



Upgrade Guide for DB2 UDB for z/OS and OS/390

Version 7.7, Rev. C
May 2005

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

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

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

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

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

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

Proprietary Information

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

Contents

Chapter 1: What's New in This Release

Chapter 2: About Upgrade Topics

- About This Guide 19
- How the Upgrade Guide Is Organized 20
- About Topic Applicability 20
- About Terms Used in the Upgrade Guide 21
- About File Paths and Commands in the Upgrade Guide 22
- Important Upgrade Planning Resources 23

Chapter 3: How the Upgrade Works

- About Supported Upgrade Paths 27
- About Upgrade Environments 28
- About Midtier-Centric and Mainframe-Centric Development Environment Upgrades 29
- About the Upgrade Process 30
- About the Override File 33
- About the Database Server Configuration Utility 34
- About the Siebel Upgrade Wizard 38
- About the Repository Merge 38
- About the Siebel Database Server 41

Chapter 4: How to Perform the Upgrade

- Road Map for Performing the Upgrade 43
 - Road Map for Upgrading from Siebel 6.x 44
 - Road Map for Upgrading from Siebel 7.x 44
 - Road Map for Upgrading Without a Development Environment 44
- Process of Planning the Upgrade 45
- Process of Upgrading a 6.x Development Environment from the Midtier 45

Process of Upgrading a 6.x Development Environment from the Mainframe	53
Process of Upgrading a 7.x Development Environment from the Midtier	62
Process of Upgrading a 7.x Development Environment from the Mainframe	69
Process of Upgrading a Production Environment	76

Chapter 5: Database and UI Upgrade Planning

Best Practices for Doing Your Upgrade	83
Planning Your Upgrade	85
About Moving Tablespaces	86
Preparing a Storage Control File	86
New Database Schema Structure in Release 7.7	86
Testing Before a Production Upgrade	87
Considering Code Page Support	88
Executions of Jobs Using Siebel-Scheduled Mode or Vendor-Scheduled Mode	88
Capacity Planning	88
Optimizing Performance Using a Job Stream Optimization (JSO) Utility	88
Obtaining Required Software and Hardware	89
Obtaining Required IBM Utilities	89
Obtaining Required Security Privileges	89
Planning Backup and Recovery Stages	90
Creating a Schedule for the Upgrade	91
About Database Growth Estimates	91
About Upgrading Your RDBMS	92
About Multilingual Deployments	92
Upgrading to RC2 or AES Encryption	93
New Upgrade Features	93
About User Interface Changes	93
About Database Sort Order	94
About Access Control Upgrade	95
About Party Model Upgrade	96
About a New Column Added to S_SRC_PAYMENT	98
About Database Clustered Indexes	99
About HTML Attachment Migration to Base Tables	99

Chapter 6: Basic Database Preparations

Preparing Tables and Views for Upgrade	101
--	-----

Preparing Custom Indexes for Upgrade	102
Exporting Interface Table Data	102
Preserving Dock Objects and Visibility Rules	103
Verifying Database Server Configuration	103
Securing AIX Memory Allocation Segment Space	104
Creating Storage Groups	104
Preparing the Storage Layout of the Schema	104
Modifying the Storage Control File	106
Preparing to Extract the Storage Control File	107
Extracting the Storage Control File	110
Validating the Storage Control File	112
Reviewing the Storage Control File	113
Reviewing Tablespace Size	115
Backing Up the Database	118
Adding a Siebel User	118

Chapter 7: Application Upgrade Planning

Upgrade Planning for Address Migration	119
Upgrade Planning for Siebel Employee Relationship Management (ERM)	120
Upgrade Planning for Siebel Marketing	121
Upgrade Planning for Siebel Workflow Designer	126
Upgrade Planning for Handheld Devices	127
Upgrade Planning for Resonate Central Dispatch	128
Upgrade Planning for String Translation	128

Chapter 8: Preparing Application Data for Upgrade

Preparing Workflow Processes for Upgrade	131
Preparing Mobile and Dedicated User Data for the Database Upgrade	132
Setting Up Campaign Status Values for Siebel Marketing Upgrade	133
Preparing Address Data for Upgrade	135
Preparing Products for Upgrade	136
Preserving Marketing Segment Descriptions	136
Resolving Duplicate Row IDs	137

Preparing Siebel eChannel Data for Upgrade	137
Preparing Forecasting Data for Upgrade	138
Preparing Products and Quotes for Upgrade	138
Verifying Household Data Integrity	139
Preparing Financial Services Application Tables for Upgrade	141
Preparing Siebel Configurator Data for Upgrade	141

Chapter 9: Preparing a Development Environment for Upgrade

Prerequisites for a Development Upgrade	143
About Moving Tables	144
Checking In Development Repository Projects	145
Saving Interface Table Data	145
Upgrading Your DB2 Software	146
Preparing EIM Tables for Upgrade	146
Preparing for Intersection Table Maintenance and Deduplication	148
Determining Which Template File was Used During an Extract or Merge	149
Disabling Customized Triggers	150
Dropping Customized Views	150
Preparing the Development Environment Repositories for Upgrade	150

Chapter 10: Preparing a Production Environment for Upgrade

Prerequisites for a Production Upgrade	153
Importing the Development Repository and Schema Definition	155
Saving Interface Table Data	155
Preparing Production Environment Repositories for Upgrade	156

Chapter 11: Upgrading the Database

Changing the Database Server Configuration Utility Language	159
Preparing to Run the Database Server Configuration Utility	160
Running the Database Server Configuration Utility Under Windows	163
Running the Database Server Configuration Utility Under UNIX	164

Running the Siebel Upgrade Wizard	165
Prerequisites for Restarting the Siebel Upgrade Wizard	166
Deploying Stored Procedures and User-Defined Functions	167
Migrating Address Data from Custom Extension Columns	172
Preparing for a No-Development-Environment Upgrade	175
Installing New License Keys After Upgrade	176

Chapter 12: Upgrading a Database from the Midtier

Executing SQL Files After the First Pause	177
Resuming the Siebel Upgrade Wizard After the First Pause	178
Executing SQL Output Files After the Second Pause	180
Executing SQL Output Files After the Second Pause - Stage 1 of 2	180
Executing SQL Output Files After the Second Pause - Stage 2 of 2	182
Resuming the Siebel Upgrade Wizard After the Second Pause	182
Executing SQL Output Files After the Third Pause	184
Resuming the Siebel Upgrade Wizard After the Third Pause	185
Upgrading the Physical Schema	185
Synchronizing the Schema	186
Transferring Dedup Files	187
Updating Statistics	187
Migrating Custom Business Component Configurations	188

Chapter 13: Upgrading a Database from the Mainframe

Transferring the Customized Repository to z/OS Host	189
Preparing the zSeries Host Environment	190
Prerequisites for Upgrade Procedures on the zSeries	190
Transferring JCL, SQL, and Binary Files to the zSeries Host	191
Defining and Allocating zSeries Setup Datasets	191
Setting Up the Upgrade Environment on the zSeries Host	193
Preparing for Table Creation (PRET) on the Source Database	196
Resuming the Siebel Upgrade Wizard After the First Pause	199
Transferring Control Cards and Schema DDL Files to the zSeries Host	200
Preparing the Schema and JCL Files on the zSeries Host	201
Applying <i>DSNHLQ</i> to FTP DDL, Unload Cards, and Load Cards	201
Unpacking the Unload and Load Control Cards and Executing the DDL	202

Installing Stored Procedures on the zSeries Host	205
Preparing Siebel-Scheduled Jobs on the zSeries Host	206
Optimizing the Source Database Job Stream	210
Loading Log Tables on the Source Database	210
Loading Log Tables on the Target Database	211
Executing the Upgrade on the zSeries Host (Phase 1 of 2)	212
Running Unload Jobs on the Source Database	212
Running Load Jobs on the Target Database	213
Executing Index DDL and Executing Rebuild Indexes	214
Migrating eChannel Data	215
Migrating Household Data	216
Migrating Preschm Data	217
Migrating Address Data	218
Migrating Preschm Data for FINS	218
Migrating Product Configurator Data	219
Migrating Product Configurator Data for FINS	220
Migrating ISS Data	221
Resuming the Siebel Upgrade Wizard on the Midtier After the Second Pause	222
Transferring the Index Schema to the zSeries Host	224
Preparing Siebel-Scheduled JCL	225
Optimizing the Target Database Job Stream	227
Executing the Upgrade on the zSeries Host (Phase 2 of 2)	228
Loading Target Siebel Log Tables	228
Dropping Old Indexes and Creating New Indexes	229
Rebuilding the Gen Primary 7.7 Indexes	231
Migrating the Gen Primary 7.7 SQL	232
Creating New EIM Indexes	233
Generating RUNSTATS	234
Performing Intersection Table Maintenance	234

Chapter 14: Reviewing the Database Upgrade Log Files

About the Database Upgrade Logs	239
Viewing the Siebel Job Log Status	240
Reviewing Upgrade Log Files for Errors	241
Manually Archiving Upgrade Log Files	243
Reviewing Log Files for Errors in Jobs Run from the Midtier	243

Running SQL in Siebel Logs 243

Chapter 15: Performing the Siebel Tools Repository Merge

About Inheriting Upgrade Behavior 245

About the Postmerge Utilities 248

About the Incorporate Custom Layout (ICL) Upgrade Option 248

Configuring Objects to Inherit Upgrade Behavior 253

Renaming the Tools Repository 254

Migrating Repository Objects to the Standard UI 255

Running the Repository Preparation Wizard on Release 6.x Repository 259

Performing a Repository Merge 262

Determining if a Repository Merge was Successful 271

Reviewing Attribute Conflicts in the Repository Merge 273

Reviewing Deleted Objects in the Repository Merge 275

Reviewing Obsolete Objects in the Repository Merge 275

Reviewing Customized Business Components 276

Generating EIM Temporary Columns After a Repository Merge 277

Regenerating the Repository Definition Files 278

Deleting Unneeded Repository Files 280

Safeguarding the New Custom Repository Export File 280

Moving the Repository Files 281

Creating a New SRF File 282

Chapter 16: Migrating Siebel 6.x Customizations

About Migrating to the Siebel Web Client 285

About Migrating 6.x Scripts 291

About Migrating 6.x Client-Side Interfaces 292

Identifying What Will Be Migrated 293

Exposing Hidden Properties of Applets and Views 294

Running the Siebel Web Client Migration Wizard 295

Reviewing Applet and View Migration 298

Running the Web Layout Wizard 299

Migrating Scripts Attached to Controls	300
Migrating Scripts Attached to Applets	302
Migrating Business Component, Business Service, and Application Scripts	305
Migrating Outbound COM Interfaces	306
Migrating Inbound COM Interfaces	307
Regenerating the Postmerge Utilities Log	309
Reviewing Customized Business Components	310

Chapter 17: Running the Postmerge Utilities

Running the Postmerge Utilities	313
Troubleshooting Postmerge UI Problems	314
Verifying Business Address Applet Configuration	315
Reviewing Grid-Based Applets After the Merge	316
Reviewing UI Navigation After the Merge	317
Reviewing Multi-Value Group (MVG) Shuttle Applets After the Merge	318
Revising Rich Text Controls After the Merge	320
Reviewing New Aggregate Categories After the Merge	321
Revising Visibility Filters to Display Correctly After the Merge	322
Assigning a Category and Type to Chart Views After the Merge	323
Assigning a Category and Type to Explorer Views After the Merge	324
Setting Up Navigation to Inaccessible Detail Views After the Merge	324
Eliminating Obsolete UI Fields After the Merge	325
Reviewing UI Objects Affected by ICL After the Merge	326
Reviewing Required Fields in the UI After Merge	326
Setting Label Alignment for Text Fields	327
Assigning an Item Identifier to Web Template Items	327

Chapter 18: Postupgrade Tasks for a Development Environment

Reapplying Custom Extensions and FK Columns on Obsolete Tables	329
Resolving Business Component and Join Conflicts	331
Deleting Duplicate EIM Mappings	332

Exporting the New Custom Repository Export File	333
Regenerating the Database Template File	334
Extracting Developers or Siebel Tools Clients	334
Resetting Database Server Configuration Parameters	335
Upgrading to the Symbolic String Model	335

Chapter 19: Postupgrade Tasks for a Production Environment

Upgrading Siebel Mobile and Dedicated Web Clients	337
Upgrading Regional Servers	337

Chapter 20: Postupgrade Tasks for Applications

Generating Reporting Relationships After Upgrade	340
Setting Up Your Environment to Support Global Time Zone	342
Updating Enterprise Application Integration (EAI) After Upgrade	343
Setting Visibility Modes for Access Control	344
Removing Call Center Duplicate Logins After Upgrade	347
Upgrading ERM Customized Microsite and Group News Pages	347
Migrating Course Duration Information for Siebel Training	349
Upgrading the Launch Field in Siebel Training LOV	349
Upgrading the Test Status in Siebel Training After Upgrade	350
Verifying Class and Session Times in Siebel Training	350
Upgrading ERM Approval Business Process Workflows	352
Migrating Data to the Bankruptcy Status Field	353
Upgrading File System Attachments for Siebel Financial Services Call Reports	353
Upgrading File System for Household Notes and Attachments	354
Upgrading Marketing Responsibilities	356
Reviewing Marketing Campaign Data	356
Reviewing Renamed Fields in Siebel Marketing	357
Displaying Marketing Regions	357
Revising Marketing Program Flowchart Icons	358
Setting Default Campaign Execution Options	358

Upgrading Activity Plans for Programs and Campaigns	358
Upgrading the Newsletter Offer Type	359
Configuring Universal Inbox	359
Reviewing Data Warehouse Schema Changes	360
Upgrading Siebel Purchase Orders	360
Configuring Asset-Based Ordering	361
Reviewing Address Records After Upgrade	362
Configuring Products and Quotes	363
Upgrading Attribute Pricing	366
Verifying the Upgrade to Aggregate Discounts in Pricer	367
Migrating Data to the Bankruptcy Status Field	368
Upgrading Seeded Workflows	368
Upgrading Inbound Workflows	368
Upgrading Siebel Product Configurator	369

Chapter 21: Postupgrade Tasks for Database and File System

Checking for Inactivated EIM Table Columns	371
Validating Dock Objects and Rule Definitions	371
Updating the File System Directory	372
Upgrading to the Symbolic String Model	373
Updating File System Attachments	374

Chapter 22: Tuning the Upgrade Scripts

About Tuning the Production Upgrade Scripts	377
Deactivating SQL that Affects No Data	377
Transferring UNIX Files for Use by Upgrade Tuner	378
Discarding Changes Made by the Upgrade Tuner	380

Appendix A: Schema Changes

Important Schema Changes at Release 7.7	381
---	-----

Appendix B: Siebel Marketing Upgrade Reference

Obsolete Siebel Marketing Data	387
Obsolete Business Objects in Siebel Marketing	393
Obsolete or Replaced Views in Siebel Marketing	394

Appendix C: Upgrade Planning Worksheet

Master Upgrade Worksheet	403
Team Lead Summary	403
DB2 Connect Information	403
Siebel Development Environment Information	404
Siebel Production Environment Information	405

Appendix D: Columns Denormalized During Upgrade to Release 7.7

Denormalized Columns for 6.2.1 Siebel Financial Services Applications	407
Denormalized Columns for 6.2.1 Siebel eBusiness Applications	416
Denormalized Columns for 6.3 Siebel Industry Solutions	423
Denormalized Columns for 7.0.4 Siebel eBusiness Applications	433
Denormalized Columns for 7.0.4 Siebel Financial Services Applications	435
Denormalized Columns for 7.0.4 Siebel Industry Solutions	439
Denormalized Columns for 7.5.2 Siebel Industry Applications	443

Index

1

What's New in This Release

What's New in Upgrade Guide for DB2 UDB for z/OS and OS/390, Version 7.7 Rev. C

Table 1 lists changes in this version of the documentation to support Release 7.7 of the software.

Table 1. New Features in Upgrade Guide for DB2 UDB for z/OS and OS/390, Version 7.7 Rev. C

Topic	Description
Combined book <i>Upgrade Guide for DB2 UDB for z/OS and OS/390</i>	This guide is now a stand-alone guide containing all of the procedures necessary to perform z/OS and OS/390 upgrades. You no longer need to use the <i>Upgrade Guide</i> as a companion to this guide.
Chapter 4, "How to Perform the Upgrade"	New chapter. This chapter contains checklists for performing z/OS and OS/390 upgrades.
Chapter 13, "Upgrading a Database from the Mainframe"	New chapter. There are now two ways to upgrade a development environment database: from the midtier or from the mainframe. Review this chapter for information on upgrading a development environment database from the mainframe.
"Upgrade Planning for Address Migration" on page 119	New topic. The way addresses are stored has been changed at Release 7.7. Review this topic to determine how to handle table customizations.
"Preparing Products for Upgrade" on page 136	New topic. To migrate products in workspaces, you must release them. Class products must have the orderable flag unchecked.
"Preparing Address Data for Upgrade" on page 135	New topic. You must check for duplicate row-IDs within and between S_ADDR_PER and S_ADDR_ORG before performing the upprep of the Siebel Database.
"Migrating Repository Objects to the Standard UI" on page 255	New topic. If you selected the Incorporate Custom Layout (ICL) option for the previous Release 7.x upgrade, you must return your user interface to the standard look and feel before performing the repository merge.
"About the Incorporate Custom Layout (ICL) Upgrade Option" on page 248	Revised topic. Added guidelines for deciding when to use Incorporate Custom Layout (ICL).

Table 1. New Features in Upgrade Guide for DB2 UDB for z/OS and OS/390, Version 7.7 Rev. C

Topic	Description
“Migrating Address Data from Custom Extension Columns” on page 172	Revised topic. The changes required to the ddl.ctf and preschem_sia.sql files to migrate address data in custom extension columns has been heavily updated. Review this topic carefully before migrating address data from custom extension columns.
“Running the Postmerge Utilities” on page 313	New topic. You must start the postmerge utilities manually. You can run them multiple times.
“Setting Label Alignment for Text Fields” on page 327	New topic. If you select the ICL option for the upgrade and choose Label on Top, you must edit main.css to obtain correct label alignment.
“Verifying Class and Session Times in Siebel Training” on page 350	New topic. You must verify class and session start times in Siebel Training after upgrade.
“Reviewing Address Records After Upgrade” on page 362	New topic. You must review address records after upgrade to eliminate duplicate and obsolete records.
“Upgrading Attribute Pricing” on page 366	New topic. You must run a business service method to upgrade attribute pricing.
“Verifying the Upgrade to Aggregate Discounts in Pricer” on page 367	New topic. You must verify bundle factors have been correctly updated to the new aggregate discounts feature.
“Important Schema Changes at Release 7.7” on page 381	New topic. Lists important schema changes in Release 7.7.
“Upgrading ERM Customized Microsite and Group News Pages” on page 347	Revised topic. Corrected error in procedure.
Chapter 17, “Running the Postmerge Utilities”	All topics rewritten to improve clarity. Tools procedures added where necessary. Three new topics added covering Issue 7, 8, and 9 sections of postmerge utilities log.
“Migrating Data to the Bankruptcy Status Field” on page 368	New topic. BK_STATUS_CD is added to S_BANKRUPTCY. This column stores bankruptcy status for use by Siebel Financial Services applications. If you have implemented a bankruptcy status field, you must migrate the data to BK_STATUS_CD.

