:

- **Not Used.** These tables are not used in the current release, but you may continue to use them. These tables are supported as is (for instance, with docking or EIM).

- **EOL (end of life).** These tables are not used in the current release, and they are not supported in future releases.
- **Inactive.** These tables have been discontinued, and are not supported in the current release. You can choose to move extension columns and foreign key columns that reside on inactive tables to alternate tables.

If no tables are listed in `xtndobstbl.txt`, no action is required. If this file lists any tables, you can reapply their custom extensions and foreign key columns to tables in the current release using Siebel Tools. For further information on this task, see *Configuring Siebel Business Applications*.

Table 20 lists examples of previously used tables that are inactive in the current release (you can no longer use these tables) and lists the suggested new tables to which custom extensions can be reapplied. The new tables are recommendations only; the tables that you choose to apply the extensions to might vary depending on their type and use. For help with validating the reapplication of extension columns and reviewing the steps necessary to migrate any extension column data to the new tables, create a service request (SR) on My Oracle Support.

This data must be migrated during both the development and production environment upgrades.

Table 20. Examples of Tables That Are Inactive in Release 8

Inactive Table	Suggested New Table
S_EMPLOYEE	S_CONTACT, S_USER, S_EMP_PER
S_EMP_POSTN	S_PARTY_PER
S_ORG_INT	S_ORG_EXT, S_BU
S_POSTN_RPT_REL	S_PARTY_RPT_REL

If you have created many custom extension columns on the tables `S_EMPLOYEE` or `S_ORG_INT`, neither of which are used in Release 8, the joins between the tables will not be accurate. This may result in SQL errors when you launch the Siebel client.

In such cases, using Siebel Tools, you can manually create corresponding extension columns in the new target tables, and manually move the data to the new extension column on the new table before you continue migration of the application. Then review the business component configuration to make sure that the client will operate properly.

You may need to do this in one of the following instances:

- Fields based on custom extension columns in `S_EMPLOYEE` or `S_ORG_EXT`
- Fields based on custom extension tables from `S_EMPLOYEE` or `S_ORG_INT` with or without join
- Custom joins to custom extension tables from `S_EMPLOYEE` or `S_ORG_INT`

If you review the `xtndobstbl.txt` file after you run the business component migration utility, you will find a list of fields that require your attention.

Table 21 lists examples of previously used tables that are no longer used in Release 8.

Table 21. Examples of Tables That Are Not Used In the Current Release

Previous Table	Suggested New Table
S_CRSE	S_SRC, S_SRC_EVT
S_CRSE_OFFR	S_SRC, S_SRC_EVT
S_CRSE_REG	S_SRC_REG
S_CTLG_CAT_REL	S_CTLG_CAT
S_OPTY_PROD	S_REVN
S_TMSHT_LINE	S_TMSHT_ITEM, S_TMSHT_LN

Table 22 lists examples of tables which were unused in previous releases of Siebel Business Applications, but are used in the current release.

Table 22. Examples of Previously Unused Tables That Are Used In the Current Release

Now Used Table	Used to be...
S_ACT_EMP	S_EVT_ACT
S_ACT_CON	S_EVT_ACT

Deleting Duplicate EIM Mappings

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

Any custom Enterprise Integration Manager (EIM) mappings that were added to *_IF tables in previous versions of Siebel Business Applications are automatically migrated to *_IF tables during the upgrade to Siebel 8.0. However, those customizations are not migrated to the EIM tables that were introduced in Siebel 7.x to replace the *_IF tables. If you choose to use the new EIM tables rather than the *_IF tables, you must manually add any customizations to them.

If custom mappings in *_IF tables are duplicated by new mappings added as part of the current release of Siebel Business Applications, errors will result when you use EIM.

The Database Configuration Wizard that you ran to upgrade the physical schema (see [“Process of Upgrading a Siebel Development Environment” on page 49](#)) generated a file called mapclash.txt. This file lists any custom EIM mappings that conflict with Siebel-provided mappings. The file contains the columns illustrated and defined in [Table 23](#).

Table 23. Mapclash.txt File Column Names and Definitions

Column Name	Definition
IT_Name	Interface table name (example: S_ACCOUNT_I F)
DT_Name	Destination table (base table) name
DC_Name	Destination column (base table column) name

Before proceeding with the upgrade, use Siebel Tools as described below to delete any custom mappings listed in this file.

To delete duplicate custom EIM mappings

- 1 Launch Siebel Tools and select Prior Customer Repository.
- 2 From the Object Explorer, choose EIM Interface Table > EIM Table Mapping.
- 3 From the EIM Table applet, select the interface table name (IT_Name), for example, S_ACCOUNT_I F, and query for the interface table.
- 4 From the EIM Table Mappings applet, choose the name assigned for the destination table (DT_Name); for example, S_ORG_EXT_X.
- 5 From the Object Explorer, choose EIM Table Mapping > Attribute Mapping.
The fourth column in this applet is the DC_Name.
- 6 Delete the duplicate mappings.

Regenerating the Database Template File

Upgrades: All upgrades.

Environments: Development environment only.

Following the upgrade, you must regenerate the SQL Anywhere template database file used by Siebel Remote. This process updates its schema to the same version as the Siebel Database Configuration Utilities. You will use the Generate New Database component from a new Siebel Server to do this.

For procedures on regenerating the SQL Anywhere Template file, refer to *Siebel Remote and Replication Manager Administration Guide*.

19 Postupgrade Tasks for Siebel Database and File System

This chapter describes the Siebel database and file system tasks you perform after upgrading to Release 8.0 on z/OS. Perform the tasks described in this chapter and the applicable tasks in the chapter of the *Siebel Database Upgrade Guide* that describes the postupgrade tasks for the Siebel database and file system. This chapter contains the following topics:

- [Updating the File System Directory on page 249](#)
- [Updating File System Attachments on page 249](#)

Updating the File System Directory

Upgrades: Release 6.2.1 only.

Environments: All environments.

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

As of Siebel 7.0, Siebel Anywhere requires that file system attachments be located in the `\att` subdirectory of the Siebel File System. If you are upgrading from Siebel 6.x, you must create this subdirectory in the Siebel File System and copy file system attachments to it.

To make file attachments accessible to Siebel Anywhere

- 1 In the existing file system structure (for example `C:\siebfile`), create an `att` subdirectory if it does not already exist.

Windows example: `C:\siebfile\att`

Creating the `att` subdirectory does not adversely affect the installation of your Siebel Server. (You install the Siebel Server at a later point.)

- 2 Copy (do not move) all files located under the `\siebfile` directory to the `\siebfile\att` directory so that all file attachments are accessible by Siebel Business Applications.
- 3 Verify that files have copied correctly to the `\siebfile\att` directory.

After this has been verified, clean up the file system.

Updating File System Attachments

Upgrades: All upgrades.

Environments: All environments.

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

Since Siebel 7.5, file names of attachments include the attachment table name. If an upgrade migrates the records in an attachment table to a new attachment table, you must run a utility to update the file system attachment names.

Perform the tasks in the sections below to update Siebel File System attachments.

Updating Attachment File Names

The task in this section applies to all upgrades.

Oracle provides a utility to update attachment file names in the Siebel File system. [Table 24](#) lists the input table names to use when you run the utility. The utility updates all files containing the table names you specify.

For example, if you specify the tables S_OLDTABLE_ATT and S_NEWTABLE_ATT, the utility updates the files system attachments by copying all attachment files containing the string S_OLDTABLE_ATT to attachment files containing the string S_NEWTABLE_ATT.

Table 24. Input Arguments for the Update Utility

Upgrade Path	Old Table	New Table
Upgrades: Release 6.2.1 only.	■ S_EMPLOYEE_ATT	■ S_CONTACT_ATT
Upgrades: Release 6.2.1 only.	■ S_ORG_INT_ATT	■ S_ACCNT_ATT
■ Upgrades from: Siebel 6.2.1 FINS. ■ Upgrades from: Siebel 7.0.x, 7.5.x, 7.7.x, & 7.8.x. Siebel Insurance only.	■ S_INSCLM_BL_ATT	■ S_INVOICE_ATT

To update file system attachments

- 1 Navigate to the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Run the following utility:

Windows: `chg_file_sys.bat OLD_TABLE NEW_TABLE "FILE_SYSTEM"`

UNIX: `chg_file_sys.ksh -s OLD_TABLE -t NEW_TABLE -f "FILE_SYSTEM"`

where:

- `OLD_TABLE` = The name of the attachment table in the release you are upgrading from. This table is obsolete in the new release.
- `NEW_TABLE` = Attachment records in `OLD_TABLE` were migrated to `NEW_TABLE` in the new release. The utility copies file system attachments containing the string `OLD_TABLE` to attachments containing the string `NEW_TABLE`.

- "*FILE_SYSTEM*" = name of the directory where the Siebel File System attachments reside (entered inside quotation marks)

Windows example:

```
chgng_file_sys.bat S_EMPLOYEE_ATT S_CONTACT_ATT
"C:\siebel\att"
```

UNIX example:

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

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

For example, since S_EMPLOYEE_ATT is migrated to S_CONTACT_ATT, verify that files such as S_EMPLOYEE_12-1ABC.SAF are renamed to S_CONTACT_12-1ABC.SAF.

Updating Attachments for S_LIT

Perform the task in this section if you are upgrading from Siebel 7.0.x.

At Siebel 7.5, the records in S_LIT are migrated to S_CB_ASSET_VER. You must update the related file names of attachments from S_LIT*SAF to S_CB_ASSET_VER*.SAF.

To update attachments for S_LIT

- 1 Navigate to the following directory:

Windows: *SIEBEL_ROOT*\bin

UNIX: *\$SIEBEL_ROOT*/bin

- 2 Enter the following command:

```
Windows: file_upg_mm.bat ODBC_SOURCE USER_NAME PASSWORD TABLE_OWNER
"FILE_SYSTEM_LOCATION" "SIEBEL_ROOT" "DBSRVR_ROOT"
```

```
UNIX: file_upg_mm.ksh ODBC_SOURCE USER_NAME PASSWORD TABLE_OWNER
FILE_SYSTEM_LOCATION $SIEBEL_ROOT DBSRVR_ROOT
```

where:

- *ODBC_SOURCE* = the ODBC source of the database
- *USER_NAME* = the database user name
- *PASSWORD* = the password for the database user name
- *TABLE_OWNER* = the database tableowner
- *FILE_SYSTEM_LOCATION* = the directory where the file system resides
- *SIEBEL_ROOT* = the directory where the Siebel Server is installed

- *DBSRVR_ROOT* = the directory where Siebel Database Server files are installed

Windows example:

```
file_upg_mm.bat SEBL sadmi n sadmi npw SIEBEL "C:\siebfie" "C:\sba80\siebsrvr"  
"C:\sba80\dbsrvr"
```

UNIX example:

```
file_upg_mm.ksh SEBL sadmi n sadmi npw SIEBEL /usr/siebel/siebfie $SIEBEL_ROOT  
/usr/siebel/dbsrvr
```

Note that the UNIX syntax does not use quotes around *FILE_SYSTEM_LOCATION*, and *DBSRVR_ROOT*.

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

20 Postupgrade Tasks for Siebel Applications

After you upgrade to Siebel 8.0, you must perform a number of Siebel application-specific tasks. This chapter describes the z/OS-specific application upgrade tasks. You must also perform the relevant tasks in the chapter of the *Siebel Database Upgrade Guide* that describes postupgrade tasks for Siebel applications before your Siebel 8.0 system is ready for use. This chapter includes the following topics:

- [Setting Visibility Modes for Access Control on page 253](#)
- [Removing Call Center Duplicate Logins After Upgrade on page 256](#)
- [Upgrading File System Attachments for Siebel Financial Services Call Reports on page 256](#)
- [Upgrading File System for Household Notes and Attachments on page 258](#)
- [Configuring Products and Quotes on page 259](#)
- [Upgrading Siebel Seeded Workflows on page 262](#)
- [Upgrading Inbound Workflows on page 262](#)

Setting Visibility Modes for Access Control

Upgrades: Release 6.2.1 only.

Environments: Development environment only.

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

Certain areas of the Release 8.0 default configuration use Catalog visibility. If you want to continue to use Organization visibility, you must make changes to the business component, view, and picklist properties in order to use Organization visibility. This applies to the following objects:

- Business Components with Catalog Visibility
- Applet picklists with Auto Query Mode or New Query set to None
- Views with Catalog Visibility

After the upgrade, you must carefully evaluate the following business components, views, and picklists, all of which use catalog visibility as the default configuration.

Business Components That Use Catalog by Default

The following business components use Catalog as the default pop-up visibility type:

- Banter Resolution Item
- Competitor

- Component Product
- Contact Decision Issue
- Decision Issue
- ERM Internal News Category
- Internal Product
- My Competitor
- My Internal Product
- Recommended Product
- Related Issue
- Sales Tool
- Solution

To change the visibility properties

- 1 Log into Siebel Tools as a valid user.
- 2 Choose the Types tab.
- 3 Choose the Business Component object in Object Explorer.
- 4 Scroll across the list applet to find the property Popup Visibility Type and change this value from Catalog to Organization or to another valid property.

Picklists That Start in Query Mode by Default

The following picklists, by default, start with Auto Query Mode set to New Query or None:

- Asset Mgmt - Internal Product Pick Applet
- CPG Internal Product Pick Applet
- Catalog Admin Category Assoc Applet—Competitor
- Catalog Admin Category Assoc Applet—Decision Issue
- Catalog Admin Category Assoc Applet—Literature
- Catalog Admin Category Assoc Applet—Product
- Catalog Admin Category Assoc Applet—Resolution Item
- Catalog Admin Category Assoc Applet—Solution
- Catalog Admin Category Assoc Applet—Training
- Catalog Admin Category Assoc Applet—eEvents Event
- FS Parts & Tools Pick Applet
- FS Use Plan Products Pick Applet
- Incentive Compensation Product Pick Applet

- Internal Product Pick Applet (eSales)
- Issue Assoc Applet
- Opportunity Management—Product Number Pick Applet
- Opportunity Management—Product Pick Applet
- Order Entry—Line Item Product Pick Applet
- Partner Finder List Applet
- Product Number Pick Applet
- Product Pick Applet
- Product Pick Applet—No Clear
- Product Pick Applet—No Insert
- SR Internal Product Pick Applet
- Sales Tool Pick Applet
- Service Locator List Applet
- Solution Create List Applet
- Training Course Product Pick Applet

To change the visibility properties

- 1 Log into Siebel Tools as a valid user.
- 2 Choose the Types tab.
- 3 Choose the Applet object in Object Explorer.
- 4 Scroll across the list applet to find the property Auto Query Mode and change this value from New Query or None to no value.

To change the Auto Query Mode property to have no value, remove any search specification that would otherwise be inherited from the Business Component.

Views That Use Catalog by Default

The following views use Catalog as the default visibility applet type:

- Auction Place Bid
- Competitive Company Across Catalogs
- Decision Issue View
- Products Across Catalogs
- SHP Sales Product View
- SWLS eChannel Solution Display View
- Sales Tools Across Catalogs

- Service Solution List View (SCW)
- Solutions Across Catalogs
- eAuction Auction Item Search View

To change the visibility properties

- 1 Log into Siebel Tools as a valid user.
- 2 Choose the Types tab.
- 3 Choose the View object in Object Explorer.
- 4 Scroll across the list applet to find the property Visibility Applet Type and change this value from Catalog to Organization or to another valid property.

Removing Call Center Duplicate Logins After Upgrade

Upgrades: Release 6.2.1 only.

Environments: Production test, production.

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

During the upgrade to Release 8.0, employee and contact data were moved to the same tables. As a result, your database may contain duplicate logins for Contact user login names.

The upgrade appends +ROW_ID to duplicate logins. You must resolve user logins after the upgrade; otherwise, users are unable to log in. For example, contact user login names appear concatenated with their row ID.

To locate user logins that require resolution of duplicates

- 1 Open your Call Center application and navigate to Site Map > Administration - User.
- 2 For each User Administration view (Employees, Persons, and Users) query the User ID field for login name=**+. This query brings up all names that are appended with +ROW_ID.
- 3 Repeat this procedure for each User Administration view (Employees, Persons, and Users).

Upgrading File System Attachments for Siebel Financial Services Call Reports

Upgrades: Release 6.2.1 only.

Environments: Production test, production.

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

When you upgrade to Release 8.0, call report attachments from S_ACTIVITY_ATT are migrated to S_COMM_ATT. This data migration updated the records in the database and the pointer to the physical file, but did not update file system attachments, which use a naming convention that includes the name of the table they are associated with: *TableName_RowID_Revision#.saf*.

Therefore, you need to run a script to copy and rename all files named S_ACTIVITY*.SAF to S_COMM*.SAF so that they correspond to new table names.

To generate the file attachment update script

- 1 Navigate to the following directory:

Windows: *SIEBEL_ROOT\bin*

UNIX: *\$SIEBEL_ROOT/bin*

- 2 Type the following command:

Windows:

```
odbcsql /s "ODBC_DATASOURCE" /u TABLEOWNER  
/p PASSWORD /separator / /a /c rem /e /h  
/o FILESYSTEM_LOCATION/MV_ACT_TO_COMM_ATT_NT.bat  
/l LOGFILE_LOCATION/MV_ACT_TO_COMM_ATT_NT.log SCRIPT_LOCATION/  
MV_ACT_TO_COMM_ATT_NT.sql /v y
```

UNIX:

```
odbcsql /s "ODBC_DATASOURCE" /u TABLEOWNER  
/p PASSWORD /separator / /a /c rem /e /h  
/o FILESYSTEM_LOCATION/mv_act_to_comm_att_unix.ksh  
/l LOGFILE_LOCATION/mv_act_to_comm_att_unix.log SCRIPT_LOCATION/  
mv_act_to_comm_att_unix.sql /v y
```

where:

"ODBC_DATASOURCE" = Data source of the database (entered in quotation marks)

TABLEOWNER = Tableowner

PASSWORD = Tableowner password

FILESYSTEM_LOCATION = Location of the file system

LOGFILE_LOCATION = Location of the log file

SCRIPT_LOCATION = Location of the script

To update the file names of call report attachments

- Make the file system your current directory, then type the following command:

Windows: *MV_ACT_TO_COMM_ATT_NT.BAT*

UNIX: *mv_act_to_comm_att_unix.ksh*

Upgrading File System for Household Notes and Attachments

Upgrades: Release 6.2.1 only.

Environments: Production test, production.

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

If you are upgrading a Siebel Financial Services application, and you are migrating your implementation to the new household design, perform the following procedure to update the file system for household notes and attachments.

The upgrade migrates household note attachments from S_ORGGRP_ATT to S_CONTACT_ATT. This updates the records in the database, but does not update file system attachments, which use a naming convention that includes the name of the table they are associated with:

TableName_RowID_Revision#.saf.

You must run two scripts to complete this migration. First, run a script to migrate attachments from S_ORGGRP_ATT to S_CONTACT_ATT, and notes from S_NOTE_ORGGROUP to S_NOTE_CON in the database. Then, run a script to copy and rename all files named S_ORGGRP*.SAF to S_CONTACT*.SAF so that they correspond to new table names.

To update file attachments

1 Run `Mig_HH_note_att.sql` to migrate attachments from S_ORGGRP_ATT to S_CONTACT_ATT.

2 Navigate to the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

3 Run the following utility:

Windows: `chg_file_sys.bat OLD_TABLE NEW_TABLE "FILE_SYSTEM"`

UNIX: `chg_file_sys.ksh -s OLD_TABLE -t NEW_TABLE -f "FILE_SYSTEM"`

where:

- `OLD_TABLE` = Name of the original table.
- `NEW_TABLE` = Name of the new table to which the original data was migrated.
- `"FILE_SYSTEM"` = Name of the directory where the file system resides (entered inside quotation marks).

Windows example:

```
chg_file_sys.bat S_ORGGRP_ATT S_CONTACT_ATT "C:\siebel\att"
```

UNIX example:

```
chg_file_sys.ksh -s S_ORGGRP_ATT -t S_CONTACT_ATT  
-f "/usr/siebel/siebel/att"
```

- 4 Review the renamed files carefully to verify that they can be accessed by Siebel Financial Services applications.

Opportunity Product Migration

During the upgrade, data is migrated from S_OPTY_PROD to S_REVN. As a result, if you have custom objects that point to S_OPTY_PROD, you must remap the affected base or extension tables. You may have to remap extension columns pointing to S_OPTY_PROD as well.

The following base or extension tables are affected in the migration from S_OPTY_PROD to S_REVN:

- S_FN_OFFR_COLT
- S_FN_OFFR_FEE
- S_FN_OFFR_SCHD
- S_OPTYPRD_ORG
- S_OPTY_PROD1_FNX
- S_OPTY_PROD_FNX
- S_OPTY_PROD_FNXM

For example, if a child object pointed to S_OPTY_PROD, that child object needs to be manually remapped to S_REVN.