What's New in Upgrade Guide for DB2 UDB for z/OS and OS/390, Version 7.7, Rev. B

Table 2 lists changes described in this version of the documentation to support Release 7.7 of the software.

Table 2. New Product Features in Upgrade Guide for DB2 UDB for z/OS and OS/390, Version 7.7, Rev. B

Topic	Description
"New Database Schema Structure in Release 7.7" on page 86	<ul style="list-style-type: none"> Enhanced description of the schema structure that is a new feature of Release 7.7.
"Deploying Stored Procedures and User-Defined Functions" on page 167	<ul style="list-style-type: none"> Corrected sequence in which SQL files are executed. Split execution of output files into two stages.

What's New in Upgrade Guide for DB2 UDB for z/OS and OS/390, Version 7.7, Rev. A

Table 3 lists changes described in this version of the documentation to support Release 7.7 of the software.

Table 3. New Product Features in Upgrade Guide for DB2 UDB for z/OS and OS/390, Version 7.7, Rev. A

Topic	Description
"Deploying Stored Procedures and User-Defined Functions" on page 167	<ul style="list-style-type: none"> Added prerequisite: schema must be upgraded prior to performing procedure. Added missing steps to two jobs in the procedure.
Incorporate Custom Layout (ICL)	Incorporate Custom Layout is not supported at Siebel 7.7. Removed all references to this feature in the book.

What's New in Upgrade Guide for DB2 UDB for z/OS and OS/390, Version 7.7

Table 4 lists changes described in this version of the documentation to support Release 7.7 of the software.

Table 4. New Product Features in Upgrade Guide for DB2 UDB for z/OS and OS/390, Version 7.7

Topic	Description
<p>New book</p> <p><i>Upgrade Guide for DB2 UDB for z/OS and OS/390</i></p>	<p>Combined the <i>Upgrade Guide for Microsoft Windows, DB2 UDB for OS/390 and z/OS</i> and <i>Upgrade Guide for UNIX, DB2 UDB for OS/390 and z/OS</i> into one book.</p> <p>This guide is no longer a stand-alone guide. This guide is a companion to the <i>Upgrade Guide</i>.</p> <ul style="list-style-type: none"> ■ Refer to <i>Upgrade Guide</i> for tasks to Migrate Client Customizations to the Siebel 7 Web Client (for upgrades from Release 6.x). ■ Refer to <i>Upgrade Guide</i> for tasks to reconfigure the user interface for Release 7.7.
<p>Enhanced development upgrade infrastructure</p> <p>See Chapter 12, "Upgrading a Database from the Midtier"</p>	<p>The Release 7.7 development upgrade for Siebel eBusiness applications on DB2 features infrastructure enhancements for a midtier upgrade process using the Siebel Database Server Configuration Utility.</p>
<p>Enhanced production upgrade infrastructure</p> <p>See Chapter 13, "Upgrading a Database from the Mainframe"</p>	<p>The Release 7.7 production upgrade for Siebel eBusiness applications on DB2 features a streamlined process run from the midtier against the mainframe, and enhanced panel-driven upgrade process on the mainframe.</p>

2

About Upgrade Topics

This chapter contains the following topics:

- “About This Guide” on page 19
- “How the Upgrade Guide Is Organized” on page 20
- “About Topic Applicability” on page 20
- “About Terms Used in the Upgrade Guide” on page 21
- “About File Paths and Commands in the Upgrade Guide” on page 22
- “Important Upgrade Planning Resources” on page 23

About This Guide

Upgrades: All upgrades.

This guide describes the process of upgrading to Release 7.7 from several prior releases of Siebel eBusiness applications on DB2 UDB for z/OS. While the general upgrade procedure is the same for all releases and platforms, the steps may vary slightly depending on the release from which you are upgrading.

For a list of currently supported systems and supported upgrade paths, see *System Requirements and Supported Platforms* on Siebel SupportWeb at <http://business.siebel.com/supportweb/>.

A successful upgrade requires:

- Detailed understanding of customizations to support the previous implementation of Siebel eBusiness applications.
- Familiarity with the basic conventions of the operating system under which your Siebel Servers run.
- Familiarity with your specific Siebel eBusiness applications deployment.
- Expertise in network connectivity, disk and file sharing, and software for DB2 UDB for z/OS.
- UNIX midtier: User accounts with root access on UNIX to install new software.
- Windows midtier: User accounts with administration privileges on Microsoft Windows.
- Expertise in database installation, tuning, and administration in your relational database management system (RDBMS).

This book is useful primarily to people whose titles or job descriptions match one of the following:

System Administrators	Persons responsible for administering the whole system, including installing, maintaining, and upgrading Siebel products.
------------------------------	---

Database Administrators Persons who administer the database system, including data loading; system monitoring, backup, and recovery; space allocation and sizing; and user account management.

How the Upgrade Guide Is Organized

Upgrades: All upgrades.

Use the road map and process topics in [Chapter 4, “How to Perform the Upgrade,”](#) to guide you through the upgrade process. These topics provide a checklist of the steps required to complete an upgrade, in the order you must perform them. Each step includes a link to a topic that explains how to complete the step. The remaining chapters of the *Upgrade Guide for DB2 UDB for z/OS* are organized according to the major phases of the upgrade.

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

The order of topics in a chapter is as follows:

- A link listing all the topics in the chapter.
- **Road map topic.** A road map is a numbered list of processes. For example, *Road Map for Performing an Upgrade from Siebel 7.x* is a list of the processes required to upgrade your Siebel environments.
- **Process topics.** A process topic has a numbered list of tasks. For example, *Process of Upgrading a 7.x Development Environment* lists the tasks required to upgrade a single Siebel environment. These topics begin with *Process of...*
- **Concept topics.** Concept topics explain key concepts required to perform groups of tasks. To get an overview of how each part of the upgrade works, read the concept topics. These topics usually begin with *About...* Concept topics do *not* begin with a gerund (such as, *Planning your Upgrade* or *Moving the Repository Files*).
- **Task topics.** Task topics explain how to do each step in an upgrade process, and typically include a procedure. Task topics begin with a gerund, for example *Updating File System Attachments*.

About Topic Applicability

Upgrades: All upgrades.

Environments: All environments.

The applicability of topics in the book is listed at the beginning of each topic. [Table 5 on page 21](#) lists the applicability categories and their meaning.

For each topic, only the relevant categories are listed.

Table 5. Topic Applicability Categories

Applicability Category	Meaning
Upgrades	<p>Lists the upgrades to which the topic applies.</p> <p>For example, Upgrades: Releases 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 Upgrade Environments” on page 28.</p>
Platforms	<p>Lists the operating system platform to which the topic applies.</p> <p>For example, Platforms: Windows only. means the topic applies only to Windows platforms.</p>

About Terms Used in the Upgrade Guide

Upgrades: All upgrades.

Environments: All environments.

This guide follows several naming conventions:

- **DB2** refers to IBM DB2 UDB for z/OS.
 - NOTE:** This version of the guide replaces the *IBM DB2 UDB for z/OS and OS/390* naming convention with *IBM DB2 UDB for z/OS*.
- Current release means the currently shipping release of the Siebel eBusiness applications.
- Release 7.x refers collectively to all versions of Siebel 7 prior to Release 7.7, for example Release 7.0.4 or 7.5.3. See the Upgrade section in *System Requirements and Supported Platforms* on Siebel SupportWeb at <https://ebusiness.siebel.com/supportweb/> for specific information on which release numbers are meant by Release 7.x.
- Release 6.x refers collectively to all versions of Siebel 2000 that are supported in upgrade to Release 7.7 on DB2. Refer to the Upgrade section in *System Requirements and Supported Platforms* on Siebel SupportWeb for specific information on which release numbers are meant by Release 6.x.

- The term Windows refers to all Microsoft Windows operating systems listed as supported for this release in *System Requirements and Supported Platforms* on Siebel SupportWeb.
- The term UNIX refers to all forms of the UNIX operating system supported for this release in *System Requirements and Supported Platforms* on Siebel SupportWeb.

About File Paths and Commands in the Upgrade Guide

Upgrades: All upgrades.

Environments: All environments.

Environment variables and path placeholders for both Windows and UNIX paths are used throughout the *Upgrade Guide for DB2 UDB for z/OS*. You must enter UNIX commands in either a C-shell or Korn shell. Enter Windows commands in a Windows Command Prompt window.

Windows Paths

The following path conventions specify file system locations in *Upgrade Guide for DB2 UDB for z/OS* topics:

- *SIEBEL_ROOT* is the absolute path of the Siebel Server installation directory. When you install a Siebel Server, the installation script queries for the installation path. The script then installs the Siebel Server in a subdirectory of this path called *siebsrvr*. For example, if you specified C: \sea7xx as the installation path, then *SIEBEL_ROOT* is C: \sea7xx\si ebsrvr.
- *DBSRVR_ROOT* is the absolute path to the Siebel Database Server files on the Siebel Server. When you install the Siebel Database Server, the installation script queries for the Siebel Server installation directory. The script then installs the Siebel Database Server files at the same level in a subdirectory called *dbsrvr*. For example, if *SIEBEL_ROOT* is C: \sea7xx\si ebsrvr, then *DBSRVR_ROOT* is C: \sea7xx\dbsrvr.

UNIX Paths

The following environment variables and path conventions specify file system locations in *Upgrade Guide for DB2 UDB for z/OS* topics:

- *SIEBEL_ROOT* is an environment variable that defines the absolute path of the Siebel Server installation directory. When you install a Siebel Server, the installation script queries for the installation path. The script then installs the Siebel Server in a subdirectory of this path called *siebsrvr*. For example, if you specified C: /sea7xx as the installation path, then *\$SIEBEL_ROOT* is C: /sea7xx/si ebsrvr.

The definition of *SIEBEL_ROOT* and other environment variables required for doing an upgrade are located in */si ebsrvr/si ebenv.csh*, (*si ebenv.sh* for Korn shells). The Siebel Server installation script creates this shell script. Do not edit or delete this file.

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

- *DBSRVR_ROOT* is a path convention used in the *Upgrade Guide for DB2 UDB for z/OS*. It is not an environment variable and is not defined in `siebenv.csh` or `siebenv.sh`.

DBSRVR_ROOT is the absolute path to the Siebel Database Server files on the Siebel Server. When you install the Siebel Database Server, the installation script queries for the Siebel Server installation directory. The script then installs the Siebel Database Server files at the same level in a subdirectory called `dbsrvr`. For example, if *\$SIEBEL_ROOT* is `C:/sea7xx/siebsrvr`, then *DBSRVR_ROOT* is `C:/sea7xx/dbsrvr`.

Path Navigation

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

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

Executing Commands

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

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

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

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

Important Upgrade Planning Resources

Upgrades: All upgrades.

Environments: All environments.

Your Siebel implementation team performs a number of actions to install and implement Siebel eBusiness applications that are described in several Siebel publications on the Siebel Bookshelf:

- *Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390*
- *Siebel Deployment Documentation Suite*

- *Deployment Planning Guide*
- *Siebel Installation Guide for Microsoft Windows: Servers, Mobile Web Clients, Tools*
- *Siebel Installation Guide for UNIX: Servers, Mobile Web Clients, Tools*
- *Monitoring and Diagnostics Guide for Siebel eBusiness Applications*
- *Going Live with Siebel eBusiness Applications* for information about how to migrate customizations from the development environment to the production environment
- *Siebel System Administration Guide* for details on how to administer, maintain, and expand your Siebel Servers
- *Security Guide for Siebel eBusiness Applications*
- *Performance Tuning Guide*
- *Configuring Siebel eBusiness Applications* for information about configuring Siebel eBusiness applications in Siebel Tools
- **Product Documentation.** Product documentation is collectively called the Siebel Bookshelf. The Siebel Bookshelf is available on CD-ROM. It is also available on SupportWeb under Product Documentation.
- **Siebel SupportWeb.** This is the Siebel technical support Web site. It provides search engine access to the Siebel Bookshelf, Technical Notes, Siebel Alerts, troubleshooting information and other important information. SupportWeb is located at <https://ebusiness.siebel.com/supportweb/>.
 - *System Requirements and Supported Platforms* on Siebel SupportWeb. This document is the definitive list of system requirements and supported third-party products. It is located on SupportWeb at Product Documentation > System Requirements and Supported Platforms & Miscellaneous Documentation.
 - *Release Notes.* *Release Notes contain* late-breaking information that the *Upgrade Guide* does not yet include. Release Notes regarding upgrade are located on SupportWeb at Product Documentation > Release Notes > Siebel eBusiness, Industry, Midmarket Edition, and CRM applications 7.x and later > Categories > General > Upgrade.
 - *Maintenance Release Guides.* Maintenance Release guides contain important information about updates to applications in maintenance releases. Maintenance Release guides are located on SupportWeb at Product Documentation > Maintenance Release Guides.
 - *Documentation Updates.* Typically, the Bookshelf is updated monthly. During the month, PDF updates are posted to SupportWeb > Product Documentation > Documentation Updates.
 - *Technical Notes.* Technical notes provide important information on specific upgrade issues. Technical Notes related to upgrade are located on SupportWeb at Technical Notes > Upgrade. Technical Notes of particular importance are as follows:
 - Technical Note 521 has information on critical upgrade issues.
 - Technical Note 447 has information on upgrading multilingual deployments to Siebel 7.5.x and later.

- *Siebel Alerts.* Alerts provide time-critical information on key product behaviors and issues. Siebel Alerts about upgrade issues are located on SupportWeb at Siebel Alerts > Product Areas > Upgrade. Siebel Alert 1002 lists Technical Notes and other resources that pertain to upgrades.
- *Troubleshooting Steps.* Troubleshooting Steps contain information about how to troubleshoot common error messages and unwanted behavior in Siebel applications. Troubleshooting Steps are for upgrade list error messages found in upgrade logs and describe how to resolve them. Troubleshooting Steps are located on SupportWeb at Troubleshooting Steps > Product Areas > Upgrade.
- *Siebel Weekly Content Notification Service.* This service notifies you by email weekly of important content changes on SupportWeb. This includes new product documentation, technical notes, alerts, and troubleshooting steps. To subscribe, see the Siebel Weekly Content Notification banner on the SupportWeb search page.
- **Technical Account Manager.** If you need assistance planning your upgrade or encounter problems during the upgrade, your Technical Account Manager can advise you on how best to use available Siebel resources.
- **Siebel Expert Services.** Siebel Expert Services offers detailed implementation planning and technical consulting services. They also provide rapid response and resolution for critical technical issues affecting Siebel deployments. Contact Siebel Expert Services for important information in the following areas:
 - Migrating from Siebel Industry Solutions or Siebel Financial Services applications to Siebel eBusiness applications during upgrade
 - Migrating to Unicode codepage support during upgrade
 - Changing operating system type during upgrade

3

How the Upgrade Works

This chapter contains the following topics:

- “About Supported Upgrade Paths” on page 27
- “About Upgrade Environments” on page 28
- “About Midtier-Centric and Mainframe-Centric Development Environment Upgrades” on page 29
- “About the Upgrade Process” on page 30
- “About the Override File” on page 33
- “About the Database Server Configuration Utility” on page 34
- “About the Siebel Upgrade Wizard” on page 38
- “About the Repository Merge” on page 38
- “About the Siebel Database Server” on page 41

About Supported Upgrade Paths

Upgrades: All upgrades.

Environments: All environments.

Supported upgrade paths are described in *System Requirements and Supported Platforms* on Siebel SupportWeb. The *Upgrade Guide for DB2 UDB for z/OS* describes how to upgrade the following installations:

- A Siebel 6.x FINS installation to Release 7.7.
- A Siebel 7.x installation to Release 7.7.

The *Upgrade Guide for DB2 UDB for z/OS* does not cover the following specific upgrade paths or infrastructure changes. Contact Siebel Technical Support or Siebel Expert Services for help with these tasks:

- Changing operating system type during an upgrade, for example changing from Windows to UNIX.
- Migrating to Unicode.
- Migrating from Siebel Industry Solutions applications or Siebel Financial Services applications to Siebel eBusiness applications.

If your enterprise uses SAP and you have implemented the Siebel Systems 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 Upgrade Environments

Upgrades: All upgrades.

Environments: All environments.

The *Upgrade Guide for DB2 UDB for z/OS* describes how to upgrade two environments:

- Development environment
- Production environment

Development Environment

A development environment contains the following elements:

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

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

Siebel Systems provides these tools to help you transition from development to production:

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

Mapping Your Environments

You may have more or fewer environments than those described above. [Table 6](#) gives recommendations for mapping your environments to the ones used in the *Upgrade Guide for DB2 UDB for z/OS*.

Table 6. 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 is used for live business transactions by both local and remote users.</p>	Production environment upgrade.

About Midtier-Centric and Mainframe-Centric Development Environment Upgrades

Upgrades: All upgrades.

Environments: Development environment only.

Historically, all development environment upgrades have been done from the midtier (midtier-centric) and applied to the z/OS host using Windows- or UNIX-based client-side tools. In Release 7.7, the development environment upgrade has been enhanced to allow for all DDL (CREATE, DROP, DELETE, ALTER, and GRANT) and DML (INSERT, UPDATE, DELETE, and SELECT) to be executed on the mainframe host. Based on your business requirements, you can choose to perform either a midtier-centric or mainframe-centric development environment upgrade. Table 7 lists some of the advantages and disadvantages of each type of upgrade.

NOTE: By default, the Siebel software contains the midtier-centric development environment upgrade files. If you want to perform a mainframe-centric development environment upgrade, you need to copy and replace these files in order to generate the appropriate upgrade output files.

Table 7. Midtier-centric and Mainframe-centric Environment Upgrades

Upgrade	Advantage	Disadvantage
Midtier-centric development environment.	This type of upgrade follows previous Siebel upgrade flows and allows you to work with existing expertise.	This type of upgrade does not allow you to take advantage of any enhancements that were built into the mainframe-centric development environment upgrade.
Mainframe-centric development environment.	You can build a level of experience with this type of upgrade that can be leveraged during your production environment upgrade.	This type of upgrade requires someone with a TSO account, proper accesses and authorities. Also, since this type of upgrade is new for Siebel 7.7, you will need to spend time learning how to perform it.

About the 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. This merges your customizations with the new release. A merged repository file and database schema file are created and become inputs to the development 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 development environment and production environment.

Flow of the Upgrade Within an Environment

The basic flow of the upgrade process is shown in Figure 1. This flow applies to development and production environment upgrades. However, when upgrading a production environment, there are several additional deployment steps.