A report generated during the upgrade identifies which columns or tables you need to examine. This report, xtnndobstbl.log, lists extension columns that reside on obsolete tables and therefore need to be moved to alternate tables. For more information about reapplying custom extension columns on obsolete tables, see [“Updating the File System Directory” on page 249](#).

Configuring Products and Quotes

Upgrades: Release 6.2.1 only.

Environments: Production test, production.

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

You may need to adjust or configure some customizable products and quotes after upgrade.

Customizable Products

After upgrading to Release 8.0, the administrator must verify that customizable products work as needed. In order to do so, the administrator must validate and release customizable products for use. This action is performed in the Product Administration screen. If the administrator does not release customizable products after the upgrade, customizable products are not active.

Packaged products do not require additional modifications in the postupgrade process.

Quote Items

After upgrade, run the following script to correct an issue where quote items had trailing spaces added accidentally. Running this script is mandatory.

```
update S_QUOTE_ITEM
```

```

setROW_ID = rtrim(ROW_ID)
,   ROOT_QUOTE_ITEM_ID = rtrim(ROOT_QUOTE_ITEM_ID)
,   PAR_SO_ITEM_ID = rtrim(PAR_SO_ITEM_ID)
,   PORT_VALID_PROD_ID = rtrim(PORT_VALID_PROD_ID)
,   PROD_PORT_ID = rtrim(PROD_PORT_ID)
,   INTEGRATION_ID = rtrim(INTEGRATION_ID)
;
commi t
;

```

File System Attachments for Quotes

During the upgrade, quote attachments from S_QUOTE_ATT are migrated to S_ORDER_ATT. This updated the records in the database and the pointer to the physical file, but did not update file system attachments, which use a naming convention that includes the name of the table they are associated with: *TableName_RowID_Revision#.saf*.

You must run a script to copy and rename certain files named S_QUOTE*.SAF to S_ORDER*.SAF so that they correspond to new table names.

To update the file names of quote attachments

- 1 Navigate to the following directory:

Windows: *SIEBEL_ROOT*\bin

UNIX: *\$SIEBEL_ROOT*/bin

- 2 Type the following command:

Windows:

```

odbcsql /s "ODBC_DATASOURCE" /u TABLEOWNER
/p PASSWORD /separator / /a /c rem /e /h
/o FILESYSTEM_LOCATION/MV_QUOTE_TO_ORDER_ATT_NT.bat
/l LOGFILE_LOCATION/MV_QUOTE_TO_ORDER_ATT_NT.log SCRIPT_LOCATION/
MV_QUOTE_TO_ORDER_ATT_NT_NT.sql /v y

```

UNIX:

```
odbcsql /s "ODBC_DATASOURCE" /u TABLEOWNER
/p PASSWORD /separator / /a /c rem /e /h
/o FILESYSTEM_LOCATION/mv_quote_to_order_att_uni x.ksh
/l LOGFILE_LOCATION/mv_quote_to_order_att_uni x.log SCRIPT_LOCATION/
mv_quote_to_order_att_uni x.sql /v y
```

where:

"ODBC_DATASOURCE" = Data source of the database (entered in quotation marks).

TABLEOWNER = Tableowner.

PASSWORD = Tableowner password.

FILESYSTEM_LOCATION = Location of the file system.

LOGFILE_LOCATION = Location of the log file.

SCRIPT_LOCATION = Location of the script.

Flag	Parameter	Description	Required
/s	<i>"ODBC_DATASOURCE"</i>	Data source of the database (entered in quotation marks)	Y
/u	<i>TABLEOWNER</i>	User name to log into database	Y
/p	<i>PASSWORD</i>	Password to log into database	Y
/a	Not applicable	Turn on ODBC auto-commit for session	N
/c	<i>rem</i>	Remark. Specify the string that begins at comment (at the beginning of the line)	Y
/e	Not applicable	Turn off statement printing in spool file	N
/h	Not applicable	Turn off column headers for queries	N
/o	<i>FILESYSTEM_LOCATION</i>	Directory where the file system resides	Y
/l	<i>LOGFILE_LOCATION</i>	Write errors and status to log file specified here	Y
	<i>SCRIPT_LOCATION</i>	Location of the script	Y
/v	Not applicable	Turn on statement printing at execute	N

- 3 Change your directory to the file system, then type the following command:

Windows: MV_QUOTE_TO_ORDER_ATT_NT .BAT

UNIX: mv_quote_to_order_att_uni x.ksh

Upgrading Siebel Seeded Workflows

Upgrades: All upgrades.

Environments: All environments.

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

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

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

To upgrade a seeded workflow

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

Upgrading Inbound Workflows

Upgrades: Releases 6.x, 7.0.x.

Environments: All environments.

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

- Change inbound workflows that contain a “String” type process property to pass the value into type Binary; otherwise, the workflow presents the following error message:

```
Output argument '<Value>' in step 'Read from File' contains data that cannot be passed to string type property 'InputXML'. Data type: 'MEMBLOCK'; String representation of data body: '<?xml version="1.0" encoding="UTF-8"?><?'
```

- After an upgrade from Release 6.2.1 to 8.0, manually change the name of the “EAI MQSeries Transport” business service to the name “EAI MQ Series Server Transport”; otherwise, the workflow presents the following error message:

```
[1] Unable to create the Business Service 'EAI MQSeries Transport'
[2] Could not find 'Class' named 'EAI MQSeries Transport'. This object is inactive or nonexistent.
```

Transitioning from Development to Production

To transition to production, you must import tables from the main Siebel database, as well as importing the Siebel Tools Repository. Refer to the *Siebel Business Process Framework: Workflow Guide* for the steps required to transition to production.

The tables are named as follows:

- Siebel database tables containing workflow information begin S_WFA.
- Siebel Tools Repository tables containing workflow information begin S_WFR.

- Siebel database tables that contain workflow information for releases prior to Release 7.8 begin S_WF_ (Note the underscore after WF.) After the upgrade to Release 8.0, these tables are obsolete and are not referenced by applications.

If you have long-running workflows, verify they are all functioning correctly before transitioning to production.

21 Tuning the Siebel Production Upgrade Scripts

This chapter describes the ways in which you can improve the performance of the production environment upgrade by tuning the production upgrade scripts in a test environment. This chapter contains the following topics:

- [About Tuning the Upgrade Scripts on page 265](#)
- [Optimizing Unload and Load Job Performance on page 266](#)
- [Adding the Statistics Clause to Load Cards on page 267](#)

NOTE: The *Siebel Database Upgrade Guide* describes how to use the Siebel Upgrade Tuner to tune your production upgrade scripts. The Upgrade Tuner is not supported on the IBM DB2 for z/OS database platform.

About Tuning the Upgrade Scripts

Upgrades: All upgrades.

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

In Release 8.0, you can tune the SQL upgrade scripts in a production test environment to improve their performance and then re-use these tested scripts in the live production environment. For example, the scripts used to upgrade your Siebel database are generic. They update your Siebel database to support all Siebel applications' functionality. You can reduce downtime by tuning these scripts to optimize performance by eliminating unneeded SQL statements. You can then re-use these revised scripts in your production upgrade.

You can tune your production upgrade scripts at any time after upgrading the Siebel database schema in your production test environment.

Contacting Oracle's Application Expert Services

CAUTION: You are required to contact your Oracle sales representative for Oracle Advanced Customer Services to request approval from Oracle's Application Expert Services for any upgrade script tuning that you perform. If you do not, you may invalidate your support agreement.

It is recommended (but not required) that you contact Oracle's Application Expert Services for help with the following tasks:

- Running load and unload jobs in parallel
- Changing the job submission order of load and unload jobs.

If you want to change the submission order of jobs other than the load and unload jobs, you must first obtain approval from Oracle's Application Expert Services because many jobs have dependencies on other jobs and must be submitted in a specified sequence.

Optimizing Unload and Load Job Performance

Upgrades: All upgrades.

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

This topic describes ways in which you can improve the performance of the unload and load jobs for the production upgrade. You can do the following:

- Optimize the unload and load jobs to reach maximum parallelism:
 - Run as many of the unload / load jobs in parallel as the DB2 subsystem can support.
 - Change the generated REXX exec job submission order to submit the longest running unload/load jobs first.

If all the unload jobs are run in parallel, the shortest amount of time this process can take is the length of time it takes for the longest unload job to complete.
 - For partitioned tables, split the unload files so that data is unloaded and loaded in parallel for each partition. Add the WHERE clause to the unload SQL to control the data that is unloaded.
 - Overlap load and unload jobs.

Once an unload job for a table has completed, the load job for that table can be started (assuming you have a database schema structure in which there is one table in each table space). This means that load jobs can be running at the same time as unload jobs.
- Add the ORDER BY clauses to the unload SQL to load data in clustering sequence (you must manually add ORDER BY clauses to the unload SQL).

The *.pretedit.jcl (pretkeys) job builds ORDER BY clauses for individual tables into the data set *.syskeys.orderby.
- Use third-party utilities to accelerate the unload/load process.

NOTE: Before using third-party utilities, you are required to contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

You can use the DB2 Cross Loader (an option of the IBM Load utility) to load data directly from the source to the target database, thereby eliminating the unload step.
- Populate new columns as part of the unload SQL.

NOTE: If you want to populate new columns as part of the unload SQL, you are required to contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- Add any large tables to *.TABLIST so that the unload and load processes use the large proc, SIEBEL.PROC(SV7LD10L), which allocates more memory, instead of using the standard proc, SIEBEL.PROC(SV7LD10S).

Adding the Statistics Clause to Load Cards

Upgrades: All upgrades.

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

If your database schema structure follows the *1:1:1* model, and if LOAD REPLACE is specified on a load card (so tables are loaded from scratch), you can improve upgrade performance by collecting statistics while running the load job rather than having to run a separate RUNSTATS job. You can do this by adding the STATISTICS clause to the load cards, for example:

```
STATISTICS TABLE(ALL) INDEX(ALL)
```

```
UPDATE ACCESSPATH
```

NOTE: If LOAD RESUME is specified on a load card, you cannot collect statistics while running the load job.

A

Siebel Upgrade Planning Worksheet

This appendix contains the Upgrade Planning Worksheet. Before you upgrade your Siebel application, photocopy this worksheet, complete it, and give a copy to each member of the upgrade team.

The Upgrade Planning Worksheet contains the following sections:

- [Team Lead Summary on page 269](#)
- [DB2 Connect Information on page 269](#)
- [Siebel Development Environment Information on page 270](#)
- [Siebel Production Environment Information on page 271](#)
- [z/OS Host System Variables Information on page 272](#)

Team Lead Summary

Deployment Team Lead:	
Siebel Administrator:	
Privileged User/Siebel Database User:	
DB2 Systems Programmer (SYSADM):	
DB2 Database Administrator (DBADM):	
Security Administrator:	
z/OS System Programmer:	
Midtier System Administrator:	

DB2 Connect Information

DB2 Host Name/IP Address:

■ DB2 Port Number:

Siebel Development Environment Information

Siebel Gateway Name
Server Name: _____

Enterprise Server Name: _____

Siebel Server Directory: _____

Siebel Database
Configuration Utilities
Directory _____

Database Alias: _____

Siebel Administrator User Name	Siebel Administrator Password	Siebel Administrator User Group
---------------------------------------	--------------------------------------	--

Staging Siebel Schema Qualifier ID (Max. 8 chars)	Staging ODBC Data Source Name (Subsystem name)	Staging Database User Name
--	---	-----------------------------------

Target Siebel Schema Qualifier ID (Max. 8 chars)	Target ODBC Data Source Name (Subsystem name)	Target Database User Name
---	--	----------------------------------

Target Security Group ID	EIM User Group ID (Max. 8 characters)	Siebel User Group ID (Max. 8 characters)
---------------------------------	--	---

Storage Control File Name	Storage Group for Temporary Indexes	Database Name Prefix (Max. 4 characters)
----------------------------------	--	---

4-KB Bufferpool	8-KB Bufferpool	16-KB Bufferpool
------------------------	------------------------	-------------------------

32-KB Bufferpool	Index Bufferpool	
-------------------------	-------------------------	--

NOTE: The Security Group ID is also known as the secondary authorization ID.

Siebel Production Environment Information

Siebel Gateway Name
 Server Name: _____
 Enterprise Server Name: _____
 Siebel Server Directory: _____
 Database Configuration
 Utilities Directory: _____
 Database Alias: _____

Siebel Administrator User Name	Siebel Administrator Password	Siebel Administrator User Group
Staging Siebel Schema Qualifier ID (Max. 8 chars)	Staging ODBC Data Source Name (Subsystem name)	Staging Database User Name
Target Siebel Schema Qualifier ID (Max. 8 chars)	Target ODBC Data Source Name (Subsystem name)	Target Database User Name
Target Security Group ID	EIM User Group ID (Max. 8 characters)	Siebel User Group ID (Max. 8 characters)
Storage Control File Name	Storage Group for Temporary Indexes	Database Name Prefix (Max. 4 characters)
ODBC DSN for Development Database	Database User Name for Development Database	Database Table Owner for Development Database
Import Repository Name	4-KB Bufferpool	16-KB Bufferpool
32-KB Bufferpool	Index Bufferpool	

NOTE: The Security Group ID is also known as the secondary authorization ID.

z/OS Host System Variables Information

**DSN High Level Qualifier
Name (DSNHLO)**

Host/LPAR Name:

DB2 WLM Name:

Code Page / CCSID

**DB2 Load Libraries:
Staging Database**

**DB2 Load Libraries:
Target Database**

To obtain the correct values for the system variables, talk to your DBA or systems programmer.

B

Columns Denormalized During Upgrade to Siebel Release 8.0

This appendix lists columns that are denormalized during upgrades to the current release of Siebel Industry Applications. This appendix contains the following topics:

- [Denormalized Columns for 6.2.1 Siebel Financial Services Applications on page 273](#)
- [Denormalized Columns for 6.2.1 Siebel Business Applications on page 282](#)
- [Denormalized Columns for 6.3 Siebel Industry Solutions on page 289](#)
- [Denormalized Columns for 7.0.4 Siebel Business Applications on page 299](#)
- [Denormalized Columns for 7.0.4 Siebel Financial Services Applications on page 301](#)
- [Denormalized Columns for 7.0.4 Siebel Industry Solutions on page 305](#)
- [Denormalized Columns for 7.5.2 Siebel Industry Applications on page 309](#)

NOTE: If you reduced column lengths when you installed the Siebel Database Configuration Utilities on DB2 UDB for z/OS, you must review them before upgrading to Release 8.0. The upgrade does not recognize the denormalized columns.

Denormalized Columns for 6.2.1 Siebel Financial Services Applications

Table 25 lists columns that are denormalized during upgrades from Release 6.2.1 Siebel Financial Services applications to Release 7.8 Siebel Industry Applications.

Table 25. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ACCNT_POSTN	ACCNT_TYPE_CD	[OU_EXT_ID].[ACCN T_TYPE_CD]	S_ORG_EXT	ACCNT_TYPE_CD
S_ACT_CAL_RSRC	ACT_APPT_REPT_FLG	[ACTIVITY_ID].[APP T_REPT_FLG]	S_EVT_ACT	APPT_REPT_FLG
S_ACT_CAL_RSRC	ACT_APPT_RPTEN D_DT	[ACTIVITY_ID].[APP T_REPT_END_DT]	S_EVT_ACT	APPT_REPT_END_DT
S_ACT_CAL_RSRC	ACT_CAL_TYPE_CD	[ACTIVITY_ID].[CAL _TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ACT_CAL_RSRC	ACT_TEMPLATE_FLG	[ACTIVITY_ID].[TEM PLATE_FLG]	S_EVT_ACT	TEMPLATE_FLG

Table 25. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ACT_CAL_RSRC	ACT_TODO_PLNEND_DT	[ACTIVITY_ID].[TODO_PLAN_END_DT]	S_EVT_ACT	TODO_PLAN_END_DT
S_ACT_CAL_RSRC	ACT_TODO_PLNSTRTDT	[ACTIVITY_ID].[TODO_PLAN_START_DT]	S_EVT_ACT	TODO_PLAN_START_DT
S_ACT_EMP	ACT_ALARM_FLG	[ACTIVITY_ID].[ALARM_FLAG]	S_EVT_ACT	ALARM_FLAG
S_ACT_EMP	ACT_APPT_REPT_FLG	[ACTIVITY_ID].[APPT_REPT_FLG]	S_EVT_ACT	APPT_REPT_FLG
S_ACT_EMP	ACT_APPT_RPTEND_DT	[ACTIVITY_ID].[APPT_REPT_END_DT]	S_EVT_ACT	APPT_REPT_END_DT
S_ACT_EMP	ACT_APPT_START_DT	[ACTIVITY_ID].[APPT_START_DT]	S_EVT_ACT	APPT_START_DT
S_ACT_EMP	ACT_CAL_DISP_FLG	[ACTIVITY_ID].[CAL_DISP_FLG]	S_EVT_ACT	CAL_DISP_FLG
S_ACT_EMP	ACT_CAL_TYPE_CD	[ACTIVITY_ID].[CAL_TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ACT_EMP	ACT_EVT_STAT_CD	[ACTIVITY_ID].[EVT_STAT_CD]	S_EVT_ACT	EVT_STAT_CD
S_ACT_EMP	ACT_TEMPLATE_FLG	[ACTIVITY_ID].[TEMPLATE_FLG]	S_EVT_ACT	TEMPLATE_FLG
S_ACT_EMP	ACT_TODO_PLNEND_DT	[ACTIVITY_ID].[TODO_PLAN_END_DT]	S_EVT_ACT	TODO_PLAN_END_DT
S_ACT_EMP	ACT_TODO_PLNSTRTDT	[ACTIVITY_ID].[TODO_PLAN_START_DT]	S_EVT_ACT	TODO_PLAN_START_DT
S_AGREE_POSTN	AGREE_NAME	[AGREE_ID].[NAME]	S_DOC_AGREE	NAME
S_AGREE_POSTN	AGREE_STAT_CD	[AGREE_ID].[STAT_CD]	S_DOC_AGREE	STAT_CD
S_AGREE_POSTN	AGREE_VALID_FLG	[AGREE_ID].[VALID_FLG]	S_DOC_AGREE	VALID_FLG
S_ASSET_BU	ASSET_NUM	[ASSET_ID].[ASSET_NUM]	S_ASSET	ASSET_NUM
S_ASSET_BU	TYPE_CD	[ASSET_ID].[TYPE_CD]	S_ASSET	TYPE_CD
S_CASE_BU	CASE_NAME	[CASE_ID].[NAME]	S_CASE	NAME
S_CASE_BU	CASE_STATUS_CD	[CASE_ID].[STATUS_CD]	S_CASE	STATUS_CD

Table 25. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_CASE_BU	CASE_TYPE_CD	[CASE_ID].[TYPE_CD]	S_CASE	TYPE_CD
S_CONTACT_BU	AGENT_FLG	[CONTACT_ID].[AGENT_FLG]	S_CONTACT	AGENT_FLG
S_CONTACT_BU	CON_EMP_FLG	[CONTACT_ID].[EMP_FLG]	S_CONTACT	EMP_FLG
S_CONTACT_BU	CON_FST_NAME	[CONTACT_ID].[FST_NAME]	S_CONTACT	FST_NAME
S_CONTACT_BU	CON_LAST_NAME	[CONTACT_ID].[LAST_NAME]	S_CONTACT	LAST_NAME
S_CONTACT_BU	CON_MID_NAME	[CONTACT_ID].[MID_NAME]	S_CONTACT	MID_NAME
S_CONTACT_BU	MEMBER_FLG	[CONTACT_ID].[MEMBER_FLG]	S_CONTACT	MEMBER_FLG
S_CONTACT_BU	PROVIDER_FLG	[CONTACT_ID].[PROVIDER_FLG]	S_CONTACT	PROVIDER_FLG
S_CTLG_CAT_AUC	AUC_AUC_LOT_NUM	[AUC_ITEM_ID].[AUC_LOT_NUM]	S_AUC_ITEM	AUC_LOT_NUM
S_CTLG_CAT_CRSE	CRSE_NAME	[CRSE_ID].[NAME]	S_CRSE	NAME
S_CTLG_CAT_CRSE	CRSE_SUB_TYPE	[CRSE_ID].[SUB_TYPE]	S_CRSE	SUB_TYPE
S_CTLG_CAT_DFCT	DFCT_DFCT_NUM	[PROD_DEFECT_ID].[DEFECT_NUM]	S_PROD_DEFECT	DEFECT_NUM
S_CTLG_CAT_ISS	ISS_NAME	[ISS_ID].[NAME]	S_ISS	NAME
S_CTLG_CAT_LIT	LIT_NAME	[LIT_ID].[NAME]	S_LIT	NAME
S_CTLG_CAT_ORG	ORG_CMPT_FLG	[ORG_ID].[CMPT_FLG]	S_ORG_EXT	CMPT_FLG
S_CTLG_CAT_ORG	ORG_LOC	[ORG_ID].[LOC]	S_ORG_EXT	LOC
S_CTLG_CAT_ORG	ORG_NAME	[ORG_ID].[NAME]	S_ORG_EXT	NAME
S_CTLG_CAT_ORG	ORG_PRTNR_FLG	[ORG_ID].[PRTNR_FLG]	S_ORG_EXT	PRTNR_FLG
S_CTLG_CAT_PROD	PROD_EFF_END_DT	[PROD_ID].[EFF_END_DT]	S_PROD_INT	EFF_END_DT
S_CTLG_CAT_PROD	PROD_EFF_START_DT	[PROD_ID].[EFF_START_DT]	S_PROD_INT	EFF_START_DT