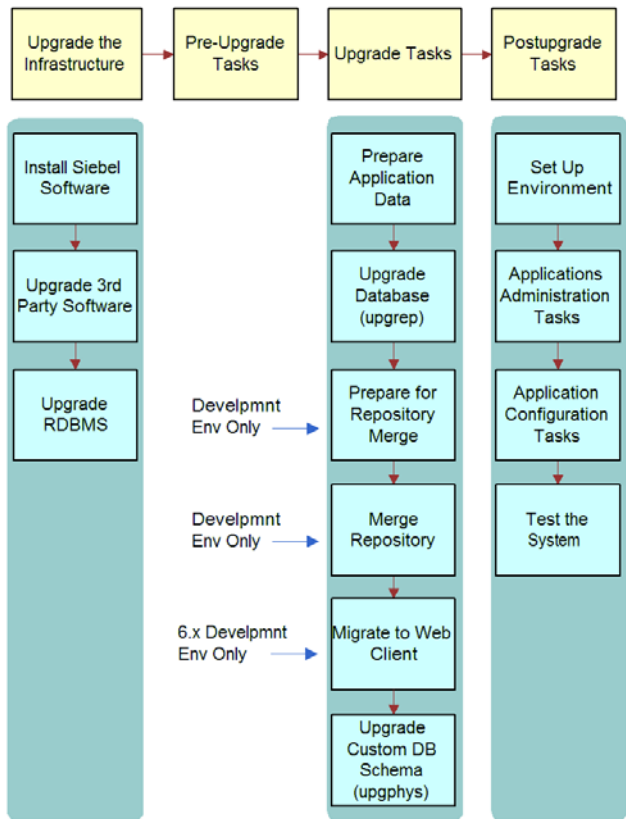


Figure 1. Flow of the Upgrade Process

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.

Perform Preupgrade Tasks

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

Perform Upgrade Tasks (Development Environment)

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

- **Prepare application data.** These tasks prepare test data for migration.
- **Upgrade database (upgrep).** You run the Database Server Configuration utility in upgrep mode. This utility performs a basic upgrade of the Siebel Database schema and loads repositories to prepare for the repository merge.
- **Merge repository.** You use Siebel Tools to merge your existing repository with the repository in the new release. Postmerge utilities upgrade form applets and verify that applets and views are configured correctly.
- **Upgrade database (upgphys).** You run the Database Server Configuration utility in upgphys mode. It further upgrades the Siebel Database with changes resulting from the repository merge and completes the database upgrade.

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

Perform Upgrade Tasks (Production Environments)

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

- **Prepare Application Data.** These tasks are about preparing application data for migration.
- **Upgrade database (upgrep + upgphys).** You run the Database Server Configuration utility in upgrep + upgphys mode. It performs a basic upgrade of the Siebel Database schema by importing the repository and schema definition files from the development environment upgrade and using these files to upgrade the Siebel Database and perform miscellaneous administrative tasks.

Perform Postupgrade Tasks

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

- **Set Up the Environment.** These tasks set up the postupgrade environment, which includes extracting the developers' databases and running database statistics.
- **Application Administration.** These tasks set up applications and include such things as setting up user access and visibility of views and screens.
- **Application Configuration.** These tasks prepare applications for testing, including data migration for specific applications.

- **Test the System.** These tasks test the system. For development environment upgrades, you perform basic unit tests to verify application function followed by a full suite of regression and stress tests to verify the system is ready for production.

Related Topics

[“About the Database Server Configuration Utility” on page 34](#)

[“About the Siebel Upgrade Wizard” on page 38](#)

[“About the Repository Merge” on page 38](#)

[“About the Siebel Database Server” on page 41](#)

About the Override File

Upgrades: All upgrades.

Environments: Development environment only.

There are three options when upgrading to the current release:

- Preserve the existing model of multiple tables per tablespace 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 source 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 of the source tables but specify specific source tables to use the *1:1:1* model.

To use this option, use the `override.inp` file. By default, this file has two table entries:

- S_DOCK_TXN_LOG
- S_SERVICE_SCRIPT

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

- Move all source 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 with the *1:1:1* paradigm. Therefore, each table is created in its own database or tablespace and the data is moved from the source to the target table.

The use of the preconfigured storage control file allows you to have the *1:1:1* model for all tables except for 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 tablespaces are not cleaned up unless they are empty. Some of the obsolete and customer extended tables are still in those tablespaces so the upgrade process cannot just do a general cleanup.

About the Database Server Configuration Utility

Upgrades: All upgrades.

Environments: All environments.

The Database Server Configuration utility provides a user interface for performing the Siebel Database upgrade. To complete an upgrade, you must run the utility several times, each time in a different mode, as shown in [Figure 2](#).

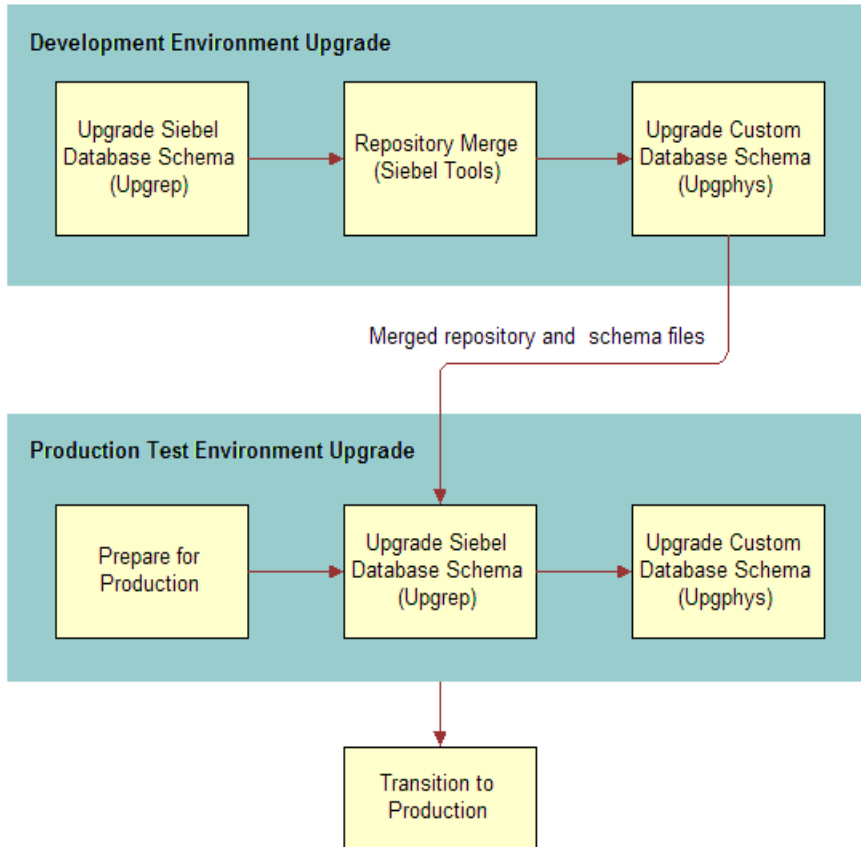


Figure 2. Database Server Configuration Utility Modes

Development Environment Upgrades

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

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

- Imports the 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
- **upgphys.** This mode makes the following changes:
 - Synchronizes the Siebel Database schema to the logical schema definition in the merged repository
 - Deduplicates intersection tables
 - Exports repository object definitions to a file, custrep.dat
 - Exports the logical schema definition to a file, schema.ddlThese two files are used as input to the development environment and production environment upgrades.
- Renames the New Customer Repository to Siebel Repository
- Updates the schema version in S_APP_VER

Production Upgrades

For a production upgrade, you run the utility in the following mode:

- **upgrep + upgphys.** This mode makes the following changes:
 - Drops interface tables and database triggers
 - Populates columns that must change from NULL to NOT NULL
 - Uses the custrep.dat and schema.ddl files from the development environment upgrade to create new tables and merge existing tables
 - Prepares for index creation. Verifies that there are no unique key violations.
 - Creates indexes
 - Imports seed data
 - Updates primary children foreign key references
 - Performs miscellaneous file actions
 - Makes several administrative changes to table data, including updating the schema version in S_APP_VER. It does not make schema changes.

How SQL Files Are Handled

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

- Collects configuration information

- Forwards the information to an SQL generator that creates or populates SQL files with the required commands. The SQL generator extracts these commands from an intermediate XML file containing all the SQL commands required for an upgrade.
- Creates an upgrade configuration file (UCF)
- Prompts you to start the Siebel Upgrade Wizard. The wizard uses the UCF file to execute the SQL files and other input files against the Siebel Database.

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 run the Siebel Upgrade Wizard. To do this, answer No when prompted to run the Siebel Upgrade Wizard. Then, edit the SQL files and manually launch the Siebel Upgrade Wizard.

How To Identify Upgrade Configuration Files

Upgrade configuration files are stored in the following location:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

Upgrade configuration files use the following naming convention:

`master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`

where:

- `UPGRADEOPTION` = the upgrade process you are performing
 - Siebel Database schema upgrade = `upgrep`
 - Custom database schema upgrade = `upgphys`
- `ENVIRONMENT` = the environment that you are upgrading
 - Development environment upgrades = `dev`
 - Production environment upgrades = `prod`
- `VERSION` = the version from which you are upgrading

For example, if you are upgrading from Siebel 7.0.4, the UCF file generated from the development environment `upgrep` is as follows:

`master_upgrep_dev_704.ucf`

Related Topic

[“About the Siebel Upgrade Wizard” on page 38](#)

About Parameter Validation

The Database Server Configuration Wizard validates certain parameters, and does not proceed if you enter an invalid parameter. The configuration wizard validates the following information:

- That you have renamed the *Siebel Repository* to *Prior Customer Repository*

- That tablespaces exist in your database
- That username and password are valid
- That tableowner and tableowner password are valid
- That the language pack is installed
- That the directories chosen exist

About the Siebel Upgrade Wizard

Upgrades: All upgrades.

Environments: All environments.

The Siebel Upgrade Wizard makes changes to the Siebel Database using the SQL files and the upgrade configuration file (UCF) generated by the Database Server Configuration utility. After the Siebel Upgrade Wizard starts, it executes this process:

- Reads the UCF
- Calls a driver
- Passes the information in the UCF to the driver, which then passes UCF file information to:
 - ddlimp, which executes ddl-type SQL commands
 - dataimp, which executes data-related SQL commands

The Siebel Upgrade Wizard also does the following:

- Upgrades Siebel seed data
- Loads the following repositories for development environment upgrades:
 - Prior Standard Repository
 - New Standard Repository
 - New Customer Repository

If ddlimp 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.

Related Topics

[“About the Database Server Configuration Utility” on page 34](#)

[“About the Repository Merge” on page 38](#)

About the Repository Merge

Upgrades: All upgrades.

Environments: Development environment only.

The Siebel Repository consists of records stored in a group of tables in the Siebel Database. There are two types of records:

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

Think of the Siebel Repository as these two types of records, rather than the tables that store the records. Repository records include a repository ID that lists the ID of the repository to which each record belongs. The repository ID forms part of the user index for these records and allows several repositories to be stored in the same set of tables.

List-Applets in Siebel Tools display the information in the repository records.

Upgrading the Siebel Database adds new records to the tables that store the repository records. These new records make up three repositories, as shown in [Table 8](#).

Table 8. Development Database Repositories

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

How the Merge Works When You Are Upgrading from Siebel 7.x

The repository merge does the following things:

- Identifies customizations you have made in the Prior Customer Repository and uses decision criteria to merge them into the New Customer Repository
- Identifies logical schema changes you have made in the Prior Customer Repository and tries to include them in the logical schema in the New Customer Repository
- Upgrades both the base repository and all locales
- Writes a log describing actions, conflicts, and errors
- After the merge is complete, launches postmerge utilities that perform further upgrades to layout objects such as Web templates and controls

The merge process makes numerous record comparisons and inserts. As a result, the merge is resource-intensive and typically takes five to seven hours to complete. The time required varies depending on the complexity of customizations and amount of computing power.

How the Merge Works When You Are Upgrading from Siebel 6.x FINS

The repository merge does the following things:

- Identifies customizations you have made in the Prior Customer Repository and uses decision criteria to merge them into the New Customer Repository
- Identifies logical schema changes you have made in the Prior Customer Repository and tries to include them in the logical schema in the New Customer Repository
- Upgrades both the base repository and all locales
- Writes a log describing actions, conflicts, and errors
- After the merge is complete, launches postmerge utilities that perform further upgrades to layout objects such as Web templates and controls

The merge process makes numerous record comparisons and inserts. As a result, the merge is resource-intensive and typically takes five to seven hours to complete. The time required varies depending on the complexity of customizations and amount of computing power.

How the Merge Handles Your Customizations

The repository merge handles customizations in the Prior Customer Repository as follows:

- If an object has not changed in the New Siebel Repository but was customized in the Prior Customer Repository, the customization is added to the object in the New Customer Repository.
- If a new object was created in the Prior Customer Repository, it is added to the New Customer Repository.
- If an object is present in the Prior Standard Repository and Prior Customer Repository, but is not present in the New Siebel Repository, the object is obsolete and is not used in the New Customer Repository.

- If an object attribute has changed in the New Siebel Repository and also has changed in the Prior Customer Repository, a conflict occurs. An alternative definition for conflicts is that the attribute value is different in all three repositories: Prior Standard, Prior Customer, and New Siebel.

The merge resolves about 90% of conflicts by using the object attribute as it is defined in the New Siebel Repository.

- If an object was deleted from the Prior Customer Repository, it is added to the New Customer Repository. The following objects are exceptions. If they were deleted from the Prior Customer Repository, they are not added to the New Customer Repository:
 - Control
 - List Column
 - Page Tab
 - Chart
 - Applet Web Template Item
 - View Web Template Item

About the Siebel Database Server

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

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

The Siebel Database Server files will be installed at the same directory level as the Siebel Server in a directory called `dbsrvr`. For example, if the Siebel Server is installed in `C:\sea7xx\si ebsrvr` (Windows), then the Siebel Database Server will be installed in `C:\sea7xx\dbsrvr`. You need to install only one Siebel Database Server.

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

4

How to Perform the Upgrade

This chapter contains the following topics:

- [“Road Map for Performing the Upgrade” on page 43](#)
- [“Process of Planning the Upgrade” on page 45](#)
- [“Process of Upgrading a 6.x Development Environment from the Midtier” on page 45](#)
- [“Process of Upgrading a 6.x Development Environment from the Mainframe” on page 53](#)
- [“Process of Upgrading a 7.x Development Environment from the Midtier” on page 62](#)
- [“Process of Upgrading a 7.x Development Environment from the Mainframe” on page 69](#)
- [“Process of Upgrading a Production Environment” on page 76](#)

Related Topics

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

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

Road Map for Performing the Upgrade

Upgrades: All upgrades.

Environments: All environments.

Use one of the following roadmaps to guide you through the steps for upgrading your Siebel database.

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

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

Choose one of the following road maps to perform the upgrade:

- [“Road Map for Upgrading from Siebel 6.x” on page 44](#)
- [“Road Map for Upgrading from Siebel 7.x” on page 44](#)
- [“Road Map for Upgrading Without a Development Environment” on page 44](#)

Road Map for Upgrading from Siebel 6.x

If you are upgrading from Release 6.x, complete the processes in this road map in the order shown:

- 1 “Process of Planning the Upgrade” on page 45
- 2 “Process of Upgrading a 6.x Development Environment from the Midtier” on page 45
- 3 “Process of Upgrading a 6.x Development Environment from the Mainframe” on page 53
- 4 “Process of Upgrading a Production Environment” on page 76

NOTE: Depending on your business requirements, you can upgrade your development environment from the midtier (midtier-centric approach) or mainframe (mainframe-centric approach). For more information, see [“About Midtier-Centric and Mainframe-Centric Development Environment Upgrades”](#) on page 29.

Road Map for Upgrading from Siebel 7.x

If you are upgrading from Release 7.x, complete the processes in this road map in the order shown:

- 1 “Process of Planning the Upgrade” on page 45
- 2 “Process of Upgrading a 7.x Development Environment from the Midtier” on page 62
- 3 “Process of Upgrading a 7.x Development Environment from the Mainframe” on page 69
- 4 “Process of Upgrading a Production Environment” on page 76

NOTE: Depending on your business requirements, you can upgrade your development environment from the midtier (midtier-centric approach) or mainframe (mainframe-centric approach). For more information, see [“About Midtier-Centric and Mainframe-Centric Development Environment Upgrades”](#) on page 29.

Road Map for Upgrading Without a Development Environment

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

Upgrading without a development environment means the following are true:

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

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

- 1 “Process of Planning the Upgrade” on page 45.
- 2 “Preparing for a No-Development-Environment Upgrade” on page 175.

- 3 Upgrade your production environment. See [“Process of Upgrading a Production Environment” on page 76.](#)

Process of Planning the Upgrade

Upgrades: All upgrades.

Environments: All environments.

This process is part of an upgrade road map. See the road maps earlier in this chapter.

To plan the upgrade, read the following:

- 1 [Chapter 3, “How the Upgrade Works.”](#)
- 2 [Chapter 5, “Database and UI Upgrade Planning.”](#)
- 3 [Chapter 7, “Application Upgrade Planning.”](#)
- 4 [Appendix A, “Schema Changes.”](#)
- 5 Review upgrade-related publications on Siebel SupportWeb. See [“Important Upgrade Planning Resources” on page 23.](#)

Process of Upgrading a 6.x Development Environment from the Midtier

Upgrades: Release 6.x only.

Environments: Development (midtier-centric) environment.

This process is part of an upgrade road map. See the road maps earlier in this chapter.

This topic lists the steps required to upgrade a 6.x development environment from the midtier to a production environment. Print this topic and use it as a checklist for doing the upgrade.

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

Upgrade the Servers

Verify you have identified all the maintenance releases, Fix Packs, and quick-fix patches required for the upgrade. These requirements are located on Siebel SupportWeb under Product Documentation > Maintenance Release Guides.

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

- 1 Install Siebel Tools on development workstations. Keep at least one copy of the previous version of Siebel Tools. You will need it to perform repository operations before the repository merge.

- 2 Install the Siebel Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE). In many cases, you do not have to uninstall the Siebel Servers or SWSE before installing the new versions.
- 3 Install the Siebel Database Server files on the Siebel Server you will use to perform the upgrade.
- 4 Install language packs for your currently deployed languages and any new languages.

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

Upgrade Third-Party Software

- 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 RDBMS

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

Identify and Resolve Duplicate Row IDs

- 1 Identify duplicate row IDs.

After you install the Siebel Database Server software, but before you upgrade to Release 7.7, you must identify and resolve any duplicate row IDs in your Siebel Database.

To identify duplicate row IDs, run the script, `Find_DUP_S_PARTY_ROW_IDS.sql`, located in the `DBSRVR_ROOT\DATABASE_PLATFORM` directory. This SQL script generates a list of duplicate row IDs (if any) that you must resolve prior to beginning the upgrade to Siebel 7.7.

2 Resolve duplicate row IDs.

NOTE: During an upgrade from Release 6.x to Siebel 7, 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 source 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.

To resolve a duplicate row ID in S_EMPLOYEE and S_CONTACT for example, you can copy the existing record in S_CONTACT to generate a new contact with a new row ID and then merge the two records using the Merge Records command.

Data from the record you select as the surviving record is preserved. Data from the other records is lost. If there are other records associated with the records you merge, those records are associated with the surviving record. So, when choosing the record you want to merge, consider choosing one with the smallest number of references. The number of references depends on the foreign key relationships and the data shape in your Siebel Database. See ["Resolving Duplicate Row IDs" on page 137](#).

Preupgrade Tasks for the Siebel Database Upgrade

These steps apply to all database types.

- 1 Review Siebel Technical Notes and Siebel Alerts on upgrading to the current release. Make revisions to the upgrade process as required. See ["Important Upgrade Planning Resources" on page 23](#).
- 2 Review Technical Note 521. It contains important changes to database records and repository objects that must be made during the upgrade. See ["Important Upgrade Planning Resources" on page 23](#).
- 3 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 4 Verify all developers have checked in their projects.
- 5 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 6 Stop the Siebel Server and the Siebel Gateway Name Server.
- 7 Verify there are no open database connections.
- 8 ["Preparing Tables and Views for Upgrade" on page 101](#).
- 9 ["Preparing Custom Indexes for Upgrade" on page 102](#).
- 10 ["Exporting Interface Table Data" on page 102](#).
- 11 ["Preserving Dock Objects and Visibility Rules" on page 103](#).
- 12 ["Verifying Database Server Configuration" on page 103](#).
- 13 ["Securing AIX Memory Allocation Segment Space" on page 104](#).

- 14 "Creating Storage Groups" on page 104.
- 15 "Preparing the Storage Layout of the Schema" on page 104.
- 16 "Reviewing Tablespace Size" on page 115.
- 17 "Backing Up the Database" on page 118.
- 18 "Adding a Siebel User" on page 118.

Preupgrade Tasks for DB2 UDB for z/OS

- 1 "Prerequisites for a Development Upgrade" on page 143.
- 2 "Checking In Development Repository Projects" on page 145.
- 3 "Saving Interface Table Data" on page 145.
- 4 "Upgrading Your DB2 Software" on page 146.
- 5 "Preparing EIM Tables for Upgrade" on page 146.
- 6 "Preparing for Intersection Table Maintenance and Deduplication" on page 148.
- 7 "Determining Which Template File was Used During an Extract or Merge" on page 149.
- 8 "Disabling Customized Triggers" on page 150.
- 9 "Dropping Customized Views" on page 150.
- 10 "Preparing the Development Environment Repositories for Upgrade" on page 150.

Preupgrade Tasks for Application Data

- 1 "Preparing Workflow Processes for Upgrade" on page 131.
- 2 "Preparing Mobile and Dedicated User Data for the Database Upgrade" on page 132.

The remaining tasks are optional depending on currently installed Siebel products and whether the data in the development environment needs to be preserved or upgraded. Review and perform these tasks as necessary.

- 3 "Setting Up Campaign Status Values for Siebel Marketing Upgrade" on page 133.
- 4 "Preparing Address Data for Upgrade" on page 135.
- 5 "Preparing Products for Upgrade" on page 136.
- 6 "Preserving Marketing Segment Descriptions" on page 136.
- 7 "Preparing Siebel eChannel Data for Upgrade" on page 137.
- 8 "Preparing Forecasting Data for Upgrade" on page 138.
- 9 "Preparing Products and Quotes for Upgrade" on page 138.
- 10 "Verifying Household Data Integrity" on page 139.
- 11 "Preparing Financial Services Application Tables for Upgrade" on page 141.
- 12 "Preparing Siebel Configurator Data for Upgrade" on page 141.

Upgrade Siebel Database Schema (upgrep)

- 1 Back up the development database. (If you backed up the database as part of an RDBMS upgrade, ignore this step.)
- 2 [“Renaming the Tools Repository” on page 254.](#)
- 3 Run the Database Server Configuration utility:
 - [“Preparing to Run the Database Server Configuration Utility” on page 160.](#)
 - [“Running the Database Server Configuration Utility Under Windows” on page 163.](#)
 - [“Running the Database Server Configuration Utility Under UNIX” on page 164.](#)

When running the Database Server Configuration Utility, choose the following settings:

- Upgrade Options:** Upgrade Siebel Database Schema (upgrep + upgphys)
 - Environment Type:** Development
- 4 Review the following topic and perform the procedure if applicable: [“Migrating Address Data from Custom Extension Columns” on page 172.](#)
 - 5 [“Executing SQL Files After the First Pause” on page 177.](#)
 - 6 [“Resuming the Siebel Upgrade Wizard After the First Pause” on page 178.](#)
 - 7 [“Executing SQL Output Files After the Second Pause” on page 180.](#)
 - 8 [“Resuming the Siebel Upgrade Wizard After the Second Pause” on page 182.](#)
 - 9 [“Executing SQL Output Files After the Third Pause” on page 184.](#)
 - 10 [“Resuming the Siebel Upgrade Wizard After the Third Pause” on page 185.](#)
 - 11 This step intentionally left blank.
 - 12 [“Synchronizing the Schema” on page 186.](#)
 - 13 [“Transferring Dedup Files” on page 187.](#)
 - 14 [“Migrating Custom Business Component Configurations” on page 188.](#)
 - 15 [“Reviewing Upgrade Log Files for Errors” on page 241.](#)
 - 16 If the upgrade contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Server Configuration utility.
 - 17 [“Manually Archiving Upgrade Log Files” on page 243.](#)
 - 18 [“Installing New License Keys After Upgrade” on page 176.](#)
 - 19 Back up the upgraded database.

Prepare for Repository Merge

- 1 Set the Upgrade Ancestor property for copied objects. See [“Configuring Objects to Inherit Upgrade Behavior”](#) on page 253.
- 2 [“Migrating Repository Objects to the Standard UI”](#) on page 255.
- 3 [“Running the Repository Preparation Wizard on Release 6.x Repository”](#) on page 259.
- 4 Run database statistics on the Siebel Database. If upgrading from Release 7.0.x or 7.5.x, run statistics specifically on S_SYM_STR and SYM_STR_INT. Also, see [“Updating Statistics”](#) on page 187.
Running statistics on the Siebel Database improves merge performance.

Perform Repository Merge

- 1 [“Performing a Repository Merge”](#) on page 262.
- 2 [“Determining if a Repository Merge was Successful”](#) on page 271.
- 3 If the repository merge contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Server Configuration utility.
- 4 Back up the database.

Migrate 6.x Customizations

- 1 [“Identifying What Will Be Migrated”](#) on page 293.
- 2 [“Exposing Hidden Properties of Applets and Views”](#) on page 294.
- 3 [“Running the Siebel Web Client Migration Wizard”](#) on page 295.
- 4 [“Reviewing Applet and View Migration”](#) on page 298.
- 5 [“Running the Web Layout Wizard”](#) on page 299.
- 6 [“Migrating Scripts Attached to Controls”](#) on page 300.
- 7 [“Migrating Scripts Attached to Applets”](#) on page 302.
- 8 [“Migrating Business Component, Business Service, and Application Scripts”](#) on page 305.
- 9 [“Migrating Outbound COM Interfaces”](#) on page 306.
- 10 [“Migrating Inbound COM Interfaces”](#) on page 307.
- 11 [“Regenerating the Postmerge Utilities Log”](#) on page 309.

Run the Postmerge Utilities

- 1 [“Running the Postmerge Utilities”](#) on page 313.
- 2 (Optional.) Compile an SRF file to help review the UI. See [“Creating a New SRF File”](#) on page 282.

- 3 “Reviewing Attribute Conflicts in the Repository Merge” on page 273.
- 4 “Troubleshooting Postmerge UI Problems” on page 314.
- 5 “Verifying Business Address Applet Configuration” on page 315.
- 6 “Reviewing Grid-Based Applets After the Merge” on page 316.
- 7 “Reviewing UI Navigation After the Merge” on page 317.
- 8 “Reviewing Multi-Value Group (MVG) Shuttle Applets After the Merge” on page 318.
- 9 “Revising Rich Text Controls After the Merge” on page 320.
- 10 “Reviewing New Aggregate Categories After the Merge” on page 321.
- 11 “Revising Visibility Filters to Display Correctly After the Merge” on page 322.
- 12 “Assigning a Category and Type to Chart Views After the Merge” on page 323.
- 13 “Assigning a Category and Type to Explorer Views After the Merge” on page 324.
- 14 “Setting Up Navigation to Inaccessible Detail Views After the Merge” on page 324.
- 15 “Eliminating Obsolete UI Fields After the Merge” on page 325.
- 16 “Reviewing UI Objects Affected by ICL After the Merge” on page 326.
- 17 “Reviewing Required Fields in the UI After Merge” on page 326.
- 18 “Setting Label Alignment for Text Fields” on page 327.
- 19 “Assigning an Item Identifier to Web Template Items” on page 327.
- 20 “Generating EIM Temporary Columns After a Repository Merge” on page 277.

Upgrade Custom Database Schema (upgphys)

- 1 Run the Database Server Configuration utility:
 - “Preparing to Run the Database Server Configuration Utility” on page 160.
 - “Running the Database Server Configuration Utility Under Windows” on page 163.
 - “Running the Database Server Configuration Utility Under UNIX” on page 164.

When running the Database Server Configuration Utility, choose the following settings:

- Upgrade Options:** Upgrade Siebel Database Schema (upgrep + upgphys)
 - Environment Type:** Development
- 2 “Reviewing Upgrade Log Files for Errors” on page 241.
 - 3 If the upgrade contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Server Configuration utility.
 - 4 “Manually Archiving Upgrade Log Files” on page 243.

- 5 Back up the upgraded database.

Postupgrade Tasks for Environment Setup

- 1 “Updating File System Attachments” on page 374.
- 2 “Creating a New SRF File” on page 282.
- 3 Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide* and “Regenerating the Database Template File” on page 334.
- 4 Extract developers’ databases. See *Siebel Remote and Replication Manager Administration Guide* and “Extracting Developers or Siebel Tools Clients” on page 334.
- 5 Initialize the local database on development machines.
- 6 “Checking for Inactivated EIM Table Columns” on page 371.
- 7 Reset upgrade-specific parameters back to their defaults. See “Resetting Database Server Configuration Parameters” on page 335.
- 8 Run database statistics. For more information, see “Updating Statistics” on page 187.

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

Postupgrade Tasks for Configuration

- 1 “Reviewing Deleted Objects in the Repository Merge” on page 275.
- 2 “Reviewing Obsolete Objects in the Repository Merge” on page 275.
- 3 “Reviewing Customized Business Components” on page 276.
- 4 “Reapplying Custom Extensions and FK Columns on Obsolete Tables” on page 329.
- 5 “Resolving Business Component and Join Conflicts” on page 331.
- 6 “Deleting Unneeded Repository Files” on page 280.
- 7 If you exported data from interface tables before the upgrade, review the database and import the data as desired.
- 8 “Upgrading to RC2 or AES Encryption” on page 93.

Configure for Globalization

- 1 Install language packs for new languages. See the *Siebel Installation Guide* for the operating system you are using.
- 2 “Upgrading to the Symbolic String Model” on page 335.
- 3 “Setting Up Your Environment to Support Global Time Zone” on page 342.

Deploy Workflows

To perform these tasks, see *Siebel Business Process Designer Administration Guide*.

- 1 Activate and deploy workflows. See *Siebel Business Process Designer Administration Guide*.
- 2 “Upgrading Seeded Workflows” on page 368.
- 3 “Upgrading Inbound Workflows” on page 368.

Verify Application Integration

- 1 Verify that EAI and EIM integrations are set up correctly. For information on using EAI and EIM, see *Overview: Siebel Enterprise Application Integration* and *Siebel Enterprise Integration Manager Administration Guide*.
- 2 “Updating Enterprise Application Integration (EAI) After Upgrade” on page 343.

Postupgrade Tasks for Application Administration

- 1 Review the results of the Person and Organization merge. Make configuration changes as required.
- 2 “Generating Reporting Relationships After Upgrade” on page 340.

Perform System Tests

- 1 Perform the tasks in [Chapter 20, “Postupgrade Tasks for Applications”](#) as needed to upgrade test data in the development environment.
- 2 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
- 3 If you revise repository objects or schema definitions, regenerate the schema.dll and custrep.dat files. See [“Regenerating the Repository Definition Files”](#) on page 278.

Process of Upgrading a 6.x Development Environment from the Mainframe

Upgrades: Release 6.x only.

Environments: Development (mainframe-centric) environment.

This process is part of an upgrade road map. See the road maps earlier in this chapter.

This topic lists the steps required to upgrade a 6.x development environment from the mainframe to a production environment. Print this topic and use it as a checklist for doing the upgrade.

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

Upgrade the Servers

Verify you have identified all the maintenance releases, Fix Packs, and quick-fix patches required for the upgrade. These requirements are located on Siebel SupportWeb under Product Documentation > Maintenance Release Guides.

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

- 1 Install Siebel Tools on development workstations. Keep at least one copy of the previous version of Siebel Tools. You will need it to perform repository operations before the repository merge.
- 2 Install the Siebel Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE). In many cases, you do not have to uninstall the Siebel Servers or SWSE before installing the new versions.
- 3 Install the Siebel Database Server files on the Siebel Server you will use to perform the upgrade.
- 4 Install language packs for your currently deployed languages and any new languages.

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

Upgrade Third-Party Software

- 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 RDBMS

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

Identify and Resolve Duplicate Row IDs

- 1 Identify duplicate row IDs.

After you install the Siebel Database Server software, but before you upgrade to Release 7.7, you must identify and resolve any duplicate row IDs in your Siebel Database.

To identify duplicate row IDs, run the script, `Find_DUP_S_PARTY_ROW_IDS.sql`, located in the `DBSRVR_ROOT\DATABASE_PLATFORM` directory. This SQL script generates a list of duplicate row IDs (if any) that you must resolve prior to beginning the upgrade to Siebel 7.7.

2 Resolve duplicate row IDs.

NOTE: During an upgrade from Release 6.x to Siebel 7, 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 source 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.

To resolve a duplicate row ID in S_EMPLOYEE and S_CONTACT for example, you can copy the existing record in S_CONTACT to generate a new contact with a new row ID and then merge the two records using the Merge Records command.

Data from the record you select as the surviving record is preserved. Data from the other records is lost. If there are other records associated with the records you merge, those records are associated with the surviving record. So, when choosing the record you want to merge, consider choosing one with the smallest number of references. The number of references depends on the foreign key relationships and the data shape in your Siebel Database. See ["Resolving Duplicate Row IDs" on page 137](#).

Preupgrade Tasks for the Siebel Database Upgrade

These steps apply to all database types.

- 1 Review Siebel Technical Notes and Siebel Alerts on upgrading to the current release. Make revisions to the upgrade process as required. See ["Important Upgrade Planning Resources" on page 23](#).
- 2 Review Technical Note 521. It contains important changes to database records and repository objects that must be made during the upgrade. See ["Important Upgrade Planning Resources" on page 23](#).
- 3 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 4 Verify all developers have checked in their projects.
- 5 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 6 Stop the Siebel Server and the Siebel Gateway Name Server.
- 7 Verify there are no open database connections.
- 8 ["Preparing Tables and Views for Upgrade" on page 101](#).
- 9 ["Preparing Custom Indexes for Upgrade" on page 102](#).
- 10 ["Exporting Interface Table Data" on page 102](#).
- 11 ["Preserving Dock Objects and Visibility Rules" on page 103](#).
- 12 ["Verifying Database Server Configuration" on page 103](#).
- 13 ["Securing AIX Memory Allocation Segment Space" on page 104](#).

- 14 "Creating Storage Groups" on page 104.
- 15 "Preparing the Storage Layout of the Schema" on page 104.
- 16 "Reviewing Tablespace Size" on page 115.
- 17 "Backing Up the Database" on page 118.
- 18 "Adding a Siebel User" on page 118.

Preupgrade Tasks for DB2 UDB for z/OS

- 1 "Prerequisites for a Development Upgrade" on page 143.
- 2 "Checking In Development Repository Projects" on page 145.
- 3 "Saving Interface Table Data" on page 145.
- 4 "Upgrading Your DB2 Software" on page 146.
- 5 "Preparing EIM Tables for Upgrade" on page 146.
- 6 "Preparing for Intersection Table Maintenance and Deduplication" on page 148.
- 7 "Determining Which Template File was Used During an Extract or Merge" on page 149.
- 8 "Disabling Customized Triggers" on page 150.
- 9 "Dropping Customized Views" on page 150.
- 10 "Preparing the Development Environment Repositories for Upgrade" on page 150.

Preupgrade Tasks for Application Data

- 1 "Preparing Workflow Processes for Upgrade" on page 131.
- 2 "Preparing Mobile and Dedicated User Data for the Database Upgrade" on page 132.

The remaining tasks are optional depending on currently installed Siebel products and whether the data in the development environment needs to be preserved or upgraded. Review and perform these tasks as necessary.

- 3 "Setting Up Campaign Status Values for Siebel Marketing Upgrade" on page 133.
- 4 "Preparing Address Data for Upgrade" on page 135.
- 5 "Preparing Products for Upgrade" on page 136.
- 6 "Preserving Marketing Segment Descriptions" on page 136.
- 7 "Preparing Siebel eChannel Data for Upgrade" on page 137.
- 8 "Preparing Forecasting Data for Upgrade" on page 138.
- 9 "Preparing Products and Quotes for Upgrade" on page 138.
- 10 "Verifying Household Data Integrity" on page 139.
- 11 "Preparing Financial Services Application Tables for Upgrade" on page 141.
- 12 "Preparing Siebel Configurator Data for Upgrade" on page 141.

Upgrade Siebel Database Schema (upgrep)

- 1 Back up the development database. (If you backed up the database as part of an RDBMS upgrade, ignore this step.)
- 2 [“Renaming the Tools Repository” on page 254.](#)
- 3 Run the Database Server Configuration utility:
 - [“Preparing to Run the Database Server Configuration Utility” on page 160.](#)
 - [“Running the Database Server Configuration Utility Under Windows” on page 163.](#)
 - [“Running the Database Server Configuration Utility Under UNIX” on page 164.](#)

When running the Database Server Configuration Utility, choose the following settings:

- Upgrade Options:** Upgrade Siebel Database Schema (upgrep + upgphys)
 - Environment Type:** Development
- 4 Review the following topic and perform the procedure if applicable: [“Migrating Address Data from Custom Extension Columns” on page 172.](#)
 - 5 [“Transferring the Customized Repository to z/OS Host” on page 189.](#)
 - 6 [“Preparing the zSeries Host Environment” on page 190.](#)
 - 7 [“Resuming the Siebel Upgrade Wizard After the First Pause” on page 199.](#)
 - 8 [“Transferring Control Cards and Schema DDL Files to the zSeries Host” on page 200.](#)
 - 9 [“Preparing the Schema and JCL Files on the zSeries Host” on page 201.](#)
 - 10 [“Installing Stored Procedures on the zSeries Host” on page 205.](#)
 - 11 [“Preparing Siebel-Scheduled Jobs on the zSeries Host” on page 206.](#)
 - 12 [“Optimizing the Source Database Job Stream” on page 210.](#)
 - 13 [“Optimizing the Source Database Job Stream” on page 210.](#)
 - 14 [“Loading Log Tables on the Source Database” on page 210.](#)
 - 15 [“Loading Log Tables on the Target Database” on page 211.](#)
 - 16 [“Executing the Upgrade on the zSeries Host \(Phase 1 of 2\)” on page 212.](#)
 - 17 [“Resuming the Siebel Upgrade Wizard on the Midtier After the Second Pause” on page 222.](#)
 - 18 [“Transferring the Index Schema to the zSeries Host” on page 224.](#)
 - 19 [“Preparing Siebel-Scheduled JCL” on page 225.](#)
 - 20 [“Optimizing the Source Database Job Stream” on page 210.](#)
 - 21 [“Executing the Upgrade on the zSeries Host \(Phase 2 of 2\)” on page 228.](#)
 - 22 [“Performing Intersection Table Maintenance” on page 234.](#)
 - 23 [“Reviewing Upgrade Log Files for Errors” on page 241.](#)
 - 24 If the upgrade contains unacceptable errors, do the following:
 - a Restore the backup of the database.