Table 25. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_CTLG_CAT_PROD	PROD_NAME	[PROD_ID].[NAME]	S_PROD_INT	NAME
S_CTLG_CAT_SR	SR_SR_NUM	[SRV_REQ_ID].[SR_NUM]	S_SRV_REQ	SR_NUM
S_CTLG_CAT_SRC	SRC_NAME	[SRC_ID].[NAME]	S_SRC	NAME
S_CTLG_CAT_SRC	SRC_SRC_NUM	[SRC_ID].[SRC_NUM]	S_SRC	SRC_NUM
S_CTLG_CAT_SRC	SRC_SUB_TYPE	[SRC_ID].[SUB_TYPE]	S_SRC	SUB_TYPE
S_CTLGCAT_ASSET	ASSET_ASSET_NUM	[ASSET_ID].[ASSET_NUM]	S_ASSET	ASSET_NUM
S_CTLGCAT_ASSET	ASSET_SERIAL_NUM	[ASSET_ID].[SERIAL_NUM]	S_ASSET	SERIAL_NUM
S_CTLGCAT_PATH	CS_PATH_NAME	[CS_PATH_ID].[NAME]	S_CS_PATH	NAME
S_CTLGCT_RESI_TM	RI_INTR_PUBL_FLG	[RES_ITEM_ID].[INTR_PUBLISH_FLG]	S_RES_ITEM	INTR_PUBLISH_FLG
S_CTLGCT_RESI_TM	RI_NAME	[RES_ITEM_ID].[NAME]	S_RES_ITEM	NAME
S_CTLGCT_RESI_TM	RI_TYPE_CD	[RES_ITEM_ID].[TYPE_CD]	S_RES_ITEM	TYPE_CD
S_DOC_AGREE_BU	AGREE_NAME	[AGREEMENT_ID].[NAME]	S_DOC_AGREE	NAME
S_DOC_AGREE_BU	AGREE_STAT_CD	[AGREEMENT_ID].[STAT_CD]	S_DOC_AGREE	STAT_CD
S_DOC_AGREE_BU	AGREE_VALID_FLG	[AGREEMENT_ID].[VALID_FLG]	S_DOC_AGREE	VALID_FLG
S_DOC_PPSL_BU	PPSL_DFLT_TMPL_FLG	[DOC_PPSL_ID].[DFLT_TEMPLATE_FLG]	S_DOC_PPSL	DFLT_TEMPLATE_FLG
S_DOC_PPSL_BU	PPSL_NAME	[DOC_PPSL_ID].[NAME]	S_DOC_PPSL	NAME
S_DOC_PPSL_BU	PPSL_TEMPLATE_FLG	[DOC_PPSL_ID].[TEMPLATE_FLG]	S_DOC_PPSL	TEMPLATE_FLG
S_DOC_PPSL_BU	PPSL_TYPE_CD	[DOC_PPSL_ID].[PPSL_TYPE_CD]	S_DOC_PPSL	PPSL_TYPE_CD
S_DOC_QUOTE_BU	DOC_QTE_NUM	[DOC_QUOTE_ID].[QUOTE_NUM]	S_DOC_QUOTE	QUOTE_NUM

Table 25. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_DOC_QUOTE_BU	DOC_QTE_REV_NUM	[DOC_QUOTE_ID].[REV_NUM]	S_DOC_QUOTE	REV_NUM
S_DOC_QUOTE_BU	DOC_QUOTE_NAME	[DOC_QUOTE_ID].[NAME]	S_DOC_QUOTE	NAME
S_INSCLM_BU	INSCLAIM_NUM	[INSCLM_ID].[INSCLAIM_NUM]	S_INSCLAIM	INSCLAIM_NUM
S_INSCLM_BU	REV_NUM	[INSCLM_ID].[REVISION_NUM]	S_INSCLAIM	REVISION_NUM
S_INSCLMEL_BU	SEQ_NUM	[INSCLM_ELMNT_ID].[SEQ_NUM]	S_INSCLM_ELMNT	SEQ_NUM
S_INSCLMEL_BU	TYPE_CD	[INSCLM_ELMNT_ID].[TYPE_CD]	S_INSCLM_ELMNT	TYPE_CD
S_INV_TXN_BU	INV_TXN_NUM	[INV_TXN_ID].[INV_TXN_NUM]	S_INV_TXN	INV_TXN_NUM
S_ISS_BU	ISS_LANG_ID	[ISS_ID].[LANG_ID]	S_ISS	LANG_ID
S_ISS_BU	ISS_NAME	[ISS_ID].[NAME]	S_ISS	NAME
S_LOY_MEM_BU	MEM_NUM	[MEMBER_ID].[MEM_NUM]	S_LOY_MEMBER	MEM_NUM
S_LOY_MEM_BU	MEM_TYPE_CD	[MEMBER_ID].[MEM_TYPE_CD]	S_LOY_MEMBER	MEM_TYPE_CD
S_LOY_MEM_PSTN	MEM_NUM	[MEMBER_ID].[MEM_NUM]	S_LOY_MEMBER	MEM_NUM
S_LOY_MEM_PSTN	MEM_TYPE_CD	[MEMBER_ID].[MEM_TYPE_CD]	S_LOY_MEMBER	MEM_TYPE_CD
S_LOY_PROG_BU	PROG_NAME	[PROG_ID].[NAME]	S_LOY_PROGRAM	NAME
S_LOY_PROMO_BU	PROMO_NAME	[PROMO_ID].[NAME]	S_LOY_PROMO	NAME
S_LOY_PROMO_BU	PROMO_NUM	[PROMO_ID].[PROMO_NUM]	S_LOY_PROMO	PROMO_NUM
S_LOY_TXN_BU	TXN_NUM	[TXN_ID].[TXN_NUM]	S_LOY_TXN	TXN_NUM
S_LOY_TXN_BU	TXN_STATUS_CD	[TXN_ID].[STATUS_CD]	S_LOY_TXN	STATUS_CD
S_LOY_TXN_BU	TXN_SUB_TYPE_CD	[TXN_ID].[SUB_TYPE_CD]	S_LOY_TXN	SUB_TYPE_CD

Table 25. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_LOY_TXN_BU	TXN_TYPE_CD	[TXN_ID].[TYPE_CD]	S_LOY_TXN	TYPE_CD
S_MDF_BU	MDF_UI D	[MDF_ID].[MDF_UI D]	S_MDF	MDF_UI D
S_OPTY_BU	PTY_NAME	[OPTY_ID].[NAME]	S_OPTY	NAME
S_OPTY_BU	SUM_CLASS_CD	[OPTY_ID].[SUM_CL ASS_CD]	S_OPTY	SUM_CLASS_CD
S_OPTY_BU	SUM_COMMIT_FLG	[OPTY_ID].[SUM_CO MMIT_FLG]	S_OPTY	SUM_COMMIT_FLG
S_OPTY_BU	SUM_COST_AMT	[OPTY_ID].[SUM_CO ST_AMT]	S_OPTY	SUM_COST_AMT
S_OPTY_BU	SUM_DOWNSIDE_ AMT	[OPTY_ID].[SUM_D OWNSIDE_AMT]	S_OPTY	SUM_DOWNSIDE_ AMT
S_OPTY_BU	SUM_EFFECTIVE_ DT	[OPTY_ID].[SUM_EF FECTIVE_DT]	S_OPTY	SUM_EFFECTIVE_ DT
S_OPTY_BU	SUM_MARGI N_AMT	[OPTY_ID].[SUM_M ARGIN_AMT]	S_OPTY	SUM_MARGI N_AMT
S_OPTY_BU	SUM_REVN_AMT	[OPTY_ID].[SUM_RE VN_AMT]	S_OPTY	SUM_REVN_AMT
S_OPTY_BU	SUM_TYPE_CD	[OPTY_ID].[SUM_TY PE_CD]	S_OPTY	SUM_TYPE_CD
S_OPTY_BU	SUM_UPSI DE_AMT	[OPTY_ID].[SUM_UP SIDE_AMT]	S_OPTY	SUM_UPSI DE_AMT
S_OPTY_BU	SUM_WI N_PROB	[OPTY_ID].[SUM_WI N_PROB]	S_OPTY	SUM_WI N_PROB
S_OPTY_POSTN	NEW_LOAN_FLG	[OPTY_ID].[NEW_LO AN_FLG]	S_OPTY	NEW_LOAN_FLG
S_OPTY_POSTN	PTY_CLOSED_ FLG	[OPTY_ID].[CLOSED _FLG]	S_OPTY	CLOSED_FLG
S_OPTY_POSTN	PTY_NAME	[OPTY_ID].[NAME]	S_OPTY	NAME
S_OPTY_POSTN	SUM_CLASS_CD	[OPTY_ID].[SUM_CL ASS_CD]	S_OPTY	SUM_CLASS_CD
S_OPTY_POSTN	SUM_COMMIT_FLG	[OPTY_ID].[SUM_CO MMIT_FLG]	S_OPTY	SUM_COMMIT_FLG
S_OPTY_POSTN	SUM_COST_AMT	[OPTY_ID].[SUM_CO ST_AMT]	S_OPTY	SUM_COST_AMT

Table 25. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_OPTY_POSTN	SUM_DOWNSIDE_AMT	[OPTY_ID].[SUM_DOWNSIDE_AMT]	S_OPTY	SUM_DOWNSIDE_AMT
S_OPTY_POSTN	SUM_EFFECTIVE_DT	[OPTY_ID].[SUM_EFFECTIVE_DT]	S_OPTY	SUM_EFFECTIVE_DT
S_OPTY_POSTN	SUM_MARGIN_AMT	[OPTY_ID].[SUM_MARGIN_AMT]	S_OPTY	SUM_MARGIN_AMT
S_OPTY_POSTN	SUM_REVN_AMT	[OPTY_ID].[SUM_REVN_AMT]	S_OPTY	SUM_REVN_AMT
S_OPTY_POSTN	SUM_TYPE_CD	[OPTY_ID].[SUM_TYPE_CD]	S_OPTY	SUM_TYPE_CD
S_OPTY_POSTN	SUM_UPSIDE_AMT	[OPTY_ID].[SUM_UPSIDE_AMT]	S_OPTY	SUM_UPSIDE_AMT
S_OPTY_POSTN	SUM_WIN_PROB	[OPTY_ID].[SUM_WIN_PROB]	S_OPTY	SUM_WIN_PROB
S_ORDER_BU	ORDER_CAT_CD	[ORDER_ID].[ORDER_CAT_CD]	S_ORDER	ORDER_CAT_CD
S_ORDER_BU	ORDER_DT	[ORDER_ID].[ORDER_DT]	S_ORDER	ORDER_DT
S_ORDER_BU	ORDER_NUM	[ORDER_ID].[ORDER_NUM]	S_ORDER	ORDER_NUM
S_ORDER_POSTN	ORDER_CAT_CD	[ORDER_ID].[ORDER_CAT_CD]	S_ORDER	ORDER_CAT_CD
S_ORDER_POSTN	ORDER_DT	[ORDER_ID].[ORDER_DT]	S_ORDER	ORDER_DT
S_ORDER_POSTN	ORDER_NUM	[ORDER_ID].[ORDER_NUM]	S_ORDER	ORDER_NUM
S_ORG_BU	ORG_CMPT_FLG	[ORG_ID].[CMPT_FLG]	S_ORG_EXT	CMPT_FLG
S_ORG_BU	ORG_FACILITY_FLG	[ORG_ID].[FACILITY_FLG]	S_ORG_EXT	FACILITY_FLG
S_ORG_BU	ORG_INVSTR_FLG	[ORG_ID].[INVSTR_FLG]	S_ORG_EXT	INVSTR_FLG
S_ORG_BU	ORG_LOC	[ORG_ID].[LOC]	S_ORG_EXT	LOC
S_ORG_BU	ORG_NAME	[ORG_ID].[NAME]	S_ORG_EXT	NAME
S_ORG_BU	ORG_PRTNR_FLG	[ORG_ID].[PRTNR_FLG]	S_ORG_EXT	PRTNR_FLG

Table 25. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ORG_BU	ORG_REF_CUST_FLG	[ORG_ID].[REFERENCE_CUST_FLG]	S_ORG_EXT	REFERENCE_CUST_FLG
S_ORG_BU	ORG_SRV_PROVDR_FLG	[ORG_ID].[SRV_PROVDR_FLG]	S_ORG_EXT	SRV_PROVDR_FLG
S_ORG_DIST_LST	OU_ID	[ORG_PROD_ID].[OU_ID]	S_ORG_PROD	OU_ID
S_ORG_GROUP_BU	OG_GROUP_NAME	[ORG_GROUP_ID].[GROUP_NAME]	S_ORG_GROUP	GROUP_NAME
S_ORG_GROUP_BU	OG_GROUP_TYPE_CD	[ORG_GROUP_ID].[GROUP_TYPE_CD]	S_ORG_GROUP	GROUP_TYPE_CD
S_ORG_GROUP_BU	OG_NAME	[ORG_GROUP_ID].[NAME]	S_ORG_GROUP	NAME
S_ORGGRP_POSTN	GROUP_TYPE_CD	[ORGGRP_ID].[GROUP_TYPE_CD]	S_ORG_GROUP	GROUP_TYPE_CD
S_ORGGRP_POSTN	GRP_GROUP_NAME	[ORGGRP_ID].[GROUP_NAME]	S_ORG_GROUP	GROUP_NAME
S_PARTY_GRP_BU	PG_NAME	[PARTY_GROUP_ID].[NAME]	S_PARTY_GROUP	NAME
S_PERIOD_BU	PERIOD_NAME	[PERIOD_ID].[NAME]	S_PERIOD	NAME
S_POS_BU	POS_NUM	[POS_ID].[POS_NUM]	S_POS	POS_NUM
S_POSTN_CON	AGENT_FLG	[CON_ID].[AGENT_FLG]	S_CONTACT	AGENT_FLG
S_POSTN_CON	MEMBER_FLG	[CON_ID].[MEMBER_FLG]	S_CONTACT	MEMBER_FLG
S_PRIMFCTR_ITM	PRIMDL_ID	[PRIMDL_FCTR_ID].[PRIMDL_ID]	S_PRIMDL_FCTR	PRIMDL_ID
S_PROD_DFCT_BU	DEFECT_NUM	[PROD_DEFECT_ID].[DEFECT_NUM]	S_PROD_DEFECT	DEFECT_NUM
S_PROD_INT_BU	PROD_EFF_END_DT	[PROD_INT_ID].[EFF_END_DT]	S_PROD_INT	EFF_END_DT
S_PROD_INT_BU	PROD_EFF_START_DT	[PROD_INT_ID].[EFF_START_DT]	S_PROD_INT	EFF_START_DT
S_PROD_STYL_TNT	SETUP_STYLE_CD	[PROP_STYLE_ID].[SETUP_STYLE_CD]	S_PROD_STYL_TNT	SETUP_STYLE_CD

Table 25. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_PROJ_BU	PROJ_NAME	[PROJ_ID].[NAME]	S_PROJ	NAME
S_PROJ_BU	PROJ_STATUS_CD	[PROJ_ID].[STATUS_CD]	S_PROJ	STATUS_CD
S_PROJ_BU	PROJ_TYPE_CD	[PROJ_ID].[PROJ_TYPE_CD]	S_PROJ	PROJ_TYPE_CD
S_PSP_PROC_BU	VOD_NAME	[VOD_ID].[VOD_NAME]	S_VOD	VOD_NAME
S_QUOTE_POSTN	QUOTE_NUM	[QUOTE_ID].[QUOTE_NUM]	S_DOC_QUOTE	QUOTE_NUM
S_SALES_MTHD_BU	SALES_METHOD_NAME	[SALES_METHOD_ID].[NAME]	S_SALES_METHOD	NAME
S_SRC_BU	SRC_NAME	[SRC_ID].[NAME]	S_SRC	NAME
S_SRV_REQ_BU	SR_AREA	[SRV_REQ_ID].[SR_AREA]	S_SRV_REQ	SR_AREA
S_SRV_REQ_BU	SR_CST_NUM	[SRV_REQ_ID].[SR_CST_NUM]	S_SRV_REQ	SR_CST_NUM
S_SRV_REQ_BU	SR_NUM	[SRV_REQ_ID].[SR_NUM]	S_SRV_REQ	SR_NUM
S_SRV_REQ_BU	SR_SEV_CD	[SRV_REQ_ID].[SR_SEV_CD]	S_SRV_REQ	SR_SEV_CD
S_SRV_REQ_BU	SR_STAT_ID	[SRV_REQ_ID].[SR_STAT_ID]	S_SRV_REQ	SR_STAT_ID
S_SRV_REQ_BU	SR_SUB_STAT_ID	[SRV_REQ_ID].[SR_SUB_STAT_ID]	S_SRV_REQ	SR_SUB_STAT_ID
S_SRV_REQ_BU	SR_TITLE	[SRV_REQ_ID].[SR_TITLE]	S_SRV_REQ	SR_TITLE
S_SRV_REQ_BU	SR_TYPE_CD	[SRV_REQ_ID].[SR_TYPE_CD]	S_SRV_REQ	SR_TYPE_CD
S_TMPL_PLNIT_BU	PLANITEM_NAME	[TMPL_PLANITEM_ID].[NAME]	S_TMPL_PLANITEM	NAME
S_TMPL_PLNIT_BU	PLANITEM_TYPE_CD	[TMPL_PLANITEM_ID].[TYPE_CD]	S_TMPL_PLANITEM	TYPE_CD
S_TMPL_PLNIT_BU	PLNIT_TMPL_TYPE_CD	[TMPL_PLANITEM_ID].[TMPL_TYPE_CD]	S_TMPL_PLANITEM	TMPL_TYPE_CD
S_USERLIST_BU	UL_NAME	[USERLIST_ID].[NAME]	S_USERLIST	NAME

Table 25. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_WRNTY_CVRG_BU	WRNTY_CVRG_NAME	[WRNTY_CVRG_ID].[WRNTY_NAME]	S_WRNTY_CVRG	WRNTY_NAME
T_MASTER_BU	MASTER_ALIAS	[MASTER_ID].[ALIAS]	T_MASTER	ALIAS
T_MASTER_BU	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME
T_MASTER_DETAIL	DETAIL_SX2_NAME	[DETAIL_SX2_ID].[SX2_NAME]	T_DETAIL_SX2	SX2_NAME
T_MASTER_DETAIL	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME
T_MASTER_PER	MASTER_ALIAS	[MASTER_ID].[ALIAS]	T_MASTER	ALIAS
T_MASTER_POSTN	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME

Denormalized Columns for 6.2.1 Siebel Business Applications

Table 26 lists columns that are denormalized during upgrades from Release 6.2.1 Siebel Business Applications to Release 7.8 Siebel Industry Applications.