- b** Correct the errors.
- c** Rerun the Database Server Configuration utility.

- 25** [“Manually Archiving Upgrade Log Files” on page 243.](#)
- 26** [“Installing New License Keys After Upgrade” on page 176.](#)
- 27** Back up the upgraded database.

Prepare for Repository Merge

- 1** Set the Upgrade Ancestor property for copied objects. See [“Configuring Objects to Inherit Upgrade Behavior” on page 253.](#)
- 2** [“Migrating Repository Objects to the Standard UI” on page 255.](#)
- 3** [“Running the Repository Preparation Wizard on Release 6.x Repository” on page 259.](#)
- 4** Run database statistics on the Siebel Database. If upgrading from Release 7.0.x or 7.5.x, run statistics specifically on S_SYM_STR and SYM_STR_INT. Also, see [“Generating RUNSTATS” on page 234.](#)

Running statistics on the Siebel Database improves merge performance.

Perform Repository Merge

- 1** [“Performing a Repository Merge” on page 262.](#)
- 2** [“Determining if a Repository Merge was Successful” on page 271.](#)
- 3** If the repository merge contains unacceptable errors, do the following:
 - a** Restore the backup of the database.
 - b** Correct the errors.
 - c** Rerun the Database Server Configuration utility.
- 4** Back up the database.

Migrate 6.x Customizations

- 1** [“Identifying What Will Be Migrated” on page 293.](#)
- 2** [“Exposing Hidden Properties of Applets and Views” on page 294.](#)
- 3** [“Running the Siebel Web Client Migration Wizard” on page 295.](#)
- 4** [“Reviewing Applet and View Migration” on page 298.](#)
- 5** [“Running the Web Layout Wizard” on page 299.](#)
- 6** [“Migrating Scripts Attached to Controls” on page 300.](#)
- 7** [“Migrating Scripts Attached to Applets” on page 302.](#)
- 8** [“Migrating Business Component, Business Service, and Application Scripts” on page 305.](#)

- 9 “Migrating Outbound COM Interfaces” on page 306.
- 10 “Migrating Inbound COM Interfaces” on page 307.
- 11 “Regenerating the Postmerge Utilities Log” on page 309.

Run the Postmerge Utilities

- 1 “Running the Postmerge Utilities” on page 313.
- 2 (Optional.) Compile an SRF file to help review the UI. See “Creating a New SRF File” on page 282.
- 3 “Reviewing Attribute Conflicts in the Repository Merge” on page 273.
- 4 “Troubleshooting Postmerge UI Problems” on page 314.
- 5 “Verifying Business Address Applet Configuration” on page 315.
- 6 “Reviewing Grid-Based Applets After the Merge” on page 316.
- 7 “Reviewing UI Navigation After the Merge” on page 317.
- 8 “Reviewing Multi-Value Group (MVG) Shuttle Applets After the Merge” on page 318.
- 9 “Revising Rich Text Controls After the Merge” on page 320.
- 10 “Reviewing New Aggregate Categories After the Merge” on page 321.
- 11 “Revising Visibility Filters to Display Correctly After the Merge” on page 322.
- 12 “Assigning a Category and Type to Chart Views After the Merge” on page 323.
- 13 “Assigning a Category and Type to Explorer Views After the Merge” on page 324.
- 14 “Setting Up Navigation to Inaccessible Detail Views After the Merge” on page 324.
- 15 “Eliminating Obsolete UI Fields After the Merge” on page 325.
- 16 “Reviewing UI Objects Affected by ICL After the Merge” on page 326.
- 17 “Reviewing Required Fields in the UI After Merge” on page 326.
- 18 “Setting Label Alignment for Text Fields” on page 327.
- 19 “Assigning an Item Identifier to Web Template Items” on page 327.
- 20 “Generating EIM Temporary Columns After a Repository Merge” on page 277.

Upgrade Custom Database Schema (upgphys)

- 1 Run the Database Server Configuration utility:
 - “Preparing to Run the Database Server Configuration Utility” on page 160.
 - “Running the Database Server Configuration Utility Under Windows” on page 163.
 - “Running the Database Server Configuration Utility Under UNIX” on page 164.

When running the Database Server Configuration Utility, choose the following settings:

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

- **Environment Type:** Development
- 2 [“Reviewing Upgrade Log Files for Errors” on page 241.](#)
- 3 If the upgrade contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Server Configuration utility.
- 4 [“Manually Archiving Upgrade Log Files” on page 243.](#)
- 5 Back up the upgraded database.

Postupgrade Tasks for Environment Setup

- 1 [“Updating File System Attachments” on page 374.](#)
- 2 [“Creating a New SRF File” on page 282.](#)
- 3 Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide* and [“Regenerating the Database Template File” on page 334.](#)
- 4 Extract developers’ databases. See *Siebel Remote and Replication Manager Administration Guide* and [“Extracting Developers or Siebel Tools Clients” on page 334.](#)
- 5 Initialize the local database on development machines.
- 6 [“Checking for Inactivated EIM Table Columns” on page 371.](#)
- 7 Reset upgrade-specific parameters back to their defaults. See [“Resetting Database Server Configuration Parameters” on page 335.](#)
- 8 Run database statistics. For more information, see [“Generating RUNSTATS” on page 234.](#)

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

Postupgrade Tasks for Configuration

- 1 [“Reviewing Deleted Objects in the Repository Merge” on page 275.](#)
- 2 [“Reviewing Obsolete Objects in the Repository Merge” on page 275.](#)
- 3 [“Reviewing Customized Business Components” on page 276.](#)
- 4 [“Reapplying Custom Extensions and FK Columns on Obsolete Tables” on page 329.](#)
- 5 [“Resolving Business Component and Join Conflicts” on page 331.](#)
- 6 [“Deleting Unneeded Repository Files” on page 280.](#)
- 7 If you exported data from interface tables before the upgrade, review the database and import the data as desired.
- 8 [“Upgrading to RC2 or AES Encryption” on page 93.](#)

Configure for Globalization

- 1 Install language packs for new languages. See the *Siebel Installation Guide* for the operating system you are using.
- 2 [“Upgrading to the Symbolic String Model” on page 335.](#)
- 3 [“Setting Up Your Environment to Support Global Time Zone” on page 342.](#)

Deploy Workflows

To perform these tasks, see *Siebel Business Process Designer Administration Guide*.

- 1 Activate and deploy workflows. See *Siebel Business Process Designer Administration Guide*.
- 2 [“Upgrading Seeded Workflows” on page 368.](#)
- 3 [“Upgrading Inbound Workflows” on page 368.](#)

Verify Application Integration

- 1 Verify that EAI and EIM integrations are set up correctly. For information on using EAI and EIM, see *Overview: Siebel Enterprise Application Integration* and *Siebel Enterprise Integration Manager Administration Guide*.
- 2 [“Updating Enterprise Application Integration \(EAI\) After Upgrade” on page 343.](#)

Postupgrade Tasks for Application Administration

- 1 Review the results of the Person and Organization merge. Make configuration changes as required.
- 2 [“Generating Reporting Relationships After Upgrade” on page 340.](#)

Perform System Tests

- 1 Perform the tasks in [Chapter 20, “Postupgrade Tasks for Applications”](#) as needed to upgrade test data in the development environment.
- 2 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
- 3 If you revise repository objects or schema definitions, regenerate the schema.dll and custrep.dat files. See [“Regenerating the Repository Definition Files” on page 278.](#)

Process of Upgrading a 7.x Development Environment from the Midtier

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

Environments: Development (midtier-centric) environment.

This process is part of an upgrade road map. See the road maps earlier in this chapter.

This topic lists the steps required to upgrade a 7.x development environment from the midtier to a production environment. Print this topic and use it as a checklist for doing the upgrade.

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

Upgrade the Servers

Verify you have identified all the maintenance releases, Fix Packs, and quick-fix patches required for the upgrade. These requirements are located on Siebel SupportWeb under Product Documentation > Maintenance Release Guides.

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

- 1 Install Siebel Tools on development workstations. Keep at least one copy of the previous version of Siebel Tools. You will need it to perform repository operations before the repository merge.
- 2 Install the Siebel Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE). In many cases, you do not have to uninstall the Siebel Servers or SWSE before installing the new versions.
- 3 Install the Siebel Database Server files on the Siebel Server you will use to perform the upgrade.
- 4 Install language packs for your currently deployed languages and any new languages.

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

Upgrade Third-Party Software

- 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 RDBMS

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

Preupgrade Tasks for the Siebel Database Upgrade

These steps apply to all database types.

- 1 Review Siebel Technical Notes and Siebel Alerts on upgrading to the current release. Make revisions to the upgrade process as required. See [“Important Upgrade Planning Resources” on page 23](#).
- 2 Review Technical Note 521. It contains important changes to database records and repository objects that must be made during the upgrade. See [“Important Upgrade Planning Resources” on page 23](#).
- 3 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 4 Verify all developers have checked in their projects.
- 5 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 6 Stop the Siebel Server and the Siebel Gateway Name Server.
- 7 Verify there are no open database connections.
- 8 [“Preparing Tables and Views for Upgrade” on page 101](#).
- 9 [“Preparing Custom Indexes for Upgrade” on page 102](#).
- 10 [“Exporting Interface Table Data” on page 102](#).
- 11 [“Preserving Dock Objects and Visibility Rules” on page 103](#).
- 12 [“Verifying Database Server Configuration” on page 103](#).
- 13 [“Securing AIX Memory Allocation Segment Space” on page 104](#).
- 14 [“Creating Storage Groups” on page 104](#).
- 15 [“Preparing the Storage Layout of the Schema” on page 104](#).
- 16 [“Reviewing Tablespace Size” on page 115](#).
- 17 [“Backing Up the Database” on page 118](#).
- 18 [“Adding a Siebel User” on page 118](#).

Preupgrade Tasks for DB2 UDB for z/OS

- 1 [“Prerequisites for a Development Upgrade” on page 143](#).
- 2 [“Checking In Development Repository Projects” on page 145](#).
- 3 [“Saving Interface Table Data” on page 145](#).
- 4 [“Upgrading Your DB2 Software” on page 146](#).
- 5 [“Preparing EIM Tables for Upgrade” on page 146](#).
- 6 [“Preparing for Intersection Table Maintenance and Deduplication” on page 148](#).
- 7 [“Determining Which Template File was Used During an Extract or Merge” on page 149](#).

- 8 “Disabling Customized Triggers” on page 150.
- 9 “Dropping Customized Views” on page 150.
- 10 “Preparing the Development Environment Repositories for Upgrade” on page 150.

Preupgrade Tasks for Application Data

- 1 “Preparing Workflow Processes for Upgrade” on page 131.
- 2 “Preparing Mobile and Dedicated User Data for the Database Upgrade” on page 132.

The remaining tasks are optional depending on currently installed Siebel products and whether the data in the development environment needs to be preserved or upgraded. Review and perform these tasks as necessary.

- 3 “Setting Up Campaign Status Values for Siebel Marketing Upgrade” on page 133.
- 4 “Preparing Address Data for Upgrade” on page 135.
- 5 “Preparing Products for Upgrade” on page 136.
- 6 “Preserving Marketing Segment Descriptions” on page 136.
- 7 “Preparing Siebel eChannel Data for Upgrade” on page 137.
- 8 “Preparing Forecasting Data for Upgrade” on page 138.
- 9 “Preparing Products and Quotes for Upgrade” on page 138.
- 10 “Verifying Household Data Integrity” on page 139.
- 11 “Preparing Financial Services Application Tables for Upgrade” on page 141.
- 12 “Preparing Siebel Configurator Data for Upgrade” on page 141.

Upgrade Siebel Database Schema (upgrep)

- 1 Back up the development database. (If you backed up the database as part of an RDBMS upgrade, ignore this step.)
- 2 “Renaming the Tools Repository” on page 254.
- 3 Run the Database Server Configuration utility:
 - “Preparing to Run the Database Server Configuration Utility” on page 160.
 - “Running the Database Server Configuration Utility Under Windows” on page 163.
 - “Running the Database Server Configuration Utility Under UNIX” on page 164.

When running the Database Server Configuration Utility, choose the following settings:

- Upgrade Options:** Upgrade Siebel Database Schema (upgrep + upgphys)
- Environment Type:** Development
- 4 Review the following topic and perform the procedure if applicable: “Migrating Address Data from Custom Extension Columns” on page 172.

- 5 [“Executing SQL Files After the First Pause” on page 177.](#)
- 6 [“Resuming the Siebel Upgrade Wizard After the First Pause” on page 178.](#)
- 7 [“Executing SQL Output Files After the Second Pause” on page 180.](#)
- 8 [“Resuming the Siebel Upgrade Wizard After the Second Pause” on page 182.](#)
- 9 [“Executing SQL Output Files After the Third Pause” on page 184.](#)
- 10 [“Resuming the Siebel Upgrade Wizard After the Third Pause” on page 185.](#)
- 11 This step intentionally left blank.
- 12 [“Synchronizing the Schema” on page 186.](#)
- 13 [“Transferring Dedup Files” on page 187.](#)
- 14 [“Migrating Custom Business Component Configurations” on page 188.](#)
- 15 [“Reviewing Upgrade Log Files for Errors” on page 241.](#)
- 16 If the upgrade contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Server Configuration utility.
- 17 [“Manually Archiving Upgrade Log Files” on page 243.](#)
- 18 [“Installing New License Keys After Upgrade” on page 176.](#)
- 19 Back up the upgraded database.

Prepare for Repository Merge

- 1 Set the Upgrade Ancestor property for copied objects. See [“Configuring Objects to Inherit Upgrade Behavior” on page 253.](#)
- 2 [“Migrating Repository Objects to the Standard UI” on page 255.](#)
- 3 Run database statistics on the Siebel Database. If upgrading from Release 7.0.x or 7.5.x, run statistics specifically on S_SYM_STR and SYM_STR_INT. Also, see [“Updating Statistics” on page 187.](#)
Running statistics on the Siebel Database improves merge performance.

Perform Repository Merge

- 1 [“Performing a Repository Merge” on page 262.](#)
- 2 [“Determining if a Repository Merge was Successful” on page 271.](#)
- 3 If the repository merge contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Server Configuration utility.

- 4 Back up the database.

Run the Postmerge Utilities

- 1 “Running the Postmerge Utilities” on page 313.
- 2 (Optional.) Compile an SRF file to help review the UI. See “Creating a New SRF File” on page 282.
- 3 “Reviewing Attribute Conflicts in the Repository Merge” on page 273.
- 4 “Troubleshooting Postmerge UI Problems” on page 314.
- 5 “Verifying Business Address Applet Configuration” on page 315.
- 6 “Reviewing Grid-Based Applets After the Merge” on page 316.
- 7 “Reviewing UI Navigation After the Merge” on page 317.
- 8 “Reviewing Multi-Value Group (MVG) Shuttle Applets After the Merge” on page 318.
- 9 “Revising Rich Text Controls After the Merge” on page 320.
- 10 “Reviewing New Aggregate Categories After the Merge” on page 321.
- 11 “Revising Visibility Filters to Display Correctly After the Merge” on page 322.
- 12 “Assigning a Category and Type to Chart Views After the Merge” on page 323.
- 13 “Assigning a Category and Type to Explorer Views After the Merge” on page 324.
- 14 “Setting Up Navigation to Inaccessible Detail Views After the Merge” on page 324.
- 15 “Eliminating Obsolete UI Fields After the Merge” on page 325.
- 16 “Reviewing UI Objects Affected by ICL After the Merge” on page 326.
- 17 “Reviewing Required Fields in the UI After Merge” on page 326.
- 18 “Setting Label Alignment for Text Fields” on page 327.
- 19 “Assigning an Item Identifier to Web Template Items” on page 327.
- 20 “Generating EIM Temporary Columns After a Repository Merge” on page 277.

Upgrade Custom Database Schema (upgphys)

- 1 Run the Database Server Configuration utility:
 - “Preparing to Run the Database Server Configuration Utility” on page 160.
 - “Running the Database Server Configuration Utility Under Windows” on page 163.
 - “Running the Database Server Configuration Utility Under UNIX” on page 164.

When running the Database Server Configuration Utility, choose the following settings:

- Upgrade Options:** Upgrade Siebel Database Schema (upgrep + upgphys)
 - Environment Type:** Development
- 2 “Reviewing Upgrade Log Files for Errors” on page 241.

- 3 If the upgrade contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Server Configuration utility.
- 4 [“Manually Archiving Upgrade Log Files” on page 243.](#)
- 5 Back up the upgraded database.

Postupgrade Tasks for Environment Setup

- 1 [“Updating File System Attachments” on page 374.](#)
- 2 [“Creating a New SRF File” on page 282.](#)
- 3 Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide* and [“Regenerating the Database Template File” on page 334.](#)
- 4 Extract developers’ databases. See *Siebel Remote and Replication Manager Administration Guide* and [“Extracting Developers or Siebel Tools Clients” on page 334.](#)
- 5 Initialize the local database on development machines.
- 6 [“Checking for Inactivated EIM Table Columns” on page 371.](#)
- 7 Reset upgrade-specific parameters back to their defaults. See [“Resetting Database Server Configuration Parameters” on page 335.](#)
- 8 Run database statistics. For more information, see [“Updating Statistics” on page 187.](#)

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

Postupgrade Tasks for Configuration

- 1 [“Reviewing Deleted Objects in the Repository Merge” on page 275.](#)
- 2 [“Reviewing Obsolete Objects in the Repository Merge” on page 275.](#)
- 3 [“Reviewing Customized Business Components” on page 276.](#)
- 4 [“Reapplying Custom Extensions and FK Columns on Obsolete Tables” on page 329.](#)
- 5 [“Resolving Business Component and Join Conflicts” on page 331.](#)
- 6 [“Deleting Unneeded Repository Files” on page 280.](#)
- 7 If you exported data from interface tables before the upgrade, review the database and import the data as desired.
- 8 [“Upgrading to RC2 or AES Encryption” on page 93.](#)

Configure for Globalization

- 1 Install language packs for new languages. See the *Siebel Installation Guide* for the operating system you are using.
- 2 [“Upgrading to the Symbolic String Model” on page 335.](#)
- 3 [“Setting Up Your Environment to Support Global Time Zone” on page 342.](#)

Deploy Workflows

To perform these tasks, see *Siebel Business Process Designer Administration Guide*.