Table 26. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Business Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ACCNT_POSTN	ORG_REF_CUST_FLG	[OU_EXT_ID].[REFERENCE_CUST_FLG]	S_ORG_EXT	REFERENCE_CUST_FLG
S_ACT_CAL_RSRC	ACT_APPT_REPT_FLG	[ACTIVITY_ID].[APPT_REPT_FLG]	S_EVT_ACT	APPT_REPT_FLG
S_ACT_CAL_RSRC	ACT_APPT_RPTEND_DT	[ACTIVITY_ID].[APPT_REPT_END_DT]	S_EVT_ACT	APPT_REPT_END_DT
S_ACT_CAL_RSRC	ACT_CAL_TYPE_CD	[ACTIVITY_ID].[CAL_TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ACT_CAL_RSRC	ACT_TEMPLATE_FLG	[ACTIVITY_ID].[TEMPLATE_FLG]	S_EVT_ACT	TEMPLATE_FLG
S_ACT_CAL_RSRC	ACT_TODO_PLNEND_DT	[ACTIVITY_ID].[TODO_PLAN_END_DT]	S_EVT_ACT	TODD_PLAN_END_DT

Table 26. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Business Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ACT_CAL_RSRC	ACT_TODO_PLNSTRDT	[ACTIVITY_ID].[TODO_PLAN_START_DT]	S_EVT_ACT	TODO_PLAN_START_DT
S_ACT_EMP	ACT_ALARM_FLG	[ACTIVITY_ID].[ALARM_FLAG]	S_EVT_ACT	ALARM_FLAG
S_ACT_EMP	ACT_APPT_REPT_FLG	[ACTIVITY_ID].[APPT_REPT_FLG]	S_EVT_ACT	APPT_REPT_FLG
S_ACT_EMP	ACT_APPT_RPTEND_DT	[ACTIVITY_ID].[APPT_REPT_END_DT]	S_EVT_ACT	APPT_REPT_END_DT
S_ACT_EMP	ACT_APPT_START_DT	[ACTIVITY_ID].[APPT_START_DT]	S_EVT_ACT	APPT_START_DT
S_ACT_EMP	ACT_CAL_DISP_FLG	[ACTIVITY_ID].[CAL_DISP_FLG]	S_EVT_ACT	CAL_DISP_FLG
S_ACT_EMP	ACT_EVT_STAT_CD	[ACTIVITY_ID].[EVT_STAT_CD]	S_EVT_ACT	EVT_STAT_CD
S_ACT_EMP	ACT_TEMPLATE_FLG	[ACTIVITY_ID].[TEMPLATE_FLG]	S_EVT_ACT	TEMPLATE_FLG
S_ACT_EMP	ACT_TODO_PLNEND_DT	[ACTIVITY_ID].[TODO_PLAN_END_DT]	S_EVT_ACT	TODO_PLAN_END_DT
S_ACT_EMP	ACT_TODO_PLNSTRDT	[ACTIVITY_ID].[TODO_PLAN_START_DT]	S_EVT_ACT	TODO_PLAN_START_DT
S_ACT_EMP	ACT_CAL_TYPE_CD	[ACTIVITY_ID].[CAL_TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_AGREE_POSTN	AGREE_NAME	[AGREE_ID].[NAME]	S_DOC_AGREE	NAME
S_AGREE_POSTN	AGREE_STAT_CD	[AGREE_ID].[STAT_CD]	S_DOC_AGREE	STAT_CD
S_AGREE_POSTN	AGREE_VALID_FLG	[AGREE_ID].[VALID_FLG]	S_DOC_AGREE	VALID_FLG
S_CONTACT_BU	CON_FST_NAME	[CONTACT_ID].[FST_NAME]	S_CONTACT	FST_NAME
S_CONTACT_BU	CON_LAST_NAME	[CONTACT_ID].[LAST_NAME]	S_CONTACT	LAST_NAME
S_CONTACT_BU	CON_MID_NAME	[CONTACT_ID].[MID_NAME]	S_CONTACT	MID_NAME
S_CONTACT_BU	CON_EMP_FLG	[CONTACT_ID].[EMP_FLG]	S_CONTACT	EMP_FLG
S_CTLG_CAT_AUC	AUC_AUC_LOT_NUM	[AUC_ITEM_ID].[AUC_LOT_NUM]	S_AUC_ITEM	AUC_LOT_NUM

Table 26. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Business Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_CTLG_CAT_CRSE	CRSE_NAME	[CRSE_ID].[NAME]	S_CRSE	NAME
S_CTLG_CAT_CRSE	CRSE_SUB_TYPE	[CRSE_ID].[SUB_TYPE]	S_CRSE	SUB_TYPE
S_CTLG_CAT_DFCT	DFCT_DFCT_NUM	[PROD_DEFECT_ID].[DEFECT_NUM]	S_PROD_DEFECT	DEFECT_NUM
S_CTLG_CAT_ISS	ISS_NAME	[ISS_ID].[NAME]	S_ISS	NAME
S_CTLG_CAT_LIT	LIT_NAME	[LIT_ID].[NAME]	S_LIT	NAME
S_CTLG_CAT_ORG	ORG_CMPT_FLG	[ORG_ID].[CMPT_FLG]	S_ORG_EXT	CMPT_FLG
S_CTLG_CAT_ORG	ORG_LOC	[ORG_ID].[LOC]	S_ORG_EXT	LOC
S_CTLG_CAT_ORG	ORG_NAME	[ORG_ID].[NAME]	S_ORG_EXT	NAME
S_CTLG_CAT_ORG	ORG_PRTNR_FLG	[ORG_ID].[PRTNR_FLG]	S_ORG_EXT	PRTNR_FLG
S_CTLG_CAT_PROD	PROD_EFF_END_DT	[PROD_ID].[EFF_END_DT]	S_PROD_INT	EFF_END_DT
S_CTLG_CAT_PROD	PROD_EFF_START_DT	[PROD_ID].[EFF_START_DT]	S_PROD_INT	EFF_START_DT
S_CTLG_CAT_PROD	PROD_NAME	[PROD_ID].[NAME]	S_PROD_INT	NAME
S_CTLG_CAT_SR	SR_SR_NUM	[SRV_REQ_ID].[SR_NUM]	S_SRV_REQ	SR_NUM
S_CTLG_CAT_SRC	SRC_NAME	[SRC_ID].[NAME]	S_SRC	NAME
S_CTLG_CAT_SRC	SRC_SRC_NUM	[SRC_ID].[SRC_NUM]	S_SRC	SRC_NUM
S_CTLG_CAT_SRC	SRC_SUB_TYPE	[SRC_ID].[SUB_TYPE]	S_SRC	SUB_TYPE
S_CTLGCAT_ASSET	ASSET_ASSET_NUM	[ASSET_ID].[ASSET_NUM]	S_ASSET	ASSET_NUM
S_CTLGCAT_ASSET	ASSET_SERIAL_NUM	[ASSET_ID].[SERIAL_NUM]	S_ASSET	SERIAL_NUM
S_CTLGCAT_PATH	CS_PATH_NAME	[CS_PATH_ID].[NAME]	S_CS_PATH	NAME
S_CTLGCT_RESI TM	RI_INTR_PUBL_FLG	[RES_ITEM_ID].[INTR_PUBLISH_FLG]	S_RESI TEM	INTR_PUBLI SH_FLG
S_CTLGCT_RESI TM	RI_NAME	[RES_ITEM_ID].[NAME]	S_RESI TEM	NAME

Table 26. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Business Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_CTLGCT_RESI_TM	RI_TYPE_CD	[RES_ITEM_ID].[TYPE_CD]	S_RESI_TM	TYPE_CD
S_DOC_AGREE_BU	AGREE_NAME	[AGREEMENT_ID].[NAME]	S_DOC_AGREE	NAME
S_DOC_AGREE_BU	AGREE_STAT_CD	[AGREEMENT_ID].[STAT_CD]	S_DOC_AGREE	STAT_CD
S_DOC_AGREE_BU	AGREE_VALID_FLG	[AGREEMENT_ID].[VALID_FLG]	S_DOC_AGREE	VALID_FLG
S_DOC_PPSL_BU	PPSL_DFLT_TMPL_FLG	[DOC_PPSL_ID].[DFLT_TEMPLATE_FLG]	S_DOC_PPSL	DFLT_TEMPLATE_FLG
S_DOC_PPSL_BU	PPSL_NAME	[DOC_PPSL_ID].[NAME]	S_DOC_PPSL	NAME
S_DOC_PPSL_BU	PPSL_TEMPLATE_FLG	[DOC_PPSL_ID].[TEMPLATE_FLG]	S_DOC_PPSL	TEMPLATE_FLG
S_DOC_PPSL_BU	PPSL_TYPE_CD	[DOC_PPSL_ID].[PPSL_TYPE_CD]	S_DOC_PPSL	PPSL_TYPE_CD
S_DOC_QUOTE_BU	DOC_QTE_NUM	[DOC_QUOTE_ID].[QUOTE_NUM]	S_DOC_QUOTE	QUOTE_NUM
S_DOC_QUOTE_BU	DOC_QTE_REV_NUM	[DOC_QUOTE_ID].[REV_NUM]	S_DOC_QUOTE	REV_NUM
S_DOC_QUOTE_BU	DOC_QUOTE_NAME	[DOC_QUOTE_ID].[NAME]	S_DOC_QUOTE	NAME
S_INV_TXN_BU	INV_TXN_NUM	[INV_TXN_ID].[INV_TXN_NUM]	S_INV_TXN	INV_TXN_NUM
S_ISS_BU	ISS_LANG_ID	[ISS_ID].[LANG_ID]	S_ISS	LANG_ID
S_ISS_BU	ISS_NAME	[ISS_ID].[NAME]	S_ISS	NAME
S_MDF_BU	MDF_UID	[MDF_ID].[MDF_UID]	S_MDF	MDF_UID
S_OPTY_BU	OPTY_NAME	[OPTY_ID].[NAME]	S_OPTY	NAME
S_OPTY_BU	SUM_CLASS_CD	[OPTY_ID].[SUM_CLASS_CD]	S_OPTY	SUM_CLASS_CD
S_OPTY_BU	SUM_COMMIT_FLG	[OPTY_ID].[SUM_COMMIT_FLG]	S_OPTY	SUM_COMMIT_FLG
S_OPTY_BU	SUM_COST_AMT	[OPTY_ID].[SUM_COST_AMT]	S_OPTY	SUM_COST_AMT

Table 26. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Business Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_OPTY_BU	SUM_DOWNSIDE_AMT	[OPTY_ID].[SUM_DOWNSIDE_AMT]	S_OPTY	SUM_DOWNSIDE_AMT
S_OPTY_BU	SUM_EFFECTIVE_DT	[OPTY_ID].[SUM_EFFECTIVE_DT]	S_OPTY	SUM_EFFECTIVE_DT
S_OPTY_BU	SUM_MARGIN_AMT	[OPTY_ID].[SUM_MARGIN_AMT]	S_OPTY	SUM_MARGIN_AMT
S_OPTY_BU	SUM_REVN_AMT	[OPTY_ID].[SUM_REVN_AMT]	S_OPTY	SUM_REVN_AMT
S_OPTY_BU	SUM_TYPE_CD	[OPTY_ID].[SUM_TYPE_CD]	S_OPTY	SUM_TYPE_CD
S_OPTY_BU	SUM_UPSIDE_AMT	[OPTY_ID].[SUM_UPSIDE_AMT]	S_OPTY	SUM_UPSIDE_AMT
S_OPTY_BU	SUM_WIN_PROB	[OPTY_ID].[SUM_WIN_PROB]	S_OPTY	SUM_WIN_PROB
S_OPTY_POSTN	OPTY_NAME	[OPTY_ID].[NAME]	S_OPTY	NAME
S_OPTY_POSTN	SUM_CLASS_CD	[OPTY_ID].[SUM_CLASS_CD]	S_OPTY	SUM_CLASS_CD
S_OPTY_POSTN	SUM_COMMIT_FLG	[OPTY_ID].[SUM_COMMIT_FLG]	S_OPTY	SUM_COMMIT_FLG
S_OPTY_POSTN	SUM_COST_AMT	[OPTY_ID].[SUM_COST_AMT]	S_OPTY	SUM_COST_AMT
S_OPTY_POSTN	SUM_DOWNSIDE_AMT	[OPTY_ID].[SUM_DOWNSIDE_AMT]	S_OPTY	SUM_DOWNSIDE_AMT
S_OPTY_POSTN	SUM_EFFECTIVE_DT	[OPTY_ID].[SUM_EFFECTIVE_DT]	S_OPTY	SUM_EFFECTIVE_DT
S_OPTY_POSTN	SUM_MARGIN_AMT	[OPTY_ID].[SUM_MARGIN_AMT]	S_OPTY	SUM_MARGIN_AMT
S_OPTY_POSTN	SUM_REVN_AMT	[OPTY_ID].[SUM_REVN_AMT]	S_OPTY	SUM_REVN_AMT
S_OPTY_POSTN	SUM_TYPE_CD	[OPTY_ID].[SUM_TYPE_CD]	S_OPTY	SUM_TYPE_CD
S_OPTY_POSTN	SUM_UPSIDE_AMT	[OPTY_ID].[SUM_UPSIDE_AMT]	S_OPTY	SUM_UPSIDE_AMT
S_OPTY_POSTN	SUM_WIN_PROB	[OPTY_ID].[SUM_WIN_PROB]	S_OPTY	SUM_WIN_PROB
S_ORDER_BU	ORDER_CAT_CD	[ORDER_ID].[ORDER_CAT_CD]	S_ORDER	ORDER_CAT_CD

Table 26. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Business Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ORDER_BU	ORDER_DT	[ORDER_ID].[ORDER_DT]	S_ORDER	ORDER_DT
S_ORDER_BU	ORDER_NUM	[ORDER_ID].[ORDER_NUM]	S_ORDER	ORDER_NUM
S_ORDER_POSTN	ORDER_CAT_CD	[ORDER_ID].[ORDER_CAT_CD]	S_ORDER	ORDER_CAT_CD
S_ORDER_POSTN	ORDER_DT	[ORDER_ID].[ORDER_DT]	S_ORDER	ORDER_DT
S_ORDER_POSTN	ORDER_NUM	[ORDER_ID].[ORDER_NUM]	S_ORDER	ORDER_NUM
S_ORG_BU	ORG_CMPT_FLG	[ORG_ID].[CMPT_FLG]	S_ORG_EXT	CMPT_FLG
S_ORG_BU	ORG_LOC	[ORG_ID].[LOC]	S_ORG_EXT	LOC
S_ORG_BU	ORG_NAME	[ORG_ID].[NAME]	S_ORG_EXT	NAME
S_ORG_BU	ORG_PRTNR_FLG	[ORG_ID].[PRTNR_FLG]	S_ORG_EXT	PRTNR_FLG
S_ORG_BU	ORG_REF_CUST_FLG	[ORG_ID].[REFERENCE_CUST_FLG]	S_ORG_EXT	REFERENCE_CUST_FLG
S_ORG_GROUP_BU	OG_GROUP_NAME	[ORG_GROUP_ID].[GROUP_NAME]	S_ORG_GROUP	GROUP_NAME
S_ORG_GROUP_BU	OG_NAME	[ORG_GROUP_ID].[NAME]	S_ORG_GROUP	NAME
S_ORGGRP_POSTN	GRP_NAME	[ORGGRP_ID].[NAME]	S_ORG_GROUP	NAME
S_ORGGRP_POSTN	GRP_GROUP_NAME	[ORGGRP_ID].[GROUP_NAME]	S_ORG_GROUP	GROUP_NAME
S_PARTY_GRP_BU	PG_NAME	[PARTY_GROUP_ID].[NAME]	S_PARTY_GROUP	NAME
S_PERIOD_BU	PERIOD_NAME	[PERIOD_ID].[NAME]	S_PERIOD	NAME
S_PRIMFCTR_ITM	PRIMDL_ID	[PRIMDL_FCTR_ID].[PRIMDL_ID]	S_PRIMDL_FCTR	PRIMDL_ID
S_PROD_DFCT_BU	DEFECT_NUM	[PROD_DEFECT_ID].[DEFECT_NUM]	S_PROD_DEFECT	DEFECT_NUM
S_PROD_INT_BU	PROD_EFF_END_DT	[PROD_INT_ID].[EFF_END_DT]	S_PROD_INT	EFF_END_DT

Table 26. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Business Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_PROD_INT_BU	PROD_EFF_START_DT	[PROD_INT_ID].[EFF_START_DT]	S_PROD_INT	EFF_START_DT
S_PROJ_BU	PROJ_NAME	[PROJ_ID].[NAME]	S_PROJ	NAME
S_PROJ_BU	PROJ_TYPE_CD	[PROJ_ID].[PROJ_TYPE_CD]	S_PROJ	PROJ_TYPE_CD
S_PROJ_BU	PROJ_STATUS_CD	[PROJ_ID].[STATUS_CD]	S_PROJ	STATUS_CD
S_PSP_PROC_BU	VOD_NAME	[VOD_ID].[VOD_NAME]	S_VOD	VOD_NAME
S_QUOTE_POSTN	QUOTE_NUM	[QUOTE_ID].[QUOTE_NUM]	S_DOC_QUOTE	QUOTE_NUM
S_SALES_MTHD_BU	SALES_METHOD_NAME	[SALES_METHOD_ID].[NAME]	S_SALES_METHOD	NAME
S_SRC_BU	SRC_NAME	[SRC_ID].[NAME]	S_SRC	NAME
S_SRV_REQ_BU	SR_TITLE	[SRV_REQ_ID].[SR_TITLE]	S_SRV_REQ	SR_TITLE
S_SRV_REQ_BU	SR_SUB_STAT_ID	[SRV_REQ_ID].[SR_SUB_STAT_ID]	S_SRV_REQ	SR_SUB_STAT_ID
S_SRV_REQ_BU	SR_STAT_ID	[SRV_REQ_ID].[SR_STAT_ID]	S_SRV_REQ	SR_STAT_ID
S_SRV_REQ_BU	SR_SEV_CD	[SRV_REQ_ID].[SR_SEV_CD]	S_SRV_REQ	SR_SEV_CD
S_SRV_REQ_BU	SR_NUM	[SRV_REQ_ID].[SR_NUM]	S_SRV_REQ	SR_NUM
S_SRV_REQ_BU	SR_CST_NUM	[SRV_REQ_ID].[SR_CST_NUM]	S_SRV_REQ	SR_CST_NUM
S_SRV_REQ_BU	SR_AREA	[SRV_REQ_ID].[SR_AREA]	S_SRV_REQ	SR_AREA
S_SRV_REQ_BU	SR_TYPE_CD	[SRV_REQ_ID].[SR_TYPE_CD]	S_SRV_REQ	SR_TYPE_CD
S_TMPL_PLNIT_BU	PLANITEM_NAME	[TMPL_PLANITEM_ID].[NAME]	S_TMPL_PLANITEM	NAME
S_TMPL_PLNIT_BU	PLANITEM_TYPE_CD	[TMPL_PLANITEM_ID].[TYPE_CD]	S_TMPL_PLANITEM	TYPE_CD
S_TMPL_PLNIT_BU	PLNIT_TMPL_TYPE_CD	[TMPL_PLANITEM_ID].[TMPL_TYPE_CD]	S_TMPL_PLANITEM	TMPL_TYPE_CD

Table 26. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Business Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_USERLIST_BU	UL_NAME	[USERLIST_ID].[NAME]	S_USERLIST	NAME
S_WRNTY_CVRG_BU	WRNTY_CVRG_NAME	[WRNTY_CVRG_ID].[WRNTY_NAME]	S_WRNTY_CVRG	WRNTY_NAME
T_MASTER_BU	MASTER_ALIAS	[MASTER_ID].[ALIAS]	T_MASTER	ALIAS
T_MASTER_BU	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME
T_MASTER_DETAIL	DETAIL_SX2_NAME	[DETAIL_SX2_ID].[SX2_NAME]	T_DETAIL_SX2	SX2_NAME
T_MASTER_DETAIL	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME
T_MASTER_PER	MASTER_ALIAS	[MASTER_ID].[ALIAS]	T_MASTER	ALIAS
T_MASTER_POSTN	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME

Denormalized Columns for 6.3 Siebel Industry Solutions

Table 27 lists columns that are denormalized during upgrades from Release 6.3 Siebel Industry Solutions to Release 7.8 Siebel Industry Applications.