- 1 Activate and deploy workflows. See *Siebel Business Process Designer Administration Guide*.
- 2 [“Upgrading Seeded Workflows” on page 368.](#)
- 3 [“Upgrading Inbound Workflows” on page 368.](#)

Verify Application Integration

- 1 Verify that EAI and EIM integrations are set up correctly. For information on using EAI and EIM, see *Overview: Siebel Enterprise Application Integration* and *Siebel Enterprise Integration Manager Administration Guide*.
- 2 [“Updating Enterprise Application Integration \(EAI\) After Upgrade” on page 343.](#)

Postupgrade Tasks for Application Administration

- 1 Review the results of the Person and Organization merge. Make configuration changes as required.
- 2 [“Generating Reporting Relationships After Upgrade” on page 340.](#)

Perform System Tests

- 1 Perform the tasks in [Chapter 20, “Postupgrade Tasks for Applications”](#) as needed to upgrade test data in the development environment.
- 2 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
- 3 If you revise repository objects or schema definitions, regenerate the schema.dll and custrep.dat files. See [“Regenerating the Repository Definition Files” on page 278.](#)

Process of Upgrading a 7.x Development Environment from the Mainframe

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

Environments: Development (mainframe-centric) environment.

This process is part of an upgrade road map. See the road maps earlier in this chapter.

This topic lists the steps required to upgrade a 7.x development environment from the mainframe to a production environment. Print this topic and use it as a checklist for doing the upgrade.

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

Upgrade the Servers

Verify you have identified all the maintenance releases, Fix Packs, and quick-fix patches required for the upgrade. These requirements are located on Siebel SupportWeb under Product Documentation > Maintenance Release Guides.

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

- 1 Install Siebel Tools on development workstations. Keep at least one copy of the previous version of Siebel Tools. You will need it to perform repository operations before the repository merge.
- 2 Install the Siebel Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE). In many cases, you do not have to uninstall the Siebel Servers or SWSE before installing the new versions.
- 3 Install the Siebel Database Server files on the Siebel Server you will use to perform the upgrade.
- 4 Install language packs for your currently deployed languages and any new languages.

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

Upgrade Third-Party Software

- 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 RDBMS

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

Preupgrade Tasks for the Siebel Database Upgrade

These steps apply to all database types.

- 1 Review Siebel Technical Notes and Siebel Alerts on upgrading to the current release. Make revisions to the upgrade process as required. See [“Important Upgrade Planning Resources” on page 23](#).
- 2 Review Technical Note 521. It contains important changes to database records and repository objects that must be made during the upgrade. See [“Important Upgrade Planning Resources” on page 23](#).
- 3 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 4 Verify all developers have checked in their projects.
- 5 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 6 Stop the Siebel Server and the Siebel Gateway Name Server.
- 7 Verify there are no open database connections.
- 8 [“Preparing Tables and Views for Upgrade” on page 101](#).
- 9 [“Preparing Custom Indexes for Upgrade” on page 102](#).
- 10 [“Exporting Interface Table Data” on page 102](#).
- 11 [“Preserving Dock Objects and Visibility Rules” on page 103](#).
- 12 [“Verifying Database Server Configuration” on page 103](#).
- 13 [“Securing AIX Memory Allocation Segment Space” on page 104](#).
- 14 [“Creating Storage Groups” on page 104](#).
- 15 [“Preparing the Storage Layout of the Schema” on page 104](#).
- 16 [“Reviewing Tablespace Size” on page 115](#).
- 17 [“Backing Up the Database” on page 118](#).
- 18 [“Adding a Siebel User” on page 118](#).

Preupgrade Tasks for DB2 UDB for z/OS

- 1 [“Prerequisites for a Development Upgrade” on page 143](#).
- 2 [“Checking In Development Repository Projects” on page 145](#).
- 3 [“Saving Interface Table Data” on page 145](#).
- 4 [“Upgrading Your DB2 Software” on page 146](#).
- 5 [“Preparing EIM Tables for Upgrade” on page 146](#).
- 6 [“Preparing for Intersection Table Maintenance and Deduplication” on page 148](#).
- 7 [“Determining Which Template File was Used During an Extract or Merge” on page 149](#).

- 8 “Disabling Customized Triggers” on page 150.
- 9 “Dropping Customized Views” on page 150.
- 10 “Preparing the Development Environment Repositories for Upgrade” on page 150.

Preupgrade Tasks for Application Data

- 1 “Preparing Workflow Processes for Upgrade” on page 131.
- 2 “Preparing Mobile and Dedicated User Data for the Database Upgrade” on page 132.

The remaining tasks are optional depending on currently installed Siebel products and whether the data in the development environment needs to be preserved or upgraded. Review and perform these tasks as necessary.

- 3 “Setting Up Campaign Status Values for Siebel Marketing Upgrade” on page 133.
- 4 “Preparing Address Data for Upgrade” on page 135.
- 5 “Preparing Products for Upgrade” on page 136.
- 6 “Preserving Marketing Segment Descriptions” on page 136.
- 7 “Preparing Siebel eChannel Data for Upgrade” on page 137.
- 8 “Preparing Forecasting Data for Upgrade” on page 138.
- 9 “Preparing Products and Quotes for Upgrade” on page 138.
- 10 “Verifying Household Data Integrity” on page 139.
- 11 “Preparing Financial Services Application Tables for Upgrade” on page 141.
- 12 “Preparing Siebel Configurator Data for Upgrade” on page 141.

Upgrade Siebel Database Schema (upgrep)

- 1 Back up the development database. (If you backed up the database as part of an RDBMS upgrade, ignore this step.)
- 2 “Renaming the Tools Repository” on page 254.
- 3 Run the Database Server Configuration utility:
 - “Preparing to Run the Database Server Configuration Utility” on page 160.
 - “Running the Database Server Configuration Utility Under Windows” on page 163.
 - “Running the Database Server Configuration Utility Under UNIX” on page 164.

When running the Database Server Configuration Utility, choose the following settings:

- Upgrade Options:** Upgrade Siebel Database Schema (upgrep + upgphys)
- Environment Type:** Development
- 4 Review the following topic and perform the procedure if applicable: “Migrating Address Data from Custom Extension Columns” on page 172.

- 5 “Transferring the Customized Repository to z/OS Host” on page 189.
- 6 “Preparing the zSeries Host Environment” on page 190.
- 7 “Resuming the Siebel Upgrade Wizard After the First Pause” on page 199.
- 8 “Transferring Control Cards and Schema DDL Files to the zSeries Host” on page 200.
- 9 “Preparing the Schema and JCL Files on the zSeries Host” on page 201.
- 10 “Installing Stored Procedures on the zSeries Host” on page 205.
- 11 “Preparing Siebel-Scheduled Jobs on the zSeries Host” on page 206.
- 12 “Optimizing the Source Database Job Stream” on page 210.
- 13 “Optimizing the Source Database Job Stream” on page 210.
- 14 “Loading Log Tables on the Source Database” on page 210.
- 15 “Loading Log Tables on the Target Database” on page 211.
- 16 “Executing the Upgrade on the zSeries Host (Phase 1 of 2)” on page 212.
- 17 “Resuming the Siebel Upgrade Wizard on the Midtier After the Second Pause” on page 222.
- 18 “Transferring the Index Schema to the zSeries Host” on page 224.
- 19 “Preparing Siebel-Scheduled JCL” on page 225.
- 20 “Executing the Upgrade on the zSeries Host (Phase 2 of 2)” on page 228.
- 21 “Performing Intersection Table Maintenance” on page 234.
- 22 “Reviewing Upgrade Log Files for Errors” on page 241.
- 23 If the upgrade contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Server Configuration utility.
- 24 “Manually Archiving Upgrade Log Files” on page 243.
- 25 “Installing New License Keys After Upgrade” on page 176.
- 26 Back up the upgraded database.

Prepare for Repository Merge

- 1 Set the Upgrade Ancestor property for copied objects. See “Configuring Objects to Inherit Upgrade Behavior” on page 253.
- 2 “Migrating Repository Objects to the Standard UI” on page 255.
- 3 Run database statistics on the Siebel Database. If upgrading from Release 7.0.x or 7.5.x, run statistics specifically on S_SYM_STR and SYM_STR_INT. Also, see “Generating RUNSTATS” on page 234.

Running statistics on the Siebel Database improves merge performance.

Perform Repository Merge

- 1 [“Performing a Repository Merge” on page 262.](#)
- 2 [“Determining if a Repository Merge was Successful” on page 271.](#)
- 3 If the repository merge contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Server Configuration utility.
- 4 Back up the database.

Run the Postmerge Utilities

- 1 [“Running the Postmerge Utilities” on page 313.](#)
- 2 (Optional.) Compile an SRF file to help review the UI. See [“Creating a New SRF File” on page 282.](#)
- 3 [“Reviewing Attribute Conflicts in the Repository Merge” on page 273.](#)
- 4 [“Troubleshooting Postmerge UI Problems” on page 314.](#)
- 5 [“Verifying Business Address Applet Configuration” on page 315.](#)
- 6 [“Reviewing Grid-Based Applets After the Merge” on page 316.](#)
- 7 [“Reviewing UI Navigation After the Merge” on page 317.](#)
- 8 [“Reviewing Multi-Value Group \(MVG\) Shuttle Applets After the Merge” on page 318.](#)
- 9 [“Revising Rich Text Controls After the Merge” on page 320.](#)
- 10 [“Reviewing New Aggregate Categories After the Merge” on page 321.](#)
- 11 [“Revising Visibility Filters to Display Correctly After the Merge” on page 322.](#)
- 12 [“Assigning a Category and Type to Chart Views After the Merge” on page 323.](#)
- 13 [“Assigning a Category and Type to Explorer Views After the Merge” on page 324.](#)
- 14 [“Setting Up Navigation to Inaccessible Detail Views After the Merge” on page 324.](#)
- 15 [“Eliminating Obsolete UI Fields After the Merge” on page 325.](#)
- 16 [“Reviewing UI Objects Affected by ICL After the Merge” on page 326.](#)
- 17 [“Reviewing Required Fields in the UI After Merge” on page 326.](#)
- 18 [“Setting Label Alignment for Text Fields” on page 327.](#)
- 19 [“Assigning an Item Identifier to Web Template Items” on page 327.](#)
- 20 [“Generating EIM Temporary Columns After a Repository Merge” on page 277.](#)

Upgrade Custom Database Schema (upgphys)

- 1 Run the Database Server Configuration utility:

- [“Preparing to Run the Database Server Configuration Utility” on page 160.](#)
- [“Running the Database Server Configuration Utility Under Windows” on page 163.](#)
- [“Running the Database Server Configuration Utility Under UNIX” on page 164.](#)

When running the Database Server Configuration Utility, choose the following settings:

- Upgrade Options:** Upgrade Siebel Database Schema (upgrep + upgphys)
 - Environment Type:** Development
- 2 [“Reviewing Upgrade Log Files for Errors” on page 241.](#)
 - 3 If the upgrade contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Server Configuration utility.
 - 4 [“Manually Archiving Upgrade Log Files” on page 243.](#)
 - 5 Back up the upgraded database.

Postupgrade Tasks for Environment Setup

- 1 [“Updating File System Attachments” on page 374.](#)
- 2 [“Creating a New SRF File” on page 282.](#)
- 3 Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide* and [“Regenerating the Database Template File” on page 334.](#)
- 4 Extract developers’ databases. See *Siebel Remote and Replication Manager Administration Guide* and [“Extracting Developers or Siebel Tools Clients” on page 334.](#)
- 5 Initialize the local database on development machines.
- 6 [“Checking for Inactivated EIM Table Columns” on page 371.](#)
- 7 Reset upgrade-specific parameters back to their defaults. See [“Resetting Database Server Configuration Parameters” on page 335.](#)
- 8 Run database statistics. For more information, see [“Generating RUNSTATS” on page 234.](#)

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

Postupgrade Tasks for Configuration

- 1 [“Reviewing Deleted Objects in the Repository Merge” on page 275.](#)
- 2 [“Reviewing Obsolete Objects in the Repository Merge” on page 275.](#)
- 3 [“Reviewing Customized Business Components” on page 276.](#)
- 4 [“Reapplying Custom Extensions and FK Columns on Obsolete Tables” on page 329.](#)

- 5 [“Resolving Business Component and Join Conflicts” on page 331.](#)
- 6 [“Deleting Unneeded Repository Files” on page 280.](#)
- 7 If you exported data from interface tables before the upgrade, review the database and import the data as desired.
- 8 [“Upgrading to RC2 or AES Encryption” on page 93.](#)

Configure for Globalization

- 1 Install language packs for new languages. See the *Siebel Installation Guide* for the operating system you are using.
- 2 [“Upgrading to the Symbolic String Model” on page 335.](#)
- 3 [“Setting Up Your Environment to Support Global Time Zone” on page 342.](#)

Deploy Workflows

To perform these tasks, see *Siebel Business Process Designer Administration Guide*.

- 1 Activate and deploy workflows. See *Siebel Business Process Designer Administration Guide*.
- 2 [“Upgrading Seeded Workflows” on page 368.](#)
- 3 [“Upgrading Inbound Workflows” on page 368.](#)

Verify Application Integration

- 1 Verify that EAI and EIM integrations are set up correctly. For information on using EAI and EIM, see *Overview: Siebel Enterprise Application Integration* and *Siebel Enterprise Integration Manager Administration Guide*.
- 2 [“Updating Enterprise Application Integration \(EAI\) After Upgrade” on page 343.](#)

Postupgrade Tasks for Application Administration

- 1 Review the results of the Person and Organization merge. Make configuration changes as required.
- 2 [“Generating Reporting Relationships After Upgrade” on page 340.](#)

Perform System Tests

- 1 Perform the tasks in [Chapter 20, “Postupgrade Tasks for Applications”](#) as needed to upgrade test data in the development environment.
- 2 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
- 3 If you revise repository objects or schema definitions, regenerate the schema.dll and custrep.dat files. See [“Regenerating the Repository Definition Files” on page 278](#).

Process of Upgrading a Production Environment

Upgrades: All upgrades.

Environments: Production environment only.

This process is part of a road map. See [“Road Map for Performing the Upgrade” on page 43](#).

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

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

Upgrade the Servers

Verify you have identified all the maintenance releases, Fix Packs, and quick-fix patches required for the upgrade. These requirements are located on Siebel SupportWeb under Product Documentation > Maintenance Release Guides.

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

- 1 Install Siebel Tools on development workstations. Keep at least one copy of the previous version of Siebel Tools. You will need it to perform repository operations before the repository merge.
- 2 Install the Siebel Gateway Name Server, Siebel Servers, and Siebel Web Server Extension (SWSE). In many cases, you do not have to uninstall the Siebel Servers or SWSE before installing the new versions.
- 3 Install the Siebel Database Server files on the Siebel Server you will use to perform the upgrade.
- 4 Install language packs for your currently deployed languages and any new languages.

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

Upgrade Third-Party Software

- 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 RDBMS

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

Preupgrade Tasks for the Siebel Database Upgrade

These steps apply to all database types.

- 1 Review Siebel Technical Notes and Siebel Alerts on upgrading to the current release. Make revisions to the upgrade process as required. See [“Important Upgrade Planning Resources” on page 23](#).
- 2 Review Technical Note 521. It contains important changes to database records and repository objects that must be made during the upgrade. See [“Important Upgrade Planning Resources” on page 23](#).
- 3 Review guidelines for configuring the RDBMS. See the *Siebel Installation Guide* for the operating system you are using.
- 4 Verify all developers have checked in their projects.
- 5 Verify that the Workflow Monitor and Workflow action agents have processed all pending requests.
- 6 Stop the Siebel Server and the Siebel Gateway Name Server.
- 7 Verify there are no open database connections.
- 8 [“Preparing Tables and Views for Upgrade” on page 101](#).
- 9 [“Preparing Custom Indexes for Upgrade” on page 102](#).
- 10 [“Exporting Interface Table Data” on page 102](#).
- 11 [“Preserving Dock Objects and Visibility Rules” on page 103](#).
- 12 [“Verifying Database Server Configuration” on page 103](#).
- 13 [“Securing AIX Memory Allocation Segment Space” on page 104](#).
- 14 [“Creating Storage Groups” on page 104](#).
- 15 [“Preparing the Storage Layout of the Schema” on page 104](#).
- 16 [“Reviewing Tablespace Size” on page 115](#).
- 17 [“Backing Up the Database” on page 118](#).
- 18 [“Adding a Siebel User” on page 118](#).

Preupgrade Tasks for DB2 UDB for z/OS

- 1 "Prerequisites for a Development Upgrade" on page 143.
- 2 "Checking In Development Repository Projects" on page 145.
- 3 "Saving Interface Table Data" on page 145.
- 4 "Upgrading Your DB2 Software" on page 146.
- 5 "Preparing EIM Tables for Upgrade" on page 146.
- 6 "Preparing for Intersection Table Maintenance and Deduplication" on page 148.
- 7 "Determining Which Template File was Used During an Extract or Merge" on page 149.
- 8 "Disabling Customized Triggers" on page 150.
- 9 "Dropping Customized Views" on page 150.
- 10 "Preparing the Development Environment Repositories for Upgrade" on page 150.

Preupgrade Tasks for Application Data

- 1 "Preparing Workflow Processes for Upgrade" on page 131.
- 2 "Preparing Mobile and Dedicated User Data for the Database Upgrade" on page 132.

The remaining tasks are optional depending on currently installed Siebel products and whether the data in the development environment needs to be preserved or upgraded. Review and perform these tasks as necessary.

- 3 "Setting Up Campaign Status Values for Siebel Marketing Upgrade" on page 133.
- 4 "Preparing Address Data for Upgrade" on page 135.
- 5 "Preparing Products for Upgrade" on page 136.
- 6 "Preserving Marketing Segment Descriptions" on page 136.
- 7 "Preparing Siebel eChannel Data for Upgrade" on page 137.
- 8 "Preparing Forecasting Data for Upgrade" on page 138.
- 9 "Preparing Products and Quotes for Upgrade" on page 138.
- 10 "Verifying Household Data Integrity" on page 139.
- 11 "Preparing Financial Services Application Tables for Upgrade" on page 141.
- 12 "Preparing Siebel Configurator Data for Upgrade" on page 141.

Prepare the Siebel Database for Upgrade

- 1 Verify you have a current backup of the production environment database.
- 2 On the Siebel Server you used to upgrade the development environment database, create an ODBC to connect to the production environment database.
- 3 Disconnect the Siebel Server from the development environment database.