Table 27. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ACCNT_POSTN	ACCNT_TYPE_CD	[OU_EXT_ID].[ACCNT_TYPE_CD]	S_ORG_EXT	ACCNT_TYPE_CD
S_ACCNT_POSTN	CLIENT_FLG	[OU_EXT_ID].[CLIENT_FLG]	S_ORG_EXT	CLIENT_FLG
S_ACCNT_POSTN	FACILITY_FLG	[OU_EXT_ID].[FACILITY_FLG]	S_ORG_EXT	FACILITY_FLG
S_ACCNT_POSTN	INVSTR_FLG	[OU_EXT_ID].[INVSTR_FLG]	S_ORG_EXT	INVSTR_FLG
S_ACCNT_POSTN	ORG_REF_CUST_FLG	[OU_EXT_ID].[REFERENCE_CUST_FLG]	S_ORG_EXT	REFERENCE_CUST_FLG

Table 27. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ACCNT_POSTN	SRV_PROVDR_FLG	[OU_EXT_ID].[SRV_PROVDR_FLG]	S_ORG_EXT	SRV_PROVDR_FLG
S_ACT_CAL_RSRC	ACT_APPT_REPT_FLG	[ACTIVITY_ID].[APP T_REPT_FLG]	S_EVT_ACT	APPT_REPT_FLG
S_ACT_CAL_RSRC	ACT_APPT_RPTEND_DT	[ACTIVITY_ID].[APP T_REPT_END_DT]	S_EVT_ACT	APPT_REPT_END_DT
S_ACT_CAL_RSRC	ACT_CAL_TYPE_CD	[ACTIVITY_ID].[CAL _TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ACT_CAL_RSRC	ACT_TEMPLATE_FLG	[ACTIVITY_ID].[TEM PLATE_FLG]	S_EVT_ACT	TEMPLATE_FLG
S_ACT_CAL_RSRC	ACT_TODO_PLNEND_DT	[ACTIVITY_ID].[TOD O_PLAN_END_DT]	S_EVT_ACT	TODO_PLAN_END_DT
S_ACT_CAL_RSRC	ACT_TODO_PLNSTRDT	[ACTIVITY_ID].[TOD O_PLAN_START_DT]	S_EVT_ACT	TODO_PLAN_START_DT
S_ACT_EMP	ACT_ALARM_FLG	[ACTIVITY_ID].[ALA RM_FLAG]	S_EVT_ACT	ALARM_FLAG
S_ACT_EMP	ACT_APPT_REPT_FLG	[ACTIVITY_ID].[APP T_REPT_FLG]	S_EVT_ACT	APPT_REPT_FLG
S_ACT_EMP	ACT_APPT_RPTEND_DT	[ACTIVITY_ID].[APP T_REPT_END_DT]	S_EVT_ACT	APPT_REPT_END_DT
S_ACT_EMP	ACT_APPT_START_DT	[ACTIVITY_ID].[APP T_START_DT]	S_EVT_ACT	APPT_START_DT
S_ACT_EMP	ACT_CAL_DISP_FLG	[ACTIVITY_ID].[CAL _DISP_FLG]	S_EVT_ACT	CAL_DISP_FLG
S_ACT_EMP	ACT_CAL_TYPE_CD	[ACTIVITY_ID].[CAL _TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ACT_EMP	ACT_EVT_STAT_CD	[ACTIVITY_ID].[EVT _STAT_CD]	S_EVT_ACT	EVT_STAT_CD
S_ACT_EMP	ACT_TEMPLATE_FLG	[ACTIVITY_ID].[TEM PLATE_FLG]	S_EVT_ACT	TEMPLATE_FLG
S_ACT_EMP	ACT_TODO_PLNEND_DT	[ACTIVITY_ID].[TOD O_PLAN_END_DT]	S_EVT_ACT	TODO_PLAN_END_DT
S_ACT_EMP	ACT_TODO_PLNSTRDT	[ACTIVITY_ID].[TOD O_PLAN_START_DT]	S_EVT_ACT	TODO_PLAN_START_DT
S_AGREE_POSTN	AGREE_NAME	[AGREE_ID].[NAME]	S_DOC_AGREE	NAME
S_AGREE_POSTN	AGREE_STAT_CD	[AGREE_ID].[STAT_CD]	S_DOC_AGREE	STAT_CD

Table 27. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_AGREE_POSTN	AGREE_VALID_FLG	[AGREE_ID].[VALID_FLG]	S_DOC_AGREE	VALID_FLG
S_ASSET_BU	ASSET_NUM	[ASSET_ID].[ASSET_NUM]	S_ASSET	ASSET_NUM
S_ASSET_BU	TYPE_CD	[ASSET_ID].[TYPE_CD]	S_ASSET	TYPE_CD
S_ASSET_POSTN	ASSET_NUM	[ASSET_ID].[ASSET_NUM]	S_ASSET	ASSET_NUM
S_ASSET_POSTN	TYPE_CD	[ASSET_ID].[TYPE_CD]	S_ASSET	TYPE_CD
S_CASE_BU	CASE_NAME	[CASE_ID].[NAME]	S_CASE	NAME
S_CASE_BU	CASE_STATUS_CD	[CASE_ID].[STATUS_CD]	S_CASE	STATUS_CD
S_CASE_BU	CASE_TYPE_CD	[CASE_ID].[TYPE_CD]	S_CASE	TYPE_CD
S_CONTACT_BU	AGENT_FLG	[CONTACT_ID].[AGENT_FLG]	S_CONTACT	AGENT_FLG
S_CONTACT_BU	CON_EMP_FLG	[CONTACT_ID].[EMP_FLG]	S_CONTACT	EMP_FLG
S_CONTACT_BU	CON_FST_NAME	[CONTACT_ID].[FST_NAME]	S_CONTACT	FST_NAME
S_CONTACT_BU	CON_LAST_NAME	[CONTACT_ID].[LAST_NAME]	S_CONTACT	LAST_NAME
S_CONTACT_BU	CON_MID_NAME	[CONTACT_ID].[MID_NAME]	S_CONTACT	MID_NAME
S_CONTACT_BU	MEMBER_FLG	[CONTACT_ID].[MEMBER_FLG]	S_CONTACT	MEMBER_FLG
S_CONTACT_BU	PROVIDER_FLG	[CONTACT_ID].[PROVIDER_FLG]	S_CONTACT	PROVIDER_FLG
S_CTLG_CAT_AUC	AUC_AUC_LOT_NUM	[AUC_ITEM_ID].[AUC_LOT_NUM]	S_AUC_ITEM	AUC_LOT_NUM
S_CTLG_CAT_CRSE	CRSE_NAME	[CRSE_ID].[NAME]	S_CRSE	NAME
S_CTLG_CAT_CRSE	CRSE_SUB_TYPE	[CRSE_ID].[SUB_TYPE]	S_CRSE	SUB_TYPE
S_CTLG_CAT_DFCT	DFCT_DFCT_NUM	[PROD_DEFECT_ID].[DEFECT_NUM]	S_PROD_DEFECT	DEFECT_NUM

Table 27. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_CTLG_CAT_ISS	ISS_NAME	[ISS_ID].[NAME]	S_ISS	NAME
S_CTLG_CAT_LIT	LIT_NAME	[LIT_ID].[NAME]	S_LIT	NAME
S_CTLG_CAT_ORG	ORG_CMPT_FLG	[ORG_ID].[CMPT_FLG]	S_ORG_EXT	CMPT_FLG
S_CTLG_CAT_ORG	ORG_LOC	[ORG_ID].[LOC]	S_ORG_EXT	LOC
S_CTLG_CAT_ORG	ORG_NAME	[ORG_ID].[NAME]	S_ORG_EXT	NAME
S_CTLG_CAT_ORG	ORG_PRTNR_FLG	[ORG_ID].[PRTNR_FLG]	S_ORG_EXT	PRTNR_FLG
S_CTLG_CAT_PROD	PROD_EFF_END_DT	[PROD_ID].[EFF_END_DT]	S_PROD_INT	EFF_END_DT
S_CTLG_CAT_PROD	PROD_EFF_START_DT	[PROD_ID].[EFF_START_DT]	S_PROD_INT	EFF_START_DT
S_CTLG_CAT_PROD	PROD_NAME	[PROD_ID].[NAME]	S_PROD_INT	NAME
S_CTLG_CAT_SR	SR_SR_NUM	[SRV_REQ_ID].[SR_NUM]	S_SRV_REQ	SR_NUM
S_CTLG_CAT_SRC	SRC_NAME	[SRC_ID].[NAME]	S_SRC	NAME
S_CTLG_CAT_SRC	SRC_SRC_NUM	[SRC_ID].[SRC_NUM]	S_SRC	SRC_NUM
S_CTLG_CAT_SRC	SRC_SUB_TYPE	[SRC_ID].[SUB_TYPE]	S_SRC	SUB_TYPE
S_CTLGCAT_ASSET	ASSET_ASSET_NUM	[ASSET_ID].[ASSET_NUM]	S_ASSET	ASSET_NUM
S_CTLGCAT_ASSET	ASSET_SERIAL_NUM	[ASSET_ID].[SERIAL_NUM]	S_ASSET	SERIAL_NUM
S_CTLGCAT_PATH	CS_PATH_NAME	[CS_PATH_ID].[NAME]	S_CS_PATH	NAME
S_CTLGCT_RESI TM	RI_INTR_PUBL_FLG	[RES_ITEM_ID].[INTR_PUBLISH_FLG]	S_RES_ITEM	INTR_PUBLISH_FLG
S_CTLGCT_RESI TM	RI_NAME	[RES_ITEM_ID].[NAME]	S_RES_ITEM	NAME
S_CTLGCT_RESI TM	RI_TYPE_CD	[RES_ITEM_ID].[TYPE_CD]	S_RES_ITEM	TYPE_CD
S_DOC_AGREE_BU	AGREE_NAME	[AGREEMENT_ID].[NAME]	S_DOC_AGREE	NAME

Table 27. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_DOC_AGREE_BU	AGREE_STAT_CD	[AGREEMENT_ID]. [STAT_CD]	S_DOC_AGREE	STAT_CD
S_DOC_AGREE_BU	AGREE_VALID_FLG	[AGREEMENT_ID]. [VALID_FLG]	S_DOC_AGREE	VALID_FLG
S_DOC_PPSL_BU	PPSL_DFLT_TMPL_FLG	[DOC_PPSL_ID]. [DFLT_TEMPLATE_FLG]	S_DOC_PPSL	DFLT_TEMPLATE_FLG
S_DOC_PPSL_BU	PPSL_NAME	[DOC_PPSL_ID]. [NAME]	S_DOC_PPSL	NAME
S_DOC_PPSL_BU	PPSL_TEMPLATE_FLG	[DOC_PPSL_ID]. [TEMPLATE_FLG]	S_DOC_PPSL	TEMPLATE_FLG
S_DOC_PPSL_BU	PPSL_TYPE_CD	[DOC_PPSL_ID]. [PPSL_TYPE_CD]	S_DOC_PPSL	PPSL_TYPE_CD
S_DOC_QUOTE_BU	DOC_QTE_NUM	[DOC_QUOTE_ID]. [QUOTE_NUM]	S_DOC_QUOTE	QUOTE_NUM
S_DOC_QUOTE_BU	DOC_QTE_REV_NUM	[DOC_QUOTE_ID]. [REV_NUM]	S_DOC_QUOTE	REV_NUM
S_DOC_QUOTE_BU	DOC_QUOTE_NAME	[DOC_QUOTE_ID]. [NAME]	S_DOC_QUOTE	NAME
S_INSCLM_BU	INSCLAIM_NUM	[INSCLM_ID]. [INSCLAIM_NUM]	S_INSCLAIM	INSCLAIM_NUM
S_INSCLM_BU	REV_NUM	[INSCLM_ID]. [REVISION_NUM]	S_INSCLAIM	REVISION_NUM
S_INSCLMEL_BU	SEQ_NUM	[INSCLM_ELMNT_ID]. [SEQ_NUM]	S_INSCLM_ELMNT	SEQ_NUM
S_INSCLMEL_BU	TYPE_CD	[INSCLM_ELMNT_ID]. [TYPE_CD]	S_INSCLM_ELMNT	TYPE_CD
S_INV_TXN_BU	INV_TXN_NUM	[INV_TXN_ID]. [INV_TXN_NUM]	S_INV_TXN	INV_TXN_NUM
S_ISS_BU	ISS_LANG_ID	[ISS_ID]. [LANG_ID]	S_ISS	LANG_ID
S_ISS_BU	ISS_NAME	[ISS_ID]. [NAME]	S_ISS	NAME
S_LOY_MEM_BU	MEM_NUM	[MEMBER_ID]. [MEM_NUM]	S_LOY_MEMBER	MEM_NUM
S_LOY_MEM_BU	MEM_TYPE_CD	[MEMBER_ID]. [MEM_TYPE_CD]	S_LOY_MEMBER	MEM_TYPE_CD
S_LOY_MEM_PSTN	MEM_NUM	[MEMBER_ID]. [MEM_NUM]	S_LOY_MEMBER	MEM_NUM

Table 27. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_LOY_MEM_PSTN	MEM_TYPE_CD	[MEMBER_ID]. [MEM_TYPE_CD]	S_LOY_MEMBER	MEM_TYPE_CD
S_LOY_PROG_BU	PROG_NAME	[PROG_ID].[NAME]	S_LOY_PROG	NAME
S_LOY_PROMO_BU	PROMO_NAME	[PROMO_ID]. [NAME]	S_LOY_PROMO	NAME
S_LOY_PROMO_BU	PROMO_NUM	[PROMO_ID]. [PROMO_NUM]	S_LOY_PROMO	PROMO_NUM
S_LOY_TXN_BU	TXN_NUM	[TXN_ID].[TXN_ NUM]	S_LOY_TXN	TXN_NUM
S_LOY_TXN_BU	TXN_STATUS_CD	[TXN_ID].[STATUS_ CD]	S_LOY_TXN	STATUS_CD
S_LOY_TXN_BU	TXN_SUB_TYPE_CD	[TXN_ID].[SUB_ TYPE_CD]	S_LOY_TXN	SUB_TYPE_CD
S_LOY_TXN_BU	TXN_TYPE_CD	[TXN_ID].[TYPE_ CD]	S_LOY_TXN	TYPE_CD
S_MDF_BU	MDF_UI D	[MDF_ID].[MDF_ UID]	S_MDF	MDF_UI D
S_OPTY_BU	OPTY_NAME	[OPTY_ID].[NAME]	S_OPTY	NAME
S_OPTY_BU	SUM_CLASS_CD	[OPTY_ID].[SUM_ CLASS_CD]	S_OPTY	SUM_CLASS_CD
S_OPTY_BU	SUM_COMMIT_FLG	[OPTY_ID].[SUM_ COMMIT_FLG]	S_OPTY	SUM_COMMIT_FLG
S_OPTY_BU	SUM_COST_AMT	[OPTY_ID].[SUM_ COST_AMT]	S_OPTY	SUM_COST_AMT
S_OPTY_BU	SUM_DOWNSIDE_ AMT	[OPTY_ID].[SUM_ DOWNSIDE_AMT]	S_OPTY	SUM_DOWNSIDE_ AMT
S_OPTY_BU	SUM_EFFECTIVE_ DT	[OPTY_ID].[SUM_ EFFECTIVE_DT]	S_OPTY	SUM_EFFECTIVE_ DT
S_OPTY_BU	SUM_MARGIN_AMT	[OPTY_ID].[SUM_ MARGIN_AMT]	S_OPTY	SUM_MARGIN_AMT
S_OPTY_BU	SUM_REVN_AMT	[OPTY_ID].[SUM_ REVN_AMT]	S_OPTY	SUM_REVN_AMT
S_OPTY_BU	SUM_TYPE_CD	[OPTY_ID].[SUM_ TYPE_CD]	S_OPTY	SUM_TYPE_CD
S_OPTY_BU	SUM_UPSIDE_AMT	[OPTY_ID].[SUM_ UPSIDE_AMT]	S_OPTY	SUM_UPSIDE_AMT

Table 27. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_OPTY_BU	SUM_WI N_PROB	[OPTY_ID].[SUM_WI N_PROB]	S_OPTY	SUM_WI N_PROB
S_OPTY_POSTN	CONSUMER_OPTY_FLG	[OPTY_ID].[CONSUMER_OPTY_FLG]	S_OPTY	CONSUMER_OPTY_FLG
S_OPTY_POSTN	NEW_LOAN_FLG	[OPTY_ID].[NEW_LOAN_FLG]	S_OPTY	NEW_LOAN_FLG
S_OPTY_POSTN	OPTY_CLOSED_FLG	[OPTY_ID].[CLOSED_FLG]	S_OPTY	CLOSED_FLG
S_OPTY_POSTN	OPTY_NAME	[OPTY_ID].[NAME]	S_OPTY	NAME
S_OPTY_POSTN	SECURE_FLG	[OPTY_ID].[SECURE_FLG]	S_OPTY	SECURE_FLG
S_OPTY_POSTN	SUM_CLASS_CD	[OPTY_ID].[SUM_CLASS_CD]	S_OPTY	SUM_CLASS_CD
S_OPTY_POSTN	SUM_COMMIT_FLG	[OPTY_ID].[SUM_COMMIT_FLG]	S_OPTY	SUM_COMMIT_FLG
S_OPTY_POSTN	SUM_COST_AMT	[OPTY_ID].[SUM_COST_AMT]	S_OPTY	SUM_COST_AMT
S_OPTY_POSTN	SUM_DOWNSIDE_AMT	[OPTY_ID].[SUM_DOWNSIDE_AMT]	S_OPTY	SUM_DOWNSIDE_AMT
S_OPTY_POSTN	SUM_EFFECTIVE_DT	[OPTY_ID].[SUM_EFFECTIVE_DT]	S_OPTY	SUM_EFFECTIVE_DT
S_OPTY_POSTN	SUM_MARGIN_AMT	[OPTY_ID].[SUM_MARGIN_AMT]	S_OPTY	SUM_MARGIN_AMT
S_OPTY_POSTN	SUM_REVN_AMT	[OPTY_ID].[SUM_REVN_AMT]	S_OPTY	SUM_REVN_AMT
S_OPTY_POSTN	SUM_TYPE_CD	[OPTY_ID].[SUM_TYPE_CD]	S_OPTY	SUM_TYPE_CD
S_OPTY_POSTN	SUM_UPSIDE_AMT	[OPTY_ID].[SUM_UPSIDE_AMT]	S_OPTY	SUM_UPSIDE_AMT
S_OPTY_POSTN	SUM_WI N_PROB	[OPTY_ID].[SUM_WI N_PROB]	S_OPTY	SUM_WI N_PROB
S_ORDER_BU	ORDER_CAT_CD	[ORDER_ID].[ORDER_CAT_CD]	S_ORDER	ORDER_CAT_CD
S_ORDER_BU	ORDER_DT	[ORDER_ID].[ORDER_DT]	S_ORDER	ORDER_DT
S_ORDER_BU	ORDER_NUM	[ORDER_ID].[ORDER_NUM]	S_ORDER	ORDER_NUM

Table 27. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ORDER_POSTN	ORDER_CAT_CD	[ORDER_ID]. [ORDER_CAT_CD]	S_ORDER	ORDER_CAT_CD
S_ORDER_POSTN	ORDER_DT	[ORDER_ID]. [ORDER_DT]	S_ORDER	ORDER_DT
S_ORDER_POSTN	ORDER_NUM	[ORDER_ID]. [ORDER_NUM]	S_ORDER	ORDER_NUM
S_ORG_BU	ORG_CMPT_FLG	[ORG_ID].[CMPT_ FLG]	S_ORG_EXT	CMPT_FLG
S_ORG_BU	ORG_FACI LI TY_ FLG	[ORG_ID].[FACILITY _FLG]	S_ORG_EXT	FACI LI TY_FLG
S_ORG_BU	ORG_I NVSTR_FLG	[ORG_ID].[INVSTR_ FLG]	S_ORG_EXT	I NVSTR_FLG
S_ORG_BU	ORG_LOC	[ORG_ID].[LOC]	S_ORG_EXT	LOC
S_ORG_BU	ORG_NAME	[ORG_ID].[NAME]	S_ORG_EXT	NAME
S_ORG_BU	ORG_PRTNR_FLG	[ORG_ID].[PRTNR_ FLG]	S_ORG_EXT	PRTNR_FLG
S_ORG_BU	ORG_REF_CUST_ FLG	[ORG_ID].[REFEREN CE_CUST_FLG]	S_ORG_EXT	REFERENCE_CUST_ FLG
S_ORG_BU	ORG_SRV_PROVDR_ FLG	[ORG_ID].[SRV_ PROVDR_FLG]	S_ORG_EXT	SRV_PROVDR_FLG
S_ORG_DI ST_LST	OU_ID	[ORG_PROD_ID]. [OU_ID]	S_ORG_PROD	OU_ID
S_ORG_GROUP_BU	OG_GROUP_NAME	[ORG_GROUP_ID]. [GROUP_NAME]	S_ORG_GROUP	GROUP_NAME
S_ORG_GROUP_BU	OG_GROUP_TYPE_ CD	[ORG_GROUP_ID]. [GROUP_TYPE_CD]	S_ORG_GROUP	GROUP_TYPE_CD
S_ORG_GROUP_BU	OG_NAME	[ORG_GROUP_ID]. [NAME]	S_ORG_GROUP	NAME
S_ORGGRP_POSTN	GROUP_TYPE_CD	[ORGGRP_ID].[GRO UP_TYPE_CD]	S_ORG_GROUP	GROUP_TYPE_CD
S_ORGGRP_POSTN	GRP_GROUP_NAME	[ORGGRP_ID]. [GROUP_NAME]	S_ORG_GROUP	GROUP_NAME
S_ORGGRP_POSTN	GRP_NAME	[ORGGRP_ID]. [NAME]	S_ORG_GROUP	NAME
S_PARTY_GRP_BU	PG_NAME	[PARTY_GROUP_ID]. [NAME]	S_PARTY_GROUP	NAME