- 4 Connect the Siebel Server to the production environment database.
- 5 Navigate to *DBSRVR_ROOT\common* (UNIX: *DBSRVR_ROOT/common*) and verify that the file *sql gen. usg* exists.

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

CAUTION: If this file does not exist, do *not* run the Database Server Configuration utility. It will overwrite the SQL files used to upgrade your development environment database. Contact Siebel Technical Support for guidance on proceeding with the upgrade.

Upgrade the Siebel Database Schema (upgrep)

- 1 Run the Database Server Configuration utility:
 - “Preparing to Run the Database Server Configuration Utility” on page 160.
 - “Running the Database Server Configuration Utility Under Windows” on page 163.
 - “Running the Database Server Configuration Utility Under UNIX” on page 164.

When running the Database Server Configuration Utility, choose the following settings:

- Upgrade Options:** Upgrade Siebel Database Schema (upgrep + upgphys)
 - Environment Type:** Production
- a Enter the information for the production environment instead of the development environment.
 - b When prompted whether you want to run the Siebel Upgrade Wizard, answer No and exit.

This updates the master UCF file with the production environment configuration. When you run the Siebel Upgrade Wizard, the SQL commands will be executed on the production environment database.
- 2 Upgrade the production environment database. See “Running the Siebel Upgrade Wizard” on page 165.

The Siebel Upgrade Wizard uses the SQL commands generated for the development environment to upgrade the production environment.
 - 3 “Reviewing Upgrade Log Files for Errors” on page 241.
 - 4 If the upgrade contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Server Configuration utility.
 - 5 “Manually Archiving Upgrade Log Files” on page 243.

Upgrade the Siebel Database Schema (upgphys)

- 1 Run the Database Server Configuration utility:

- [“Preparing to Run the Database Server Configuration Utility” on page 160.](#)
- [“Running the Database Server Configuration Utility Under Windows” on page 163.](#)
- [“Running the Database Server Configuration Utility Under UNIX” on page 164.](#)

When running the Database Server Configuration Utility, choose the following settings:

- Upgrade Options:** Upgrade Siebel Database Schema (upgrep + upgphys)
 - Environment Type:** Production
- 2 [“Reviewing Upgrade Log Files for Errors” on page 241.](#)
 - 3 If the upgrade contains unacceptable errors, do the following:
 - a Restore the backup of the database.
 - b Correct the errors.
 - c Rerun the Database Server Configuration utility.
 - 4 [“Manually Archiving Upgrade Log Files” on page 243.](#)
 - 5 [“Installing New License Keys After Upgrade” on page 176.](#)
 - 6 Back up the upgraded database.

Postupgrade Tasks for Environment Setup

- 1 [“Updating File System Attachments” on page 374.](#)
- 2 Generate a Siebel Remote database template file. See *Siebel Remote and Replication Manager Administration Guide* and [“Upgrading Siebel Mobile and Dedicated Web Clients” on page 337.](#)
- 3 Generate database extracts for all Regional Servers in the deployment. See [“Upgrading Regional Servers” on page 337.](#)
- 4 Reset upgrade-specific parameters back to their defaults. See [“Resetting Database Server Configuration Parameters” on page 335.](#)
- 5 [“Checking for Inactivated EIM Table Columns” on page 371.](#)
- 6 Run database statistics. For more information, see [“Generating RUNSTATS” on page 234.](#)

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

Postupgrade Tasks for Configuration

- 1 If you exported data from interface tables before the upgrade, review the database and import the data as desired.
- 2 [“Upgrading to RC2 or AES Encryption” on page 93.](#)

Configure for Globalization

- 1 [“Upgrading to the Symbolic String Model” on page 335.](#)

- 2 [“Setting Up Your Environment to Support Global Time Zone” on page 342.](#)

Deploy Workflows

To perform these tasks, see *Siebel Business Process Designer Administration Guide*.

- 1 Activate and deploy workflows. See *Siebel Business Process Designer Administration Guide*.
- 2 [“Upgrading Seeded Workflows” on page 368.](#)
- 3 [“Upgrading Inbound Workflows” on page 368.](#)

Verify Application Integration

- 1 Verify that EAI and EIM integrations are set up correctly. For information on using EAI and EIM, see *Overview: Siebel Enterprise Application Integration* and *Siebel Enterprise Integration Manager Administration Guide*.
- 2 [“Updating Enterprise Application Integration \(EAI\) After Upgrade” on page 343.](#)

Postupgrade Tasks for Application Administration

- 1 Review the results of the Person and Organization merge. Make configuration changes as required.
- 2 [“Generating Reporting Relationships After Upgrade” on page 340.](#)
- 3 [“Validating Dock Objects and Rule Definitions” on page 371.](#)
- 4 Perform the tasks in [Chapter 20, “Postupgrade Tasks for Applications”](#) to prepare for system testing.

Perform System Tests

- 1 Thoroughly test all applications.

Deploy to Users

- 1 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 eBusiness Applications*.
- 2 Use the Siebel Application Deployment Manager to migrate administrative data such as LOVs and responsibilities from development to production. See *Going Live with Siebel eBusiness Applications*.
- 3 Use Siebel Packager to create language packs for remote installation. See *Going Live with Siebel eBusiness Applications*.
- 4 Use Siebel Anywhere to create installation kits for deployment. See *Siebel Anywhere Administration Guide*.

- 5 Generate a Siebel Remote database template. See *Siebel Remote and Replication Manager Administration Guide*.
- 6 Set up database extraction for Siebel Mobile Web Clients. See *Siebel Remote and Replication Manager Administration Guide*.

5

Database and UI Upgrade Planning

This chapter contains the following topics:

- “Best Practices for Doing Your Upgrade” on page 83
- “Planning Your Upgrade” on page 85
- “About Database Growth Estimates” on page 91
- “About Upgrading Your RDBMS” on page 92
- “About Multilingual Deployments” on page 92
- “Upgrading to RC2 or AES Encryption” on page 93
- “New Upgrade Features” on page 93
- “About User Interface Changes” on page 93
- “About Database Sort Order” on page 94
- “About Access Control Upgrade” on page 95
- “About Party Model Upgrade” on page 96
- “About a New Column Added to S_SRC_PAYMENT” on page 98
- “About Database Clustered Indexes” on page 99
- “About HTML Attachment Migration to Base Tables” on page 99

Best Practices for Doing Your Upgrade

Upgrades: All upgrades.

Environments: All environments.

This section provides an overview of best practices for planning upgrade resources, estimating the upgrade time line, and managing the data migration process.

Use the following steps to help plan your upgrade.

- 1 Evaluate the complexity of the upgrade.** Determine the complexity of the upgrade effort based on Siebel modules implemented, number of integration points, number of interfaces, total number of scripts, and number of user interface scripts.
- 2 Assess the current Siebel environment and evaluate the existing implementation.** Perform a detailed assessment of the current Siebel environment to determine how the implementation will be affected by the upgrade. Evaluate the current implementation in comparison with the architecture of the current release. The assessment will help you to identify areas where you can take advantage of new Siebel Systems functionality to meet business requirements.

- 3 Estimate the level of effort to upgrade.** Determine the metrics and cost associated with each aspect of the upgrade. Determine the effort required to upgrade based on the results of your complexity evaluation, current environment assessment, and new functionality review. This will help you to estimate resources, time line, and costs.
- 4 Establish the upgrade team.** Assemble a cross-functional upgrade team that understands Siebel architecture and performance best practices. Include IT professionals, executives, and users to ensure a broad base of experience in technical, business, and Siebel-specific skills.
- 5 Review interface migration tasks.** Determine the effort to migrate modified applets and views. This includes associating applets with Web template items and mapping them to Web template controls.
- 6 Plan for upgrade tuning.** Tuning your production upgrade scripts can significantly reduce downtime during the final stages of your upgrade. Examples of upgrade tuning include eliminating SQL statements that do not affect any data, executing long-running SQL statements in parallel, and executing table creation, table rebuilds, and index creation in parallel.
- 7 Identify data migration tasks.** After the upgrade, there may be data migration and repository configuration tasks that must be performed manually. These tasks frequently involve customizations made in prior releases.
- 8 Plan for end-user training.** Analyze the impact of change on the users, and develop a plan for end-user training and adoption.

The upgrade of your Siebel application requires several key things to be successful:

- A detailed understanding of customizations made to your current deployment
- Analysis and definition of the components within your enterprise
- Analysis of how to use new functionality provided by Siebel software
- Strict adherence to industry best practices and best practices identified in this guide

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

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

Upgrade Planning Best Practices

Here are important best practices to follow when planning an upgrade:

- Review *System Requirements and Supported Platforms* on Siebel SupportWeb, Release Notes, Maintenance Release Guide, and Siebel Alerts related to upgrades to verify your upgrade path. These documents are available on Siebel SupportWeb at <http://ebusiness.siebel.com/supportweb/>.
- Gather all relevant documentation that describes the current implementation, for example requirements documents, design documents, and architecture context diagrams.

- 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.x to Release 7.7. There are significant user interface differences between the two versions, and you must change existing functionality before moving on to customized functionality.
- Implement a change management program. For example, communicate rollout dates to users, schedule training, allow adequate time for users to adjust to the enhancements, and provide a process for end users to provide feedback to the project team.
- User adoption is critical to a successful upgrade. Provide access to a test environment that allows users to become familiar with the new version of the application, and provide end-user training on the upgraded application.

Database Planning Best Practices

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

- Analyze the impact of the upgrade on table customizations that you have made. Determine if preupgrade data migration is required. Determine what postupgrade schema changes are required. For a list of tables that are affected by the upgrade, see [Appendix A, "Schema Changes."](#)
- Consider database layout in your planning. Plan to tune the database and Siebel Database Server for the upgrade, because settings and parameters for upgrade differ from those required for OLTP.
- If you are migrating multiple languages from a prior version, plan extra time (one to two weeks) for the repository merge process. The expected merge time may increase with the number of languages in the repository. You also might need to plan for additional installation-related tasks.
- Consider increasing the size of your tablespaces before going live. Make sure that your custom tablespaces are large enough for upgraded tables.

Planning Your Upgrade

Upgrades: All upgrades.

Environments: All environments.

There are several ways that you can proceed with the Siebel 7.7 upgrade. Consider your schedule, requirements, and available resources as you make the following decisions for your Siebel 7.7 upgrade.

About Moving Tablespaces

If you want to move tables from one tablespace to another, you need to recreate the tables in the new tablespace and then drop the existing tablespace. You cannot change the bufferpool designation in the storage control file to move tables because the page size is associated with the tablespace. For example, if you are making changes to an existing tablespace that is using BP1 or a 4K bufferpool and these changes cause you to receive a warning from ddlimp that the table will now need to be in a 16K bufferpool, you cannot just change the bufferpool designation in the storage control file from BP1 to BP16K1. If you change the bufferpool designation, ddlimp will generate the following statement:

```
ALTER TABLESPACE <NAME> BUFFERPOOL PB16K1
```

This command will fail when executed because it is not permissible to alter the bufferpool associated with the tablespace unless the new bufferpool designation is the same size as the existing bufferpool designation.

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.

Release 7.7 offered a new database schema structure. For more information, see [“New Database Schema Structure in Release 7.7” on page 86](#).

New Database Schema Structure in Release 7.7

Siebel offered a new database schema structure in Release 7.7 that has the following characteristics:

- One table in each tablespace
- One tablespace in each database

This new approach is referred to as the *1:1:1* model. Customers may want to switch to this database schema structure from their current database schema structure because it is more efficient.

In previous releases, the database schema was built using approximately 20 databases, each of which contained multiple tablespaces. Each of these tablespaces (if non-partitioned) contained multiple tables. Release 7.7 contains thousands of databases. For example, an SIA installation has approximately 2700 databases. Each database has one tablespace and each tablespace 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 IBM document, *DB2 UDB for OS/390 and z/OS V7 Administration Guide*.

The adoption of the 1:1:1 model for Release 7.7 means that all upgrade teams must decide how much of this new model to deploy. Upgrade teams have the following options:

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

New 7.7 tables will be 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 tablespace 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 tablespace. 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 110](#).
- 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`.

For more information on using storage control file, see *Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390*.

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 7.7 without exhaustive performance testing.

Considering Code Page Support

Development databases require ASCII 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 *Release Notes* on Siebel SupportWeb for information about known restrictions. For guidelines about choosing the code page for your subsystem, see *Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390*.

Executions of 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.

If you plan to use a third-party job scheduler, you will find it useful to first generate the upgrade scripts using Siebel scheduling to gain an understanding of job dependencies, because the Siebel-provided job scheduler submits dependent jobs by their predecessors.

Capacity Planning

For recovery purposes, an in-place upgrade is not supported on the zSeries host. If you want to keep both source and target databases, you must plan for additional disk space.

Verify that you have enough space on the zSeries host to allow for the unloading of datasets. Typically, you need at least 200 cylinders of space.

Optimizing Performance Using a Job Stream Optimization (JSO) Utility

A job stream optimization utility is an optional feature that removes unload and load jobs for tables containing no data.

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 7.7 upgrade.

CAUTION: Read *System Requirements and Supported Platforms* on Siebel SupportWeb to verify the supported computer and operating system platforms and supported third-party programs for this release of your Siebel eBusiness applications.

Obtaining Required IBM Utilities

Verify that the following IBM utilities are available for your upgrade to Siebel 7.7 for DB2 UDB for z/OS:

- **DSNTIAUL.** The upgrade uses DSNTIAUL to unload data delivered by Siebel.
Prepare and bind DSNTIAUL by following the procedures in your IBM installation documentation. If you made local modifications, you might need to prepare and bind a separate version.
- **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 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.

For information about IBM utilities, see your IBM documentation.

NOTE: You can use alternate third-party products that are preferred for your environment. Evaluate utilities by individual job.

Obtaining Required Security Privileges

For detailed information about security for DB2 UDB for z/OS installations and upgrades, see *Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390*.

In Siebel 7, access privileges to database resources such as tables, views, and triggers are granted to 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 repository 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

NOTE: 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.

Creating a Schedule for the Upgrade

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

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

- Run select 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 104](#).

- 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.

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 *Upgrade Guide* for your development environment upgrade. Then use this guide, *Upgrade Guide for DB2 UDB for z/OS and OS/390*, for your production environment upgrade.
- **Development environment database with DB2 UDB for z/OS.** During DB2 UDB 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.

The upgrade provides separate DDL steps and DML steps:

- Data definition language (DDL) files require database administrator (DBA) privileges. These files must be applied by a database administrator.
- Data manipulation language (DML) files do not require DBA privileges. These files are used to create seed data and repository data. These steps run automatically.

About Database Growth Estimates

Upgrades: All upgrades.

Environments: All environments.

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

For estimates of growth percentages for specific RDBMS platforms and releases, see [Table 9 on page 92](#).

Actual expected growth may vary widely from these estimates, depending on which Siebel application you are using (Siebel eBusiness application or Siebel Industry application), database configuration, row size of tables, and data content.

Table 9. IBM DB2 Database Growth Estimates

If Upgrading from Release...	4-KB Tablespace	16-KB Tablespace	32-KB Tablespace
7.5x	70%-80%	575%-600%	1100%-1200%
7.0x	35%-90%	25%-100%	30%-40%

About Upgrading Your RDBMS

Upgrades: All upgrades.

Environments: All environments.

If your currently installed RDBMS version is not supported in the new release, you must upgrade your RDBMS before performing the Siebel Database upgrade. For information on supported RDBMS versions, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

If your currently installed RDBMS version is supported in the new release, and you plan to upgrade your RDBMS, you can do so before or after upgrading your Siebel Database.

The following guidelines and requirements are for planning your RDBMS upgrade.

NOTE: When you upgrade the RDBMS, also be sure to upgrade your client database connectivity software. See *System Requirements and Supported Platforms* on Siebel SupportWeb.

About Multilingual Deployments

Upgrades: All upgrades.

Environments: All environments.

If you have multilingual deployments, you must install multilingual seed data (for example lists of views, responsibilities, or system preferences). To do this, you must add language packs to your database after you have installed the base language for your Siebel Database Server.

Install language packs before you perform a repository merge. (Add language packs to both the New Siebel Repository and the New Customer Repository so that they are synchronized.)

If you want to install new languages that are not included in your current deployment, wait until after the repository merge. This will improve merge performance.

See the *Siebel Installation Guide* for the operating system you are using for procedures on installing and configuring multiple Siebel language packs and importing a new language to your repository.

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

Upgrading to RC2 or AES Encryption

Upgrades: Releases 7.0.x, 7.5.x.

Environments: All environments.

Release 7.7 introduced support for the AES encryption method—the government standard for secure applications. Siebel eBusiness applications continue to support RC2 data encryption.

The Release 7.0.x default encryption method (called the standard encryptor) is no longer supported. Data that used the standard encryptor cannot be read by applications in the current release. You must upgrade your encryption method to RC2 or AES.

Use the Encryption Upgrade utility to convert unencrypted data and data that was encrypted using the standard encryptor to the RC2 or AES encryption method. You must run the Encryption Upgrade utility even if you are upgrading to stronger encryption, for example, from RC2 56-bit to RC2 128-bit encryption. Upgrading to RC2 128-bit or AES encryption requires the Siebel Strong Encryption Pack.

To upgrade your encryption method, see *Security Guide for Siebel eBusiness Applications*.

New Upgrade Features

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

Environments: All environments.

Release 7.7

The following upgrade features were introduced at Release 7.7.

- **Incorporate Custom Layouts (ICL).** This feature enables you to preserve the UI look and feel of your currently installed release during the repository merge, reducing the workload required to reconfigure customizations after the upgrade. ICL is available for Siebel 7.x upgrades only; however, it is included in Release 7.7.2 and later. For more information, see [“About the Incorporate Custom Layout \(ICL\) Upgrade Option” on page 248](#).
- **Production Upgrade Tuner.** This utility analyzes the SQL commands used to upgrade the development environment. It then helps you eliminate unneeded SQL commands and perform other operations intended to improve the performance of the production environment upgrade. For more information, see [“About Tuning the Production Upgrade Scripts” on page 377](#).

About User Interface Changes

Upgrades: Releases 7.0.x, 7.5.x.

Environments: All environments.

Several important changes to the user interface were introduced in Siebel 7.7:

- **New navigation scheme.** The key revisions are as follows:
 - The Show menu is replaced by a link bar. When you click on a screen tab, the link bar displays below the screen tab and shows the views or view list for the screen.
 - A visibility filter menu, located in the parent list applet, provides alternative methods for filtering the data displayed in the applet.
 - To display view tabs, users click on a link in the parent list applet.
- **Revised UI configuration.** In prior releases, placement of views and controls were based on repository configuration and logic applied at run time. In Release 7.7, placement of views and controls is declarative in the repository. No run-time logic is applied.
- **Expanded use of grid-layout applets.** To improve usability, most employee-specific form applets are converted to grid-based layout.
- **MVG shuttle applets.** MVG shuttle applets were introduced in Release 7.5x. In Release 7.7, MVGs are shuttle-enabled by default.

Because of these enhancements, you should plan a careful evaluation of the user interface after upgrade. The upgrade logging utility will list any user interface-related problems encountered during upgrade.