Table 27. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_PERIOD_BU	PERIOD_NAME	[PERIOD_ID]. [NAME]	S_PERIOD	NAME
S_POS_BU	POS_NUM	[POS_ID].[POS_] [NUM]	S_POS	POS_NUM
S_POSTN_CON	AGENT_FLG	[CON_ID].[AGENT_] [FLG]	S_CONTACT	AGENT_FLG
S_POSTN_CON	MEMBER_FLG	[CON_ID].[MEMBER_] [FLG]	S_CONTACT	MEMBER_FLG
S_POSTN_CON	PROVIDER_FLG	[CON_ID]. [PROVIDER_FLG]	S_CONTACT	PROVIDER_FLG
S_PRIFCTR_ITM	PRIMDL_ID	[PRIMDL_FCTR_ID]. [PRIMDL_ID]	S_PRIMDL_FCTR	PRIMDL_ID
S_PROD_DFCT_BU	DEFECT_NUM	[PROD_DEFECT_ID]. [DEFECT_NUM]	S_PROD_DEFECT	DEFECT_NUM
S_PROD_INT_BU	PROD_CD	[PROD_INT_ID]. [PROD_CD]	S_PROD_INT	PROD_CD
S_PROD_INT_BU	PROD_EFF_END_DT	[PROD_INT_ID]. [EFF_END_DT]	S_PROD_INT	EFF_END_DT
S_PROD_INT_BU	PROD_EFF_START_DT	[PROD_INT_ID]. [EFF_START_DT]	S_PROD_INT	EFF_START_DT
S_PROD_STYL_TNT	SETUP_STYLE_CD	[PROP_STYLE_ID]. [SETUP_STYLE_CD]	S_PROP_STYL_TNT	SETUP_STYLE_CD
S_PROJ_BU	PROJ_NAME	[PROJ_ID].[NAME]	S_PROJ	NAME
S_PROJ_BU	PROJ_STATUS_CD	[PROJ_ID].[STATUS_] [CD]	S_PROJ	STATUS_CD
S_PROJ_BU	PROJ_TYPE_CD	[PROJ_ID].[PROJ_] [TYPE_CD]	S_PROJ	PROJ_TYPE_CD
S_PSP_PROC_BU	VOD_NAME	[VOD_ID].[VOD_NA] [ME]	S_VOD	VOD_NAME
S_QUOTE_POSTN	QUOTE_NUM	[QUOTE_ID]. [QUOTE_NUM]	S_DOC_QUOTE	QUOTE_NUM
S_SALES_MTHD_BU	SALES_METHOD_NAME	[SALES_METHOD_] [ID].[NAME]	S_SALES_METHOD	NAME
S_SRC_BU	SRC_NAME	[SRC_ID].[NAME]	S_SRC	NAME
S_SRV_REQ_BU	SR_AREA	[SRV_REQ_ID].[SR_] [AREA]	S_SRV_REQ	SR_AREA

Table 27. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_SRV_REQ_BU	SR_CST_NUM	[SRV_REQ_ID].[SR_CST_NUM]	S_SRV_REQ	SR_CST_NUM
S_SRV_REQ_BU	SR_NUM	[SRV_REQ_ID].[SR_NUM]	S_SRV_REQ	SR_NUM
S_SRV_REQ_BU	SR_SEV_CD	[SRV_REQ_ID].[SR_SEV_CD]	S_SRV_REQ	SR_SEV_CD
S_SRV_REQ_BU	SR_STAT_ID	[SRV_REQ_ID].[SR_STAT_ID]	S_SRV_REQ	SR_STAT_ID
S_SRV_REQ_BU	SR_SUB_STAT_ID	[SRV_REQ_ID].[SR_SUB_STAT_ID]	S_SRV_REQ	SR_SUB_STAT_ID
S_SRV_REQ_BU	SR_TITLE	[SRV_REQ_ID].[SR_TITLE]	S_SRV_REQ	SR_TITLE
S_SRV_REQ_BU	SR_TYPE_CD	[SRV_REQ_ID].[SR_TYPE_CD]	S_SRV_REQ	SR_TYPE_CD
S_TMPL_PLNIT_BU	PLANITEM_NAME	[TMPL_PLANITEM_ID].[NAME]	S_TMPL_PLANITEM	NAME
S_TMPL_PLNIT_BU	PLANITEM_TYPE_CD	[TMPL_PLANITEM_ID].[TYPE_CD]	S_TMPL_PLANITEM	TYPE_CD
S_TMPL_PLNIT_BU	PLNIT_TMPL_TYPE_CD	[TMPL_PLANITEM_ID].[TMPL_TYPE_CD]	S_TMPL_PLANITEM	TMPL_TYPE_CD
S_USERLIST_BU	UL_NAME	[USERLIST_ID].[NAME]	S_USERLIST	NAME
S_WRNTY_CVRG_BU	WRNTY_CVRG_NAME	[WRNTY_CVRG_ID].[WRNTY_NAME]	S_WRNTY_CVRG	WRNTY_NAME
T_MASTER_BU	MASTER_ALIAS	[MASTER_ID].[ALIAS]	T_MASTER	ALIAS
T_MASTER_BU	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME
T_MASTER_DETAIL	DETAIL_SX2_NAME	[DETAIL_SX2_ID].[SX2_NAME]	T_DETAIL_SX2	SX2_NAME
T_MASTER_DETAIL	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME
T_MASTER_PER	MASTER_ALIAS	[MASTER_ID].[ALIAS]	T_MASTER	ALIAS
T_MASTER_POSTN	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME

Denormalized Columns for 7.0.4 Siebel Business Applications

Table 28 lists columns that are denormalized during upgrades from Release 7.0.4 Siebel Business applications to Release 7.8 Siebel Industry Applications.

Table 28. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Business Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ACCNT_POSTN	ORG_REF_CUST_FLG	[OU_EXT_ID].[REFERENCE_CUST_FLG]	S_ORG_EXT	REFERENCE_CUST_FLG
S_ACT_CAL_RSRC	ACT_APPT_REPT_FLG	[ACTIVITY_ID].[APPT_REPT_FLG]	S_EVT_ACT	APPT_REPT_FLG
S_ACT_CAL_RSRC	ACT_APPT_RPTEND_DT	[ACTIVITY_ID].[APPT_REPT_END_DT]	S_EVT_ACT	APPT_REPT_END_DT
S_ACT_CAL_RSRC	ACT_CAL_TYPE_CD	[ACTIVITY_ID].[CAL_TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ACT_CAL_RSRC	ACT_TEMPLATE_FLG	[ACTIVITY_ID].[TEMPLATE_FLG]	S_EVT_ACT	TEMPLATE_FLG
S_ACT_CAL_RSRC	ACT_TODO_PLNEND_DT	[ACTIVITY_ID].[TODO_PLAN_END_DT]	S_EVT_ACT	TODD_PLAN_END_DT
S_ACT_CAL_RSRC	ACT_TODO_PLNSTRDT	[ACTIVITY_ID].[TODO_PLAN_START_DT]	S_EVT_ACT	TODD_PLAN_START_DT
S_ACT_EMP	ACT_CAL_TYPE_CD	[ACTIVITY_ID].[CAL_TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_CONTACT_BU	CON_FST_NAME	[CONTACT_ID].[FST_NAME]	S_CONTACT	FST_NAME
S_CONTACT_BU	CON_LAST_NAME	[CONTACT_ID].[LAST_NAME]	S_CONTACT	LAST_NAME
S_CONTACT_BU	CON_MID_NAME	[CONTACT_ID].[MID_NAME]	S_CONTACT	MID_NAME
S_CONTACT_BU	CON_EMP_FLG	[CONTACT_ID].[EMP_FLG]	S_CONTACT	EMP_FLG
S_ORG_BU	ORG_REF_CUST_FLG	[ORG_ID].[REFERENCE_CUST_FLG]	S_ORG_EXT	REFERENCE_CUST_FLG
S_ORG_GROUP_BU	OG_GROUP_NAME	[ORG_GROUP_ID].[GROUP_NAME]	S_ORG_GROUP	GROUP_NAME
S_ORG_GROUP_BU	OG_NAME	[ORG_GROUP_ID].[NAME]	S_ORG_GROUP	NAME

Table 28. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Business Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ORGGRP_POSTN	GRP_GROUP_NAME	[ORGGRP_ID].[GROUP_NAME]	S_ORG_GROUP	GROUP_NAME
S_PARTY_GRP_BU	PG_NAME	[PARTY_GROUP_ID].[NAME]	S_PARTY_GROUP	NAME
S_PSP_PROC_BU	VOD_NAME	[VOD_ID].[VOD_NAME]	S_VOD	VOD_NAME
S_QUOTE_POSTN	QUOTE_NUM	[QUOTE_ID].[QUOTE_NUM]	S_DOC_QUOTE	QUOTE_NUM
S_SRC_BU	SRC_NAME	[SRC_ID].[NAME]	S_SRC	NAME
S_SRV_REQ_BU	SR_TITLE	[SRV_REQ_ID].[SR_TITLE]	S_SRV_REQ	SR_TITLE
S_SRV_REQ_BU	SR_SUB_STAT_ID	[SRV_REQ_ID].[SR_SUB_STAT_ID]	S_SRV_REQ	SR_SUB_STAT_ID
S_SRV_REQ_BU	SR_STAT_ID	[SRV_REQ_ID].[SR_STAT_ID]	S_SRV_REQ	SR_STAT_ID
S_SRV_REQ_BU	SR_SEV_CD	[SRV_REQ_ID].[SR_SEV_CD]	S_SRV_REQ	SR_SEV_CD
S_SRV_REQ_BU	SR_NUM	[SRV_REQ_ID].[SR_NUM]	S_SRV_REQ	SR_NUM
S_SRV_REQ_BU	SR_CST_NUM	[SRV_REQ_ID].[SR_CST_NUM]	S_SRV_REQ	SR_CST_NUM
S_SRV_REQ_BU	SR_AREA	[SRV_REQ_ID].[SR_AREA]	S_SRV_REQ	SR_AREA
S_SRV_REQ_BU	SR_TYPE_CD	[SRV_REQ_ID].[SR_TYPE_CD]	S_SRV_REQ	SR_TYPE_CD
S_USERLIST_BU	UL_NAME	[USERLIST_ID].[NAME]	S_USERLIST	NAME
T_MASTER_BU	MASTER_ALIAS	[MASTER_ID].[ALIAS]	T_MASTER	ALIAS
T_MASTER_BU	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME
T_MASTER_DETAIL	DETAIL_SX2_NAME	[DETAIL_SX2_ID].[SX2_NAME]	T_DETAIL_SX2	SX2_NAME
T_MASTER_DETAIL	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME
T_MASTER_PER	MASTER_ALIAS	[MASTER_ID].[ALIAS]	T_MASTER	ALIAS
T_MASTER_POSTN	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME

Denormalized Columns for 7.0.4 Siebel Financial Services Applications

Table 29 lists columns that are denormalized during upgrades from Release 7.0.4 Siebel Financial Services applications to Release 7.8 Siebel Industry Applications.

Table 29. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ACCNT_POSTN	ACCNT_TYPE_CD	[OU_EXT_ID]. [ACCNT_TYPE_CD]	S_ORG_EXT	ACCNT_TYPE_CD
S_ACT_CAL_RSRC	ACT_APPT_REPT_FLG	[ACTIVITY_ID]. [APPT_REPT_FLG]	S_EVT_ACT	APPT_REPT_FLG
S_ACT_CAL_RSRC	ACT_APPT_RPTEND_DT	[ACTIVITY_ID].[APP T_REPT_END_DT]	S_EVT_ACT	APPT_REPT_END_DT
S_ACT_CAL_RSRC	ACT_CAL_TYPE_CD	[ACTIVITY_ID]. [CAL_TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ACT_CAL_RSRC	ACT_TEMPLATE_FLG	[ACTIVITY_ID]. [TEMPLATE_FLG]	S_EVT_ACT	TEMPLATE_FLG
S_ACT_CAL_RSRC	ACT_TODO_PLNEND_DT	[ACTIVITY_ID].[TOD O_PLAN_END_DT]	S_EVT_ACT	TODDO_PLAN_END_DT
S_ACT_CAL_RSRC	ACT_TODO_PLNSTRTDT	[ACTIVITY_ID].[TOD O_PLAN_START_DT]	S_EVT_ACT	TODDO_PLAN_START_DT
S_ACT_EMP	ACT_CAL_TYPE_CD	[ACTIVITY_ID]. [CAL_TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ASSET_BU	ASSET_NUM	[ASSET_ID].[ASSET _NUM]	S_ASSET	ASSET_NUM
S_ASSET_BU	TYPE_CD	[ASSET_ID].[TYPE_ CD]	S_ASSET	TYPE_CD
S_CASE_BU	CASE_NAME	[CASE_ID].[NAME]	S_CASE	NAME
S_CASE_BU	CASE_STATUS_CD	[CASE_ID].[STATUS _CD]	S_CASE	STATUS_CD
S_CASE_BU	CASE_TYPE_CD	[CASE_ID].[TYPE_ CD]	S_CASE	TYPE_CD
S_CONTACT_BU	AGENT_FLG	[CONTACT_ID]. [AGENT_FLG]	S_CONTACT	AGENT_FLG
S_CONTACT_BU	CON_EMP_FLG	[CONTACT_ID]. [EMP_FLG]	S_CONTACT	EMP_FLG

Table 29. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_CONTACT_BU	CON_FST_NAME	[CONTACT_ID].[FST_NAME]	S_CONTACT	FST_NAME
S_CONTACT_BU	CON_LAST_NAME	[CONTACT_ID].[LAST_NAME]	S_CONTACT	LAST_NAME
S_CONTACT_BU	CON_MID_NAME	[CONTACT_ID].[MID_NAME]	S_CONTACT	MID_NAME
S_CONTACT_BU	MEMBER_FLG	[CONTACT_ID].[MEMBER_FLG]	S_CONTACT	MEMBER_FLG
S_CONTACT_BU	PROVIDER_FLG	[CONTACT_ID].[PROVIDER_FLG]	S_CONTACT	PROVIDER_FLG
S_INSCLM_BU	INSCLAIM_NUM	[INSCLM_ID].[INSCLAIM_NUM]	S_INSCCLAIM	INSCLAIM_NUM
S_INSCLM_BU	REV_NUM	[INSCLM_ID].[REVISION_NUM]	S_INSCCLAIM	REVISION_NUM
S_INSCLMEL_BU	SEQ_NUM	[INSCLM_ELMNT_ID].[SEQ_NUM]	S_INSCLM_ELMNT	SEQ_NUM
S_INSCLMEL_BU	TYPE_CD	[INSCLM_ELMNT_ID].[TYPE_CD]	S_INSCLM_ELMNT	TYPE_CD
S_LOY_MEM_BU	MEM_NUM	[MEMBER_ID].[MEM_NUM]	S_LOY_MEMBER	MEM_NUM
S_LOY_MEM_BU	MEM_TYPE_CD	[MEMBER_ID].[MEM_TYPE_CD]	S_LOY_MEMBER	MEM_TYPE_CD
S_LOY_MEM_PSTN	MEM_NUM	[MEMBER_ID].[MEM_NUM]	S_LOY_MEMBER	MEM_NUM
S_LOY_MEM_PSTN	MEM_TYPE_CD	[MEMBER_ID].[MEM_TYPE_CD]	S_LOY_MEMBER	MEM_TYPE_CD
S_LOY_PROG_BU	PROG_NAME	[PROG_ID].[NAME]	S_LOY_PROGRAM	NAME
S_LOY_PROMO_BU	PROMO_NAME	[PROMO_ID].[NAME]	S_LOY_PROMO	NAME
S_LOY_PROMO_BU	PROMO_NUM	[PROMO_ID].[PROMO_NUM]	S_LOY_PROMO	PROMO_NUM
S_LOY_TXN_BU	TXN_NUM	[TXN_ID].[TXN_NUM]	S_LOY_TXN	TXN_NUM
S_LOY_TXN_BU	TXN_STATUS_CD	[TXN_ID].[STATUS_CD]	S_LOY_TXN	STATUS_CD

Table 29. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_LOY_TXN_BU	TXN_SUB_TYPE_CD	[TXN_ID].[SUB_TYPE_CD]	S_LOY_TXN	SUB_TYPE_CD
S_LOY_TXN_BU	TXN_TYPE_CD	[TXN_ID].[TYPE_CD]	S_LOY_TXN	TYPE_CD
S_OPTY_POSTN	NEW_LOAN_FLG	[OPTY_ID].[NEW_LOAN_FLG]	S_OPTY	NEW_LOAN_FLG
S_ORG_BU	ORG_REF_CUST_FLG	[ORG_ID].[REFERENCE_CUST_FLG]	S_ORG_EXT	REFERENCE_CUST_FLG
S_ORG_DIST_LST	OU_ID	[ORG_PROD_ID].[OU_ID]	S_ORG_PROD	OU_ID
S_ORG_GROUP_BU	OG_GROUP_NAME	[ORG_GROUP_ID].[GROUP_NAME]	S_ORG_GROUP	GROUP_NAME
S_ORG_GROUP_BU	OG_GROUP_TYPE_CD	[ORG_GROUP_ID].[GROUP_TYPE_CD]	S_ORG_GROUP	GROUP_TYPE_CD
S_ORG_GROUP_BU	OG_NAME	[ORG_GROUP_ID].[NAME]	S_ORG_GROUP	NAME
S_ORGGRP_POSTN	GRP_GROUP_NAME	[ORGGRP_ID].[GROUP_NAME]	S_ORG_GROUP	GROUP_NAME
S_PARTY_GRP_BU	PG_NAME	[PARTY_GROUP_ID].[NAME]	S_PARTY_GROUP	NAME
S_PROD_STYL_TNT	SETUP_STYLE_CD	[PROP_STYLE_ID].[SETUP_STYLE_CD]	S_PROP_STYL_TNT	SETUP_STYLE_CD
S_PROJ_BU	PROJ_NAME	[PROJ_ID].[NAME]	S_PROJ	NAME
S_PROJ_BU	PROJ_STATUS_CD	[PROJ_ID].[STATUS_CD]	S_PROJ	STATUS_CD
S_PROJ_BU	PROJ_TYPE_CD	[PROJ_ID].[PROJ_TYPE_CD]	S_PROJ	PROJ_TYPE_CD
S_PSP_PROC_BU	VOD_NAME	[VOD_ID].[VOD_NAME]	S_VOD	VOD_NAME
S_QUOTE_POSTN	QUOTE_NUM	[QUOTE_ID].[QUOTE_NUM]	S_DOC_QUOTE	QUOTE_NUM
S_SRC_BU	SRC_NAME	[SRC_ID].[NAME]	S_SRC	NAME
S_SRV_REQ_BU	SR_AREA	[SRV_REQ_ID].[SR_AREA]	S_SRV_REQ	SR_AREA
S_SRV_REQ_BU	SR_CST_NUM	[SRV_REQ_ID].[SR_CST_NUM]	S_SRV_REQ	SR_CST_NUM

Table 29. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_SRV_REQ_BU	SR_NUM	[SRV_REQ_ID].[SR_NUM]	S_SRV_REQ	SR_NUM
S_SRV_REQ_BU	SR_SEV_CD	[SRV_REQ_ID].[SR_SEV_CD]	S_SRV_REQ	SR_SEV_CD
S_SRV_REQ_BU	SR_STAT_ID	[SRV_REQ_ID].[SR_STAT_ID]	S_SRV_REQ	SR_STAT_ID
S_SRV_REQ_BU	SR_SUB_STAT_ID	[SRV_REQ_ID].[SR_SUB_STAT_ID]	S_SRV_REQ	SR_SUB_STAT_ID
S_SRV_REQ_BU	SR_TITLE	[SRV_REQ_ID].[SR_TITLE]	S_SRV_REQ	SR_TITLE
S_SRV_REQ_BU	SR_TYPE_CD	[SRV_REQ_ID].[SR_TYPE_CD]	S_SRV_REQ	SR_TYPE_CD
S_USERLIST_BU	UL_NAME	[USERLIST_ID].[NAME]	S_USERLIST	NAME
T_MASTER_BU	MASTER_ALIAS	[MASTER_ID].[ALIAS]	T_MASTER	ALIAS
T_MASTER_BU	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME
T_MASTER_DETAIL	DETAIL_SX2_NAME	[DETAIL_SX2_ID].[SX2_NAME]	T_DETAIL_SX2	SX2_NAME
T_MASTER_DETAIL	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME
T_MASTER_PER	MASTER_ALIAS	[MASTER_ID].[ALIAS]	T_MASTER	ALIAS
T_MASTER_POSTN	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME

Denormalized Columns for 7.0.4 Siebel Industry Solutions

Table 30 lists columns that are denormalized during upgrades from Release 7.0.4 Siebel Industry Solutions to Release 7.8 Siebel Industry Applications.

Table 30. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ACCNT_POSTN	ACCNT_TYPE_CD	[OU_EXT_ID].[ACCNT_TYPE_CD]	S_ORG_EXT	ACCNT_TYPE_CD
S_ACCNT_POSTN	CLIENT_FLG	[OU_EXT_ID].[CLIENT_FLG]	S_ORG_EXT	CLIENT_FLG
S_ACCNT_POSTN	FACILITY_FLG	[OU_EXT_ID].[FACILITY_FLG]	S_ORG_EXT	FACILITY_FLG
S_ACCNT_POSTN	INVSTR_FLG	[OU_EXT_ID].[INVSTR_FLG]	S_ORG_EXT	INVSTR_FLG
S_ACCNT_POSTN	ORG_REF_CUST_FLG	[OU_EXT_ID].[REFERENCE_CUST_FLG]	S_ORG_EXT	REFERENCE_CUST_FLG
S_ACCNT_POSTN	SRV_PROVDR_FLG	[OU_EXT_ID].[SRV_PROVDR_FLG]	S_ORG_EXT	SRV_PROVDR_FLG
S_ACT_CAL_RSRC	ACT_APPT_REPT_FLG	[ACTIVITY_ID].[APPT_REPT_FLG]	S_EVT_ACT	APPT_REPT_FLG
S_ACT_CAL_RSRC	ACT_APPT_RPTEND_DT	[ACTIVITY_ID].[APPT_REPT_END_DT]	S_EVT_ACT	APPT_REPT_END_DT
S_ACT_CAL_RSRC	ACT_CAL_TYPE_CD	[ACTIVITY_ID].[CAL_TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ACT_CAL_RSRC	ACT_TEMPLATE_FLG	[ACTIVITY_ID].[TEMPLATE_FLG]	S_EVT_ACT	TEMPLATE_FLG
S_ACT_CAL_RSRC	ACT_TODO_PLNEND_DT	[ACTIVITY_ID].[TODO_PLAN_END_DT]	S_EVT_ACT	TODO_PLAN_END_DT
S_ACT_CAL_RSRC	ACT_TODO_PLNSTRTDT	[ACTIVITY_ID].[TODO_PLAN_START_DT]	S_EVT_ACT	TODO_PLAN_START_DT
S_ACT_EMP	ACT_CAL_TYPE_CD	[ACTIVITY_ID].[CAL_TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ASSET_BU	TYPE_CD	[ASSET_ID].[TYPE_CD]	S_ASSET	TYPE_CD
S_ASSET_POSTN	ASSET_NUM	[ASSET_ID].[ASSET_NUM]	S_ASSET	ASSET_NUM

Table 30. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ASSET_POSTN	TYPE_CD	[ASSET_ID].[TYPE_CD]	S_ASSET	TYPE_CD
S_CASE_BU	CASE_NAME	[CASE_ID].[NAME]	S_CASE	NAME
S_CASE_BU	CASE_STATUS_CD	[CASE_ID].[STATUS_CD]	S_CASE	STATUS_CD
S_CASE_BU	CASE_TYPE_CD	[CASE_ID].[TYPE_CD]	S_CASE	TYPE_CD
S_CONTACT_BU	AGENT_FLG	[CONTACT_ID].[AGENT_FLG]	S_CONTACT	AGENT_FLG
S_CONTACT_BU	CON_EMP_FLG	[CONTACT_ID].[EMP_FLG]	S_CONTACT	EMP_FLG
S_CONTACT_BU	CON_MI D_NAME	[CONTACT_ID].[MID_NAME]	S_CONTACT	MI D_NAME
S_CONTACT_BU	MEMBER_FLG	[CONTACT_ID].[MEMBER_FLG]	S_CONTACT	MEMBER_FLG
S_CONTACT_BU	PROVI DER_FLG	[CONTACT_ID].[PROVIDER_FLG]	S_CONTACT	PROVI DER_FLG
S_I NSCLM_BU	I NSCLAI M_NUM	[INSCLM_ID].[INSCLAIM_NUM]	S_I NS_CLAI M	I NSCLAI M_NUM
S_I NSCLM_BU	REV_NUM	[INSCLM_ID].[REVISION_NUM]	S_I NS_CLAI M	REVI SI ON_NUM
S_I NSCLMEL_BU	SEQ_NUM	[INSCLM_ELMNT_ID].[SEQ_NUM]	S_I NSCLM_ELMNT	SEQ_NUM
S_I NSCLMEL_BU	TYPE_CD	[INSCLM_ELMNT_ID].[TYPE_CD]	S_I NSCLM_ELMNT	TYPE_CD
S_LOY_MEM_BU	MEM_NUM	[MEMBER_ID].[MEM_NUM]	S_LOY_MEMBER	MEM_NUM
S_LOY_MEM_BU	MEM_TYPE_CD	[MEMBER_ID].[MEM_TYPE_CD]	S_LOY_MEMBER	MEM_TYPE_CD
S_LOY_MEM_PSTN	MEM_NUM	[MEMBER_ID].[MEM_NUM]	S_LOY_MEMBER	MEM_NUM
S_LOY_MEM_PSTN	MEM_TYPE_CD	[MEMBER_ID].[MEM_TYPE_CD]	S_LOY_MEMBER	MEM_TYPE_CD
S_LOY_PROG_BU	PROG_NAME	[PROG_ID].[NAME]	S_LOY_PROGRAM	NAME
S_LOY_PROMO_BU	PROMO_NAME	[PROMO_ID].[NAME]	S_LOY_PROMO	NAME

Table 30. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_LOY_PROMO_BU	PROMO_NUM	[PROMO_ID].[PROMO_NUM]	S_LOY_PROMO	PROMO_NUM
S_LOY_TXN_BU	TXN_NUM	[TXN_ID].[TXN_NUM]	S_LOY_TXN	TXN_NUM
S_LOY_TXN_BU	TXN_STATUS_CD	[TXN_ID].[STATUS_CD]	S_LOY_TXN	STATUS_CD
S_LOY_TXN_BU	TXN_SUB_TYPE_CD	[TXN_ID].[SUB_TYPE_CD]	S_LOY_TXN	SUB_TYPE_CD
S_LOY_TXN_BU	TXN_TYPE_CD	[TXN_ID].[TYPE_CD]	S_LOY_TXN	TYPE_CD
S_OPTY_POSTN	CONSUMER_OPTY_FLG	[OPTY_ID].[CONSUMER_OPTY_FLG]	S_OPTY	CONSUMER_OPTY_FLG
S_OPTY_POSTN	NEW_LOAN_FLG	[OPTY_ID].[NEW_LOAN_FLG]	S_OPTY	NEW_LOAN_FLG
S_OPTY_POSTN	OPTY_CLOSED_FLG	[OPTY_ID].[CLOSED_FLG]	S_OPTY	CLOSED_FLG
S_OPTY_POSTN	SECURE_FLG	[OPTY_ID].[SECURE_FLG]	S_OPTY	SECURE_FLG
S_ORG_BU	ORG_FACILITY_FLG	[ORG_ID].[FACILITY_FLG]	S_ORG_EXT	FACILITY_FLG
S_ORG_BU	ORG_INVSTR_FLG	[ORG_ID].[INVSTR_FLG]	S_ORG_EXT	INVSTR_FLG
S_ORG_BU	ORG_REF_CUST_FLG	[ORG_ID].[REFERENCE_CUST_FLG]	S_ORG_EXT	REFERENCE_CUST_FLG
S_ORG_BU	ORG_SRV_PROVDR_FLG	[ORG_ID].[SRV_PROVDR_FLG]	S_ORG_EXT	SRV_PROVDR_FLG
S_ORG_DIST_LST	OU_ID	[ORG_PROD_ID].[OU_ID]	S_ORG_PROD	OU_ID
S_ORG_GROUP_BU	OG_GROUP_NAME	[ORG_GROUP_ID].[GROUP_NAME]	S_ORG_GROUP	GROUP_NAME
S_ORG_GROUP_BU	OG_GROUP_TYPE_CD	[ORG_GROUP_ID].[GROUP_TYPE_CD]	S_ORG_GROUP	GROUP_TYPE_CD
S_ORG_GROUP_BU	OG_NAME	[ORG_GROUP_ID].[NAME]	S_ORG_GROUP	NAME
S_ORGGRP_POSTN	GROUP_TYPE_CD	[ORGGRP_ID].[GROUP_TYPE_CD]	S_ORG_GROUP	GROUP_TYPE_CD

Table 30. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_ORGGRP_POSTN	GRP_GROUP_NAME	[ORGGRP_ID]. [GROUP_NAME]	S_ORG_GROUP	GROUP_NAME
S_PARTY_GRP_BU	PG_NAME	[PARTY_GROUP_ID]. [NAME]	S_PARTY_GROUP	NAME
S_POS_BU	POS_NUM	[POS_ID].[POS_] NUM]	S_POS	POS_NUM
S_POSTN_CON	AGENT_FLG	[CON_ID].[AGENT_] FLG]	S_CONTACT	AGENT_FLG
S_POSTN_CON	MEMBER_FLG	[CON_ID].[MEMBER_] _FLG]	S_CONTACT	MEMBER_FLG
S_POSTN_CON	PROVI DER_FLG	[CON_ID]. [PROVIDER_FLG]	S_CONTACT	PROVI DER_FLG
S_PROD_I NT_BU	PROD_CD	[PROD_INT_ID]. [PROD_CD]	S_PROD_I NT	PROD_CD
S_PROD_STYL_TNT	SETUP_STYLE_CD	[PROP_STYLE_ID]. [SETUP_STYLE_CD]	S_PROP_STYL_ TNT	SETUP_STYLE_CD
S_PROJ_BU	PROJ_NAME	[PROJ_ID].[NAME]	S_PROJ	NAME
S_PROJ_BU	PROJ_STATUS_CD	[PROJ_ID].[STATUS_] _CD]	S_PROJ	STATUS_CD
S_PROJ_BU	PROJ_TYPE_CD	[PROJ_ID].[PROJ_] TYPE_CD]	S_PROJ	PROJ_TYPE_CD
S_PSP_PROC_BU	VOD_NAME	[VOD_ID].[VOD_] NAME]	S_VOD	VOD_NAME
S_QUOTE_POSTN	QUOTE_NUM	[QUOTE_ID]. [QUOTE_NUM]	S_DOC_QUOTE	QUOTE_NUM
S_SRC_BU	SRC_NAME	[SRC_ID].[NAME]	S_SRC	NAME
S_SRV_REQ_BU	SR_AREA	[SRV_REQ_ID].[SR_] AREA]	S_SRV_REQ	SR_AREA
S_SRV_REQ_BU	SR_CST_NUM	[SRV_REQ_ID].[SR_] CST_NUM]	S_SRV_REQ	SR_CST_NUM
S_SRV_REQ_BU	SR_NUM	[SRV_REQ_ID].[SR_] NUM]	S_SRV_REQ	SR_NUM
S_SRV_REQ_BU	SR_SEV_CD	[SRV_REQ_ID].[SR_] SEV_CD]	S_SRV_REQ	SR_SEV_CD
S_SRV_REQ_BU	SR_STAT_I D	[SRV_REQ_ID].[SR_] STAT_ID]	S_SRV_REQ	SR_STAT_I D

Table 30. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_SRV_REQ_BU	SR_SUB_STAT_ID	[SRV_REQ_ID].[SR_SUB_STAT_ID]	S_SRV_REQ	SR_SUB_STAT_ID
S_SRV_REQ_BU	SR_TITLE	[SRV_REQ_ID].[SR_TITLE]	S_SRV_REQ	SR_TITLE
S_SRV_REQ_BU	SR_TYPE_CD	[SRV_REQ_ID].[SR_TYPE_CD]	S_SRV_REQ	SR_TYPE_CD
S_USERLIST_BU	UL_NAME	[USERLIST_ID].[NAME]	S_USERLIST	NAME
T_MASTER_BU	MASTER_ALIAS	[MASTER_ID].[ALIAS]	T_MASTER	ALIAS
T_MASTER_BU	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME
T_MASTER_DETAIL	DETAIL_SX2_NAME	[DETAIL_SX2_ID].[SX2_NAME]	T_DETAIL_SX2	SX2_NAME
T_MASTER_DETAIL	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME
T_MASTER_PER	MASTER_ALIAS	[MASTER_ID].[ALIAS]	T_MASTER	ALIAS
T_MASTER_POSTN	MASTER_NAME	[MASTER_ID].[NAME]	T_MASTER	NAME

Denormalized Columns for 7.5.2 Siebel Industry Applications

Table 31 lists columns that are denormalized during upgrades from Release 7.5.2 of Siebel Industry Applications to Release 7.8 of Siebel Industry Applications provided by Oracle.

Table 31. Columns Denormalized During Upgrades from Release 7.5.2 Siebel Industry Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_LOY_MEM_BU	MEM_NUM	[MEMBER_ID].[MEM_NUM]	S_LOY_MEMBER	MEM_NUM
S_LOY_MEM_BU	MEM_TYPE_CD	[MEMBER_ID].[MEM_TYPE_CD]	S_LOY_MEMBER	MEM_TYPE_CD
S_LOY_MEM_PSTN	MEM_NUM	[MEMBER_ID].[MEM_NUM]	S_LOY_MEMBER	MEM_NUM

Table 31. Columns Denormalized During Upgrades from Release 7.5.2 Siebel Industry Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_LOY_MEM_PSTN	MEM_TYPE_CD	[MEMBER_ID].[MEM_TYPE_CD]	S_LOY_MEMBER	MEM_TYPE_CD
S_LOY_PROG_BU	PROG_NAME	[PROG_ID].[NAME]	S_LOY_PROGRAM	NAME
S_LOY_PROMO_BU	PROMO_NAME	[PROMO_ID].[NAME]	S_LOY_PROMO	NAME
S_LOY_PROMO_BU	PROMO_NUM	[PROMO_ID].[PROMO_NUM]	S_LOY_PRO	PROMO_NUM
S_LOY_TXN_BU	TXN_NUM	[TXN_ID].[TXN_NUM]	S_LOY_TXN	TXN_NUM
S_LOY_TXN_BU	TXN_STATUS_CD	[TXN_ID].[STATUS_CD]	S_LOY_TXN	STATUS_CD
S_LOY_TXN_BU	TXN_SUB_TYPE_CD	[TXN_ID].[SUB_TYPE_CD]	S_LOY_TXN	SUB_TYPE_CD
S_LOY_TXN_BU	TXN_TYPE_CD	[TXN_ID].[TYPE_CD]	S_LOY_TXN	TYPE_CD
S_POS_BU	POS_NUM	[POS_ID].[POS_NUM]	S_POS	POS_NUM
S_PROD_STYL_TNT	SETUP_STYLE_CD	[PROP_STYLE_ID].[SETUP_STYLE_CD]	S_PROP_STYL_TNT	SETUP_STYLE_CD
S_PSP_PROC_BU	VOD_NAME	[VOD_ID].[VOD_NAME]	S_VOD	VOD_NAME
S_QUOTE_POSTN	QUOTE_NUM	[QUOTE_ID].[QUOTE_NUM]	S_DOC_QUOTE	QUOTE_NUM

C

Siebel Upgrade Files

This appendix lists the files that are used to perform a development or production upgrade to Oracle's Siebel Business Applications, Release 8.0, and it lists the tables that are amended during PRET processing. This appendix contains the following topics:

- [Siebel 8.0 z/OS Upgrade Files on page 311](#)
- [Tables Amended During PRET Unload Processing on page 318](#)
- [PRET Members Generated By Pretedit.txt on page 320](#)
- [Tables Amended During PRET Processing on page 321](#)

Siebel 8.0 z/OS Upgrade Files

When you run the Siebel Upgrade Wizard on the midtier, it generates files that are used to perform the development and production upgrade. You then transfer these files to the z/OS host, where they are placed in staging data sets. Several upgrade files are also generated on the z/OS host.

[Table 32](#) lists each of the upgrade files that are generated, the name of the file on the midtier (if applicable) and on the z/OS host, the phase of the upgrade when the file is generated, the upgrade path to which the file applies, and a brief description.

Table 32. Siebel 8.0 z/OS Upgrade Files

Midtier File Name	z/OS File Name	Phase transferred to host	Description	Upgrade Paths
job0.txt	JOB0	ftp_stg	Contains REXX code and panels	All
SIEBEL.load.xmit	load.xmit	ftp_stg	Load modules	All
sbllog.txt	SBLLOG.LOADFILE	ftp_stg	Initial log file entries	All
SIEBEL.sp.dbrmlib.xmit	sp.dbrmlib.xmit	ftp_stg	Dbrm modules	All
SIEBEL.sp.spddl.xmit	sp.spddl.xmit	ftp_stg	Stored procedure DDL	All

Table 32. Siebel 8.0 z/OS Upgrade Files

Midtier File Name	z/OS File Name	Phase transferred to host	Description	Upgrade Paths
job1.txt	VSTG0000 VSTG0000 is copied into one of the following VSTG* files on the z/OS host, depending on your upgrade path: VSTG0002 VSTG0020 VSTG0021 VSTG0022 VSTG0003 VSTG0005 VSTG0006 VSTG0040 VSTG0041	ftp_stg	Generic install.jcl and help panels	All
siebin01.jcl	VSTG0001 VSTG0001 is copied into one of the following VSTG* files on the z/OS host, depending on your upgrade path: VSTG0011 VSTG0030 VSTG0031 VSTG0032 VSTG0012 VSTG0014 VSTG0015 VSTG0050 VSTG0051	ftp_stg	Install.jcl specific to each upgrade path	All
siebproc.jcl	VSTG0070	ftp_stg	JCL PROC members	All
filelist.txt	VSTG0075	Pause #1	List of files	All
dedup.jcl	VSTG0080	syncdd	Intersection Table Maintenance/Dedup process	SIA 621, SIA704, SIA752, HOR 752

Table 32. Siebel 8.0 z/OS Upgrade Files

Midtier File Name	z/OS File Name	Phase transferred to host	Description	Upgrade Paths
dedup_prod.jcl	VSTG0081	syncdd	Intersection Table Maintenance/Dedup process	SIA 621, SIA704, SIA752, HOR752
synctab.jcl	VSTG0085	syncdd	Table Synchronization	All
syncidx.sql	VSTG0087	syncdd	Index Synchronization	All
schema.staging.db.sql	VSTG0090	ftp_stg	Staging schema databases	All
schema.staging.tbsp.sql	VSTG0091	ftp_stg	Staging schema table spaces	All
schema.staging.tbl.sql	VSTG0092	ftp_stg	Staging schema tables	All
schema.staging.uind.sql	VSTG0093	ftp_stg	Staging schema unique indexes	All
schema.staging.nuind.sql	VSTG0094	ftp_stg	Staging schema NPIs	All
schema.staging.oind.sql	VSTG0095	ftp_stg	Staging schema obsolete indexes	All
schema.staging.grt.sql	VSTG0096	ftp_stg	Staging schema grants	All
schema.db.sql	VSTG0100	Pause#2	Target schema databases	All
schema.tbsp.sql	VSTG0101	Pause#2	Target schema table spaces	All
schema.tbl.sql	VSTG0102	Pause#2	Target schema tables	All
schema.grt.sql	VSTG0103	Pause#2	Target schema grants	All
schema.uind.sql	VSTG0104	Pause#2	Target schema unique indexes	All
schema.nuind.sql	VSTG0105	Pause#2	Target schema NPIs	All
schema.oind.sql	VSTG0106	Pause#2	Target schema obsolete indexes	All

Table 32. Siebel 8.0 z/OS Upgrade Files

Midtier File Name	z/OS File Name	Phase transferred to host	Description	Upgrade Paths
N/A	VSTG1010 This file is created on the z/OS host. It contains CREATE INDEX statements extracted from VSTG0104 and VSTG0105.	N/A	Generates index REBUILD control statements for the target environment.	All
N/A	VSTG1111 This file is dynamically built on the z/OS host.	N/A	Maintains a list of the additive changes that are applied to the staging database. This list is updated dynamically as changes are applied.	All
N/A	VSTG1112 This file is dynamically built on the z/OS host.	N/A	Contains the additive changes applied to the staging database during the preupgrade phase.	All
scindx.sql	VSTG0110	Pause#3	DDL for secondary indexes	All
tmptable.sql	VSTG0119	Pause#2	Staging. Common TMPTABLES (tmptable.ctf)	All
tmptable.sql	VSTG0120	Pause#2	Target. Common TMPTABLES (tmptable.ctf)	All
tmptable1.sql	VSTG0121	ftp_stg	Staging. Logging for Unload jobs and TMP_ADDR table (no storage.ctf file)	All

Table 32. Siebel 8.0 z/OS Upgrade Files

Midtier File Name	z/OS File Name	Phase transferred to host	Description	Upgrade Paths
tmptable2.sql	VSTG0122	ftp_stg	Target. Logging for Unload jobs and TMP_ADDR table (no storage.ctl file)	All
drop_view.sql	VSTG0130	ftp_stg	Apply drop view before Non-Additive	All
ddlview_sql	VSTG0131	ftp_stg	Apply create view after Non-Additive	All
siebel.translate.iconv	VSTG0150	Pause #1	Program to convert code pages for each language	All
pregen.txt	VSTG0200	ftp_stg	JCL (genclbf, gentrgd, geneimd)	All
pret.jcl	VSTG0210	Pause #1	PRET SQL	All
pret_prod.jcl	VSTG0211	Pause #1	PRET JCL	All
pret_sia.jcl	VSTG0220	Pause #1	PRET SQL	SIA621, SIA704, SIA752, SIA77, SIA78
pret_sia_prod.jcl	VSTG0221	Pause #1	PRET JCL	SIA621, SIA704, SIA752, SIA77, SIA78
unload.ldc	VSTG0300	Pause #2	Unload control cards	All
load.ldc	VSTG0310	Pause #2	Load control cards	All
preschm.jcl	VSTG0400	Pause #1	PRESCHM SQL	All
preschm_prod.jcl	VSTG0401	Pause #1	PRESCHM JCL	All
preschm_ldc	VSTG0402	Pause #1	Load control cards for PRESCHM	SIA621

Table 32. Siebel 8.0 z/OS Upgrade Files

Midtier File Name	z/OS File Name	Phase transferred to host	Description	Upgrade Paths
preschm_sia.jcl	VSTG0410	Pause #1	PRESCHM SQL	SIA621, SIA704, SIA752, SIA77, SIA78
preschm_sia_prod.jcl	VSTG0411	Pause #1	PRESCHM JCL	SIA621, SIA704, SIA752, SIA77, SIA78
prod_configurator.jcl	VSTG0500	Pause #1	Prod Configurator SQL	SIA621
prod_configurator_prod.jcl	VSTG0501	Pause #1	Prod Configurator JCL	SIA621
prod_configurator_sia.jcl	VSTG0510	Pause #1	Prod Configurator SQL	SIA621, SIA704
prod_configurator_sia_prod.jcl	VSTG0511	Pause #1	Prod Configurator JCL	SIA621, SIA704
upg_iss.jcl	VSTG0600	Pause #1	UPGISS JCL	All
upg_iss_prod.jcl	VSTG0601	Pause #1	UPGISS SQL	All
gen_primary1.jcl	VSTG0700	Pause #1	Gen Primary part 1 - SQL	All
gen_primary1_prod.jcl	VSTG0701	Pause #1	Gen Primary part 1 - JCL	All
gen_primary2.jcl	VSTG0702	Pause #1	Gen Primary part 2 - SQL	All
gen_primary2_prod.jcl	VSTG0703	Pause #1	Gen Primary part 2 - JCL	All
gen_primary3.jcl	VSTG0704	Pause #1	Gen Primary part 3 - SQL	All
gen_primary3_prod.jcl	VSTG0705	Pause #1	Gen Primary part 3 - JCL	All
gen_primary4.jcl	VSTG0706	Pause #1	Gen Primary part 4 - SQL	All
gen_primary4_prod.jcl	VSTG0707	Pause #1	Gen Primary part 4 - JCL	All

Table 32. Siebel 8.0 z/OS Upgrade Files

Midtier File Name	z/OS File Name	Phase transferred to host	Description	Upgrade Paths
echannelAccountDivMatch_fins.sql	VSTG0804	Pause #1	eChannel	SIA621
echannelEmpContactMatch_fins.sql	VSTG0805	Pause #1	eChannel	SIA621
echannel_merge_contact.sql	VSTG0800	Pause #1	eChannel	SIA621
echannel_merge_org_int.sql	VSTG0801	Pause #1	eChannel	SIA621
hhmignot.sql	VSTG0850	Pause #1	Household	SIA621, SIA704, SIA752, SIA77, SIA78
hhmigpop.sql	VSTG0851	Pause #1	Household	SIA621, SIA704, SIA752, SIA77, SIA78
household_mig_Fins.jcl	VSTG0852	Pause #1	Household	SIA621, SIA704, SIA752, SIA77, SIA78
household_mig_Fins_prod.jcl	VSTG0853	Pause #1	Household	SIA621, SIA704, SIA752, SIA77, SIA78
rpt_dup_addr_rowids.sql	VSTG0861	ftp_stg	Gen Dup Addr Report SQL	All
rpt_dup_addr_names.sql	VSTG0860	ftp_stg	Gen Dup Addr Report SQL	All
rpt_dup_addr_names.sql	VSTG0860	Pause #1	Gen Dup Addr Report SQL	SIA621

Table 32. Siebel 8.0 z/OS Upgrade Files

Midtier File Name	z/OS File Name	Phase transferred to host	Description	Upgrade Paths
schema.additive.sql	VSTG1000	Pause #1	Staging additive changes	All
N/A	VSTG1001 This file is a copy of VSTG1000; it is created on the z/OS host.	N/A	Target additive changes	All

Tables Amended During PRET Unload Processing

The unload job control cards for specific tables have been modified so that during PRET (pre-table) processing, the data in the tables is modified during the table unload process instead of being modified after the data has been loaded into the target table.

Table 33 lists the source tables containing the data that is modified during unload processing, the macro that performs the modifications, and the relevant upgrade paths.

Table 33. Tables Amended During PRET Unload Processing

PRET Tables Modified During the Table Unload Process	Macro	HOR Paths	SIA Paths
S_APPL_WEB_TMPL	PTH0062	HOR704	SIA704, SIA752
S_APPL_WTMPL_IT	PTH0064	HOR704	SIA704, SIA752
S_POSTN_CON	PTS0100	n/a	SIA704 (FINS704 only)
S_CONTROL	PTH0222	All (except SG704)	All
S_PCONTROL	PTS0223	HOR752, HOR77	SIA752
S_FN_CRDT_RPT	PTS0224	HOR752, HOR77	SIA752
S_ASGN_RULE_GRP	PTS0225	HOR752, HOR77	SIA752
S_REGION	PTS0227	n/a	SIA 704 (FINS704 only), SIA752
S_REGION	PTS0228	n/a	SIA752
S_ENTLMNT_ITEM_FEE	PTS0270	n/a	SIS704
S_ENTLMNT_ITEM_FEE	PTS0271	n/a	SIS704
S_CURRCLM_PER	PTH0289	HOR621	SIA621, SIS63
S_EVT_ACT	PTH0405	HOR621	SIA621, SIS63
S_QTA_OBJCRT	PTH0832	HOR621	SIA621, SIS63
S_QTA_OBJCRT_D	PTH0833	All and SG704	All
S_WF_PROC_FLOW	PTH1003	HOR621, HOR704	SIA621, SIS63, SIA704, SG704
S_WF_STEP	PTH1006	HOR621, HOR704	SIA621, SIS63, SIA704, SG704
S_ASSET_POSTN	PTS0301	n/a	SIS 704 only
S_ASSET_POSTN	PTS0302	n/a	SIS 704 only
S_ASSET_POSTN	PTS0303	n/a	SIS 704 only
S_ASSET_POSTN	PTS0304	n/a	SIS 704 only
S_ETL_TIME_DAY	PTS0313	n/a	SIA752
S_ETL_TIME_DAY	PTS0314	n/a	SIA752
S_DD_MEAS_ATTR	PTS0350	n/a	SIS63
S_CON_ADDR_U1	PTM0410	n/a	SIA621, SIS63, SIA704

Table 33. Tables Amended During PRET Unload Processing

PRET Tables Modified During the Table Unload Process	Macro	HOR Paths	SIA Paths
S_EXTDATA_TBL	PTS0500	n/a	SIA752
S_PAPL_WEB_TMPL	PTS0505	n/a	SIA752
S_NOTE_CON	PTM0520	n/a	SIA77
S_NOTE_PROD_INT	PTM0520	n/a	SIA77
S_NOTE_ACCNT	PTM0520	n/a	SIA77
S_DOCK_TXN_LOG	PTM0010	All Paths	All Paths
S_ESCL_REQ	PTM0010	All Paths	All Paths

NOTE: Unload and load jobs on tables that contain CLOB data are processed by the PTMCLOBx macros. These macros adjust the unload and load job control cards to handle any CLOB data in a table. The PTMCLOBx macros can be found in the *DSNHLQ.SIEBEL.EXEC* or in the VSTG0300 and the VSTG0310 staging data sets.

PRET Members Generated By Pretedit.txt

The Pretedit.txt file creates the partitioned data set (PDS) on the z/OS host that is used for PRET processing. The members in this PDS perform a number of tasks, for example, listing the source tables that contain CLOB columns, gathering information required for key processing, and deleting rows in specific tables. Data sets are generated for each upgrade path.

Table 34 shows the data set members created by the Pretedit.txt file, the objects amended by these members (the SQL statement that is run is contained in the member) and the upgrade path for which these members are generated.

Table 34. PRET Processing Members Created By the Pretedit.txt File

PDS Member Name	PRET Object Affected	SIA Paths
PRETLDIN	S_DOCK_TXN_LOG	All and SG704
PRETLDIN	S_ESCL_REQ	All and SG704
PRETLDIN	S_SRM_REQUEST	SIA621, SIS63, SG704
PRETCLBF	SQL/CLOB list, used by SBLCLOBU	All
PRETKEYS	SQL/Clustering Index Key structures	All

Tables Amended During PRET Processing

The PRET upgrade jobs perform operations on the target database tables listed in [Table 35](#). You might want to back up these tables before you start the upgrade.

[Table 35](#) shows the target tables amended by PRET upgrade processing, the type of amendment made (the table is either altered or dropped), and the upgrade path affected.

Table 35. Tables Changed During PRET Processing

Tables Amended During PRET Processing	Type of Change Made	SIA Paths
S_EMPLOYEE	Alter	SIA621, SIS63
S_INS_CLAIM	Alter	SIA621
S_ORG_GROUP	Alter	SIA621, SIS63
S_CON_ADDR	Alter	SIA704
S_FN_CRDT_RPT	Alter	SIA752
S_REGION	Alter	SIA704 (FINS704 only, n/a for SIS704), SIA752
S_EMP_TLR_LMTS	Drop	SIA621 only
S_FN_HLDVR_CTF	Drop	SIA621 only
S_FN_TXN_LMTS	Drop	SIA621 only

Index

Symbols

.srf file

See configuration file

Numerics

6.x

duplicate Row IDs, identifying and resolving 50, 59, 68
repository, running Repository Preparation Wizard 187
Siebel Configuration data, preparing for upgrade 106

7.7, planning upgrade

new database schema structure 74

7.x, planning upgrade

backup and recovery stages, planning 83
capacity planning 78
code page support, considering 76
IBM utilities, obtaining required 78
new database schema structure in 7.7 74
schedule, creating 83
security privileges, obtaining required 82
Siebel-scheduled mode or vendor-schedule mode 77
software and hardware, required 78
testing before production upgrade 76

A

adding

Siebel user 100

additive schema upgrade changes 24

AIX memory allocation segment space, securing 103

applets

7.x applet scripts, about migrating to 220
applet models, about and list of 207
customizations, migrating to applets and views 213
migrating applets to release 7.x 203
new applet migration, reviewing 216
scripts, about migrating to release 7.x 209, 220

Application scripts

7.x, about migrating to 209
migrating 223

attachments

file system attachments, updating 249

B

backing up

development database 100

backup and recovery stages, planning 83

Business Component scripts

7.x, about migrating to 209
migrating 223

business components

custom business component configurations, migrating 233
join conflicts, resolving 228
visibility properties, changing 253

Business Services scripts, migrating 223

C

Call Center, locating user logins that require resolution of duplicates 256

call report attachments

file names, updating 256, 260
updating (procedure) 256, 260

capacity planning 78

Catalog visibility

See visibility modes, setting for access control

client configuration, migrating to Web client

See also postupgrade tasks; preupgrade tasks
about 203

applet and view models 207

applet migration, reviewing 216

applet scripts, migrating to 7.x 220

applets and views, migrating, about and wizards 203

client-side interfaces, migration tasks 210

hidden properties, exposing 212

migrating Business Components, Business Services, and Application scripts 223

outbound COM interfaces, migrating 224

scripts, analyzing for migration 218

scripts, migration tasks 209

views, remigrating 217

client-side interfaces

outbound COM interfaces 210

outbound COM interfaces, migrating 224

clustered indexes

considerations for 85

code page

considering support 76

columns, denormalized

- 6.2.1 Siebel Business Applications 282
- 6.2.1 Siebel Financial Services applications 273
- 6.3 Siebel Industry Solutions 289
- 7.0.4 Siebel Business Applications 299
- 7.0.4 Siebel Financial Services applications 301
- 7.0.5 Siebel Industry Solutions 305
- 7.5.2 Siebel Financial Services applications 309

configuration file

- new .srf file, compiling 200
- new custom configuration file, producing 200

creating the staging database 137**custom business component configurations**

migrating 233

custom database schema

license keys, adding new 237

custom objects, remapping opportunity product migration objects 259**Customer Repository, preparing for merge**

New Customer repository, preparing to import 198

D**database configuration, verifying**

Siebel database expected growth during upgrade (table) 84

database schema

new structure in 7.7 74
testing before production upgrade 76

database template file, regenerating 248**database, preparing for upgrade**

AIX memory allocation segment space, securing 103
estimating database size 84
New Customer Repository, preparing to import 198

DB2, upgrade planning

new database schema structure in 7.7 74

deleting

duplicate EIM mappings 247

denormalized columns

See columns, denormalized

development environment, postupgrade tasks

database template file, regenerating 248
EIM mappings, deleting duplicate 247
obsolete tables, reapplying custom

extensions 245

development environment, preupgrade tasks

virtual memory, about allocating sufficient 249

development environment, upgrading

6.x duplicate Row IDs, identifying and resolving 50, 59, 68

custom business component configurations, migrating 233

foreground performance, increasing 192

repositories for an ASCII database, merging 193

repositories for an DB2 EBCDIC database, merging 196

repository merge, about 191

repository merge, categories of object differences 189

repository merge, optimizing performance 191

Repository Preparation Wizard, running 187

schema, synchronizing 231, 232

development upgrade

database configuration, verifying 87

DB2 software, upgrading 84

development database, backing up 100

development repository projects, checking

in 110

EIM tables, preparing for upgrade 111

intersection table maintenance and deduplication, preparing for 113

requirements 109

schema, preparing storage layout 88

Siebel Configurator data 6.x, preparing for upgrade 106

Siebel user, adding 100

storage control file, modifying 89

storage control file, reviewing 97

storage control file, validating 95

storage groups, creating 88

E**eChannel, about preparing for upgrade 104****EIM mapping**

deleting duplicates 247

EIM tables

preparing for upgrade 111

F**file system**

attachments, updating 249

Forecasting data, preparing for upgrade 104

G

- generating upgrade files** 149
- guide**
 - naming and typographical conventions 16

H

- hardware, obtaining and software** 78
- hidden properties, exposing** 212
- household notes, updating file system** 258

I

- IBM DB2 UDB, preparing database for upgrade**
 - clustered indexes, considerations for 85
- IBM utilities**
 - obtaining required 78
- importing**
 - development repository and schema definition 117
- intersection tables**
 - preparing for maintenance and deduplication 113

J

- join conflicts, resolving** 228

L

- license keys**
 - custom database schema, adding new keys 237
- log files**
 - archiving manually 242
 - SiebSrvr log files, about saving 119

M

- migrating, client configuration to Web client**
 - See also* postupgrade tasks; preupgrade tasks
 - about 203
 - applet and view models 207
 - applet migration, reviewing 216
 - applet scripts, migrating to 7.x 220
 - applets and views, migrating 203
 - client-side interfaces, migration tasks 210
 - hidden properties, exposing 212
 - migrating Business Components, Business Services, and Application scripts 223
 - outbound COM interfaces, migrating 224
 - scripts, analyzing for migration 218
 - scripts, migration tasks 209
 - views, remigrating 217

N

- naming conventions** 16

O

- objects**
 - opportunity product migration, remapping objects 259
- outbound COM interfaces, migrating** 224

P

- picklist visibility properties, changing** 254
- postupgrade tasks**
 - business components and join conflicts, resolving 228
 - call report attachments, updating 256, 260
 - dock object and rule definitions, verifying are correct 256
 - file system attachments, updating 249
 - household notes and attachments, updating file system 258
 - inbound workflows, updating 262
 - log files, manually archiving 242
 - new custom configuration file, producing 200
 - quotes 259
 - user logins, locating that require resolution of duplicates 256
 - visibility modes, setting for access control 253
- pre-upgrade tasks**
 - New Customer Repository, preparing to import 198
- preupgrade tasks**
 - AIX memory allocation segment space, securing 103
- production environment repositories**
 - preparing for upgrade 118
- production environment upgrade**
 - development environment, upgrading without 119
 - New Customer Repository, preparing to import 198
 - preparing production upgrade on UNIX 129
- production environment, upgrading**
 - intersection tables, maintaining 234
 - zSeries setup dataset, defining and allocating 141
 - zSeries upgrade environment, setting up 142
 - zSeries, requirements for upgrade 140
- production upgrade**
 - testing before 76
- production upgrade, prerequisites**

- about 117
 - development repository and schema
 - definition, importing 117
 - production environment repositories,
 - preparing for upgrade 118
- Q**
- quotes**
 - upgrade, preparing for 105
 - Quotes data, preparing for upgrade** 105
- R**
- recovery**
 - planning and backup stages 83
 - repositories**
 - production environment repositories,
 - preparing for upgrade 118
 - repositories for an ASCII database,
 - merging 193
 - repositories for an DB2 EBCDIC database,
 - merging 196
 - repository merge, about 191
 - repository merge, categories of object
 - differences 189
 - repository merge, increasing the foreground
 - performance 192
 - repository merge, optimizing
 - performance 191
 - Repository Preparation Wizard**
 - running on 6.x repository 187
 - Row IDs**
 - 6.x, identifying and resolving 50, 59, 68
- S**
- S_LIT files, upgrading to S_CB_ASSET_VER files** 249
 - S_SRC_PAYMENT table, about columns added** 107
 - schedule**
 - planning version 7.x 83
 - schema**
 - storage layout, preparing 88
 - synchronizing 231, 232
 - scripts**
 - applet scripts, about migrating to 7.x 220
 - business components, business services, and
 - applications scripts, migrating 223
 - migrating to release 7.x 209
 - migration, analyzing for 218
 - security**
 - obtaining required privileges 82
 - Siebel Application Integration (EAI), about**
 - using SAP and upgrading 20
 - Siebel Business Applications**
 - denormalized columns for 6.2.1 282
 - denormalized columns for 7.0.4 299
 - Siebel Configurator**
 - 6.x data, preparing for upgrade 106
 - Siebel data model**
 - S_SRC_PAYMENT Table, columns added
 - to 107
 - Siebel Database Schema, upgrading**
 - restarting upgrade 132
 - Siebel Database Storage Configurator**
 - control files, configuring 96
 - Siebel eChannel, about preparing for upgrade** 104
 - Siebel Enterprise Integration Manager**
 - deleting duplicate EIM mappings 247
 - Siebel Financial Services**
 - application tables, preparing for
 - upgrade 106
 - call report attachments, updating 256, 260
 - call report attachments, updating file
 - names 256, 260
 - denormalized columns for 6.2.1 273
 - denormalized columns for 7.0.4 301
 - denormalized columns for 7.5.2 309
 - household notes and attachments, updating
 - file system 258
 - opportunity product migration 259
 - Siebel Industry Solutions**
 - denormalized columns for 6.3 289
 - denormalized columns for 7.0.4 305
 - Siebel Quotes, about preparing upgrade for** 105
 - Siebel Software Configuration utility**
 - The Siebel Upgrade Wizard, about 36
 - Siebel Upgrade Wizard**
 - about 36
 - Siebel user**
 - adding 100
 - Siebel Workflow**
 - inbound workflow, updating 262
 - SiebSrvr log files, about saving** 119
 - software, obtaining and hardware** 78
 - .srf file**
 - See configuration file
 - staging database, creating** 137
 - storage control file**
 - modifying 89
 - reviewing 97
 - validating 95
 - storage groups**
 - creating 88

T**tables**

- EIM tables, preparing for upgrade 111
- intersection tables, preparing maintenance and deduplication 113
- obsolete tables, reapplying custom extensions 245

tables, preparing for upgrade

- clustered indexes, considerations 85
- DB2 UDB long columns, identifying for truncation 84, 198

target database, upgrading 167**testing**

- before production upgrade 76

typographical conventions 16**U****Unicode support** 20**upgrade environments** 21**upgrade files, generating** 149**Upgrade Planning Worksheet** 269**upgrade, planning for**

- resources 73, 265, 266, 267

user

- adding Siebel user 100

utilities

- IBM utilities, obtaining required 78

V**version 6.x**

- duplicate Row IDs, identifying and resolving 50, 59, 68
- repository, running Repository Preparation Wizard 187
- Siebel Configurator data, preparing for upgrade 106

version 7.7, planning upgrade

- new database schema structure 74

version 7.x, planning upgrade

- backup and recovery stages, planning 83
- capacity planning 78
- code page support, considering 76
- IBM utilities, obtaining required 78
- new database schema structure in 7.7 74
- schedule, creating 83

- security privileges, obtaining required 82
- Siebel-scheduled mode or vendor-schedule mode 77
- software and hardware, required 78
- testing before production upgrade 76

views

- customization, migrating to applets and views 213
- migrating to release 7.x 203
- remigrating views 217
- view models, about and list of 207
- visibility properties, changing 255

virtual memory

- development environment, about allocating sufficient 249

visibility modes, setting for access control

- about and Catalog visibility 253
- Auto Query Mode, picklists that start in (list of) 254
- business component visibility properties, changing 253
- business components, visibility type used (list of) 253
- picklists visibility properties, changing 254
- views visibility properties, changing 255
- views, visibility type used (list of) 255

W**Web Client Migration Wizard**

- customizations, migrating to applets and views 213

Web client scripting, about in release 7.x 209**workflows**

- inbound, updating 262
- test to production, transitioning 262

worksheet

- Upgrade Planning 269

Z**zSeries**

- setup dataset, defining and allocating 141
- upgrade environment, setting up 142
- upgrade requirements 140