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

About Database Sort Order

Upgrades: All upgrades.

Environments: All environments.

Sort order (also called collation sequence) is specified during the initial installation of a database and defines the way in which the database sorts character data. Sort order support depends on both the code page of the database and whether it is used in a development or a production environment.

- **Development environments.** For development databases, you must use a binary sort order due to the functional limitations of databases that use a nonbinary sort order.
- **Production environments.** For production databases, it is strongly recommended that you use binary sort order to prevent possible performance degradation.

The settings for binary sort order are unique for each database platform. See *System Requirements and Supported Platforms* on Siebel SupportWeb to determine supported settings for your database platform.

Sort Order Considerations for Siebel Databases

If your deployment requires that you use a nonbinary sort order (for example, if your local language does not use binary sort order), you must consider several functional limitations that particularly affect development environment upgrades. If these limitations are unacceptable, consider re-creating your database to use binary sort order. However, be aware of these considerations:

- You cannot use Siebel Tools to generate a Siebel Repository file (SRF) on a database that uses a nonbinary sort order.
- You cannot perform a repository merge on a database that uses a nonbinary sort order.
- Databases that use nonbinary sort order might perform slower than databases that use binary sort order.

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

About Access Control Upgrade

Upgrades: Releases 7.0.x.

Environments: All environments.

Access control was significantly revised in Release 7.5. Access control refers to all mechanisms that control visibility of screens, views, and data within Siebel eBusiness applications. Access control includes, but is not limited to positions, responsibilities, organizations, and access groups.

To implement access control within your Siebel eBusiness applications, your Siebel administrator creates relationships between people and resources (a more general term for data that includes views and functionality). These relationships or policies are authorizations. Both people and resources can be grouped and placed in hierarchies to simplify administration.

External users, such as customers and channel partners, can be assigned varying access levels that control visibility of data and application functionality. When planning access policies, consider the following:

- The complexity of access control policies (one data item or group of data items can be accessed by one or many users or groups, but not by all).
- The amount of content that is distributed by the Siebel eBusiness applications, including Master data (data that is static and referential, such as Products) and Customer data (data that is created and managed by users of applications, such as Opportunities).
- The number of users and entities that access the data. Also consider the complexity of relationships between users (partners, competitors, browsers, customers).

For more information on access control, see *Security Guide for Siebel eBusiness Applications*.

Person, Household and Service Request Visibility

Beginning with Release 7.5, Person, Household, and Service Request can be made visible to multiple organizations, also called Business Units. Release 7.5 introduced several new tables to support this:

- S_CONTACT_BU

- S_ORG_GROUP_BU

- S_SRV_REQ_BU

The upgrade to Release 7.7 populates the S_CONTACT_BU, S_ORG_GROUP_BU, and S_SRV_REQ_BU tables with one record for each record in the S_CONTACT, S_ORG_GROUP, and S_SRV_REQ tables. After the upgrade, Contacts, Households, and Service Requests continue to be visible from the Business Unit they belonged to before the upgrade.

Access Group and Userlist Attributes

In Release 7.5, two new Siebel Extension tables were added to the S_PARTY, S_PARTY_GROUP and S_USERLIST tables to hold Access Group and User List attributes, respectively.

The upgrade to Release 7.7 adds records to the S_PARTY_GROUP and S_USERLIST tables for existing S_PARTY Access Group and User List records.

To support Multi-Org visibility, the upgrade also adds corresponding intersection table records to the S_PARTY_GRP_BU and S_USERLIST_BU tables.

Technical Note 312 provides guidance and best practices for implementing access control. This Technical Note includes background information about the Access Group access control mechanism implemented in Siebel 7, discusses migration considerations, and outlines steps for deploying Access Group access for Siebel eBusiness applications.

For detailed information about access control, see *Security Guide for Siebel eBusiness Applications*.

About Party Model Upgrade

Upgrades: Release 6.x only.

Environments: All environments.

Release 7.x introduces a party table (S_PARTY), in which all persons and organizational units are held. Accounts, Organizations, Internal Divisions, Contacts, Employees, Positions, and Households are all considered parties and can be referenced from this table.

Most of the tables that formerly contained this data still exist and are still used, but they are now extension tables to the S_PARTY base table. Data is loaded into the business components through an implicit join.

Additionally, Release 7.x uses a single-person table and a single-organization unit table. For example, Employees and Contacts are now combined in the same table (S_CONTACT). Similarly, internal and external Organization Units are now combined in the same table (S_ORG_EXT).

The S_PARTY table is the primary table in the Party or Single-Person model and is the base table for all Party business components.

Several extension tables support the Party Model:

- S_USER stores Siebel User information.

- S_EMP_PER stores attributes for Brand-Owner Employees and Partner Users who are considered agents of the Brand-Owner.

- S_BU stores Organization information.
- S_CONTACT
- S_ORG_EXT
- S_POSTN

Each non-person party directly or indirectly has person members, such as employees or contacts.

The Party model makes several tables obsolete:

- S_EMPLOYEE. Its functionality is merged into S_CONTACT.
- S_ORG_INT. Its functionality is merged into S_ORG_EXT.
- S_EMP_POSTN has been replaced by S_PARTY_PER.
- S_EMPLOYEE_ATT
- S_ORG_INT_ATT
- S_POSTN_RPT_REL

Figure 3 depicts the Party changes to the data model that occur during upgrades from Release 6.x to Release 7.x.

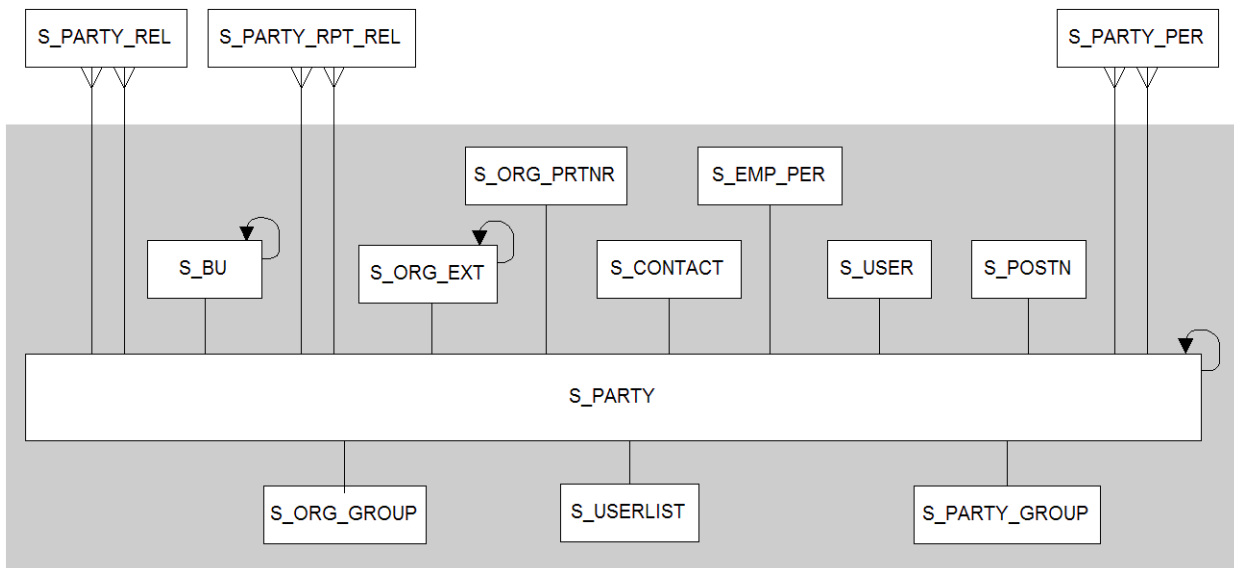


Figure 3. Party Model

How the Party Model Is Implemented During Upgrade

When you upgrade to Release 7.x, the upgrade process implements the Party model as follows:

Data Migration

- Migrates data from S_EMPLOYEE to S_CONTACT, S_USER, S_EMP_PER for standard Siebel columns
- Migrates data from S_ORG_INT to S_ORG_EXT, S_BU for standard Siebel columns
- Creates S_PARTY records for each previous contact, position, employee, account, division

Business Component Definitions

- Updates business component definitions to reference S_PARTY as the Primary Table (for example Employee, Contact, Position, and Account business components)
- Changes standard and custom joins on S_EMPLOYEE to S_CONTACT, S_USER, S_EMP_PER
- Changes standard and custom joins on S_ORG_INT to S_ORG_EXT
- Sets implicit joins for custom fields created on business components that have been retargeted to S_PARTY. For example, if a custom field, Alternate Phone, existed on the Contact business component, the upgrade would initiate the following actions:
 - Retargets Contact business component to S_PARTY
 - Defines join to S_CONTACT from S_PARTY on the Contact business component
 - Sets implicit join for the Alternate Phone field

Additional Planning Considerations

With Access Control comes two new visibility mode types, catalog visibility and group visibility. For certain standard Siebel eBusiness application configurations, the view mode and picklist mode have catalog visibility and only show data if the catalogs, categories, and access groups are defined.

During upgrade, you must decide whether you will move to the Access Control model, which requires careful planning and preparation of access groups, catalogs, and categories.

If you want to continue using the Multi-Org model, you must modify changes to the default view and picklist configuration as part of the upgrade.

For more details on the data model as it relates to the Party model, see *Siebel Data Model Reference*. For more information on Access Control and Security, see *Security Guide for Siebel eBusiness Applications*.

About a New Column Added to S_SRC_PAYMENT

Upgrades: Release 6.x only.

Environments: All environments.

The S_SRC_PAYMENT table has a new required column, TYPE_CD. The default value for this column is Payment. This value 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.

If your organization has used the S_SRC_PAYMENT table to store other types of payments (particularly, payments from your customers to your organization), you must reset the parameter in this column to fit your data and run Siebel Systems' Enterprise Integration Management (EIM) product. This updates the table and enables you to review your results before migrating to your production environment.

The valid values for this column as shipped with Siebel eBusiness applications are:

- Payment
- Receipt
- Write-Off
- Adjustment

For information on how to update the Lists of Values, see *Configuring Siebel eBusiness Applications*.

About Database Clustered Indexes

Upgrades: Release 6.x 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.)

About HTML Attachment Migration to Base Tables

Upgrades: Releases 7.0.x, 7.5.x.

Environments: All environments.

The upgrade migrates data from the S_WEB_CNTNT table to the S_CB_CNTNT_SET, S_CB_ASSET, and other Content Base tables.

HTML Attachments, introduced in Siebel 7, are upgraded into Content Base tables. The Content Base table system allows content to be stored in the database or in the Siebel file system. The system stores several types of content including plain text, markup languages (for example SGML and XML), and image files (for example GIF and JPEG). The system breaks up the content when storing it and reassembles the content before rendering it.

6

Basic Database Preparations

This chapter contains the following topics:

- “Preparing Tables and Views for Upgrade” on page 101
- “Preparing Custom Indexes for Upgrade” on page 102
- “Exporting Interface Table Data” on page 102
- “Preserving Dock Objects and Visibility Rules” on page 103
- “Verifying Database Server Configuration” on page 103
- “Securing AIX Memory Allocation Segment Space” on page 104
- “Creating Storage Groups” on page 104
- “Preparing the Storage Layout of the Schema” on page 104
- “Reviewing Tablespace Size” on page 115
- “Backing Up the Database” on page 118
- “Adding a Siebel User” on page 118

Preparing Tables and Views for Upgrade

Upgrades: All upgrades.

Environments: All environments.

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

To prepare tables and views for upgrade

- 1** Drop temporary tables and non-Siebel tables.
If the upgrade process detects a column with a datatype not acceptable to Siebel tables, the upgrade will fail.
- 2** Disable customized triggers.
You must re-create them after the upgrade.
- 3** Drop defined database views on Siebel tables.
You must re-create them after the upgrade.
- 4** Export interface table data that you want to preserve.
Interface tables are dropped and then re-created during upgrade. You can import the data after the upgrade.

Preparing Custom Indexes for Upgrade

Upgrades: All upgrades.

Environments: All environments.

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

Consider the following guidelines when preparing custom indexes for upgrade:

- **Custom indexes against extension columns on obsolete tables.** If you have created custom indexes that use extension columns on obsolete tables, you must migrate the data to new extension columns before upgrading the Siebel Database. For assistance, contact Siebel Technical Support.
- **Custom indexes that were not defined through Siebel Tools.** Custom indexes created without using Siebel Tools are not included in the schema definition in the Siebel Repository. These indexes are dropped during the database upgrade. To preserve these indexes, add them to the Siebel Repository using Siebel Tools.
- **Custom indexes on interface tables.** Custom indexes on interface tables are not re-created during the upgrade. You must re-create them after the upgrade is complete.
- **Custom indexes on base tables.** The Release 7.x upgrade automatically drops and re-creates custom indexes on base tables.
- **Custom indexes may need to be changed to reflect schema changes.** Reevaluate custom indexes for applicability in the new release. They may no longer be needed due to schema changes in the new release.

For more information about custom indexes, see *Configuring Siebel eBusiness Applications*.

Dropping Custom Temporary Tables

Release 7.x does not support custom tables created at the database level. If you have temporary tables or non-Siebel tables, drop them before you perform the upgrade. If the upgrade process detects any column with a data type under the table that is not acceptable by Siebel, the upgrade fails.

Considerations for Clustered Indexes

If you created clustered indexes on base tables and Release 7.x 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.

Exporting Interface Table Data

Upgrades: All upgrades.

Environments: All environments.

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

During the upgrade process, your interface tables are dropped and then re-created. To retain data in your interface tables, use the appropriate tools for your RDBMS to export data before the upgrade and then import the data after you have completed the upgrade.

During the upgrade, all custom indexes on interface tables are dropped from both logical and physical schema.

Preserving Dock Objects and Visibility Rules

Upgrades: All upgrades.

Environments: All environments.

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

Changes to visibility rules and dock objects require the assistance of Technical Support or Siebel Expert Services.

Modified visibility rules are dropped during a development environment upgrade. Manually record any changes to dock object visibility rules, so you can evaluate whether you must reapply the changes after the upgrade is complete.

Dock objects and visibility rules created by using Docking Wizard are preserved unless they become invalid after the upgrade. Manually record any changes that you made through the Docking Wizard so that you can evaluate whether you need to reapply the changes after the upgrade is complete.

Verifying Database Server Configuration

Upgrades: All upgrades.

Environments: Development environment only.

Verify that your development Siebel Database Server configuration meets or exceeds Siebel requirements as described in *Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390*.

- Informational APAR pq83289 is required for loading multiple tables to the same database in parallel.
- The upgrade sets the TRACKMOD parameter the default value for tablespace objects: YES.
- The DSNZPARM CMTSTAT should be set to INACTIVE to prevent timeout errors from occurring. If you do not run INACTIVE, then set IDTHTOIN to IGNORE.

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 version 7 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 eBusiness applications to Release 7.7, expect database growth in the range of 30-50%. The amount by which your database grows may vary widely, depending on your database configuration, row size of tables, and data content.

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

Securing AIX Memory Allocation Segment Space

Upgrades: All upgrades.

Environments: All environments.

Platforms: UNIX only.

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

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

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

This will prevent a shortage of memory allocation segment space that might occur on the machine where both the Siebel Database Server and Siebel Server are installed.

After a successful upgrade, reset this parameter to the original value.

Creating Storage Groups

Upgrades: All upgrades.

Environments: Development environment only.

Before upgrading the Siebel Database Server, your DBA must create storage groups on the target database. The upgrade process assumes that the names for the storage groups in the target database are the same as those defined in the source database. For this reason, the names used to define the storage groups for the target database should be the same as those defined on the source database. If the names are not the same, your DBA must manually modify the schema.sql file so that the names match.

Preparing the Storage Layout of the Schema

Upgrades: All upgrades.

Environments: Development environment only.

NOTE: Preparation of the storage control file can be performed by your DBA at any time before the upgrade.

This process consists of four tasks:

- [“Modifying the Storage Control File” on page 106](#)
- [“Extracting the Storage Control File” on page 110](#)
- [“Validating the Storage Control File” on page 112](#)
- [“Reviewing the Storage Control File” on page 113](#)

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 database. Even if you are using a preconfigured storage layout, you must make sure that the layout is valid for your schema.

Release 7.7 includes a new schema structure, for more information see [“Planning Your Upgrade” on page 85](#).

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 Siebel Database Storage Configurator for further manipulation.

NOTE: For more information on the Siebel Database Storage Configurator, see *Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390*.

To prepare the storage layout, you must perform the following procedures:

- Review the storage control file and modify it (if necessary). See [“Reviewing the Storage Control File” on page 113](#).
- Extract the storage control file. See [“Extracting the Storage Control File” on page 110](#).
- Validate the storage control file. See [“Validating the Storage Control File” on page 112](#).

NOTE: You must validate the storage control file after you extract it and after you modify it. You must also perform this validation before you execute the schema.sql on the zSeries host.

To prepare the storage control file, use the Database Server Configuration Wizard to extract the storage layout of your source 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 three templates for the storage control file:

storage_np.ctl	Contains no partitioning scheme
storage_p.ctl	Contains partitioning scheme for a set of tables on an ASCII database
storage_p_e.ctl	Contains partitioning scheme for a set of tables on an EBCDIC database

For more information about templates for the storage control file, see *Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390*.

CAUTION: The storage layout for partitioned tables ships with a default code page of ASCII. If you are upgrading an EBCDIC database, you must use the EBCDIC storage control file storage_p_e.ctl.

Modifying the Storage Control File

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

Method 1. This method consists of running the Database Server Configuration Wizard, choosing the Generate DDL into a File installation option, and pointing the utility at one of the Siebel storage control file templates provided for this purpose. Then your DBA can edit the output DDL (without any action on the storage control file) on DB2 as desired, using native DB2 tools.

- 1 Generate DDL.** Perform a standard installation to generate DDL either to an output file for later transfer to DB2 or directly against the database for manipulation there, using native tools.
- 2 Configure Storage Space.** After generating DDL (schema.sql), execute the schema directly against the database.
- 3 Extract Storage Control File from the DB2 Catalog.** Access the extract utility Siebel provides through the Database Server Configuration Wizard. This utility allows you to extract information from the DB2 catalog.

NOTE: You can use this extract utility any time you want to create a new storage control file, based on the DB2 catalog.

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.

Method 3. This method consists of using a Siebel-provided tool called the Database Storage Configurator. This is a Siebel-provided Microsoft Excel macro (dbconf.xls) that is installed in the db2390 subdirectory of your dbrvr installation directory. This tool allows you to visualize and access storage control file templates.

CAUTION: Validate the storage control file again after you modify it. See “Validating the Storage Control File” on page 112.

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 eBusiness Applications on DB2 UDB for z/OS and OS/390*.

Preparing to Extract the Storage Control File

Use this topic to identify the information you must enter when running the Database Server Configuration utility to extract the storage control file. Collect this information and verify it before running the utility.

The information in your storage control file comes from the source database. 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 source 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.

- **Extract from Catalog and Merge with Template.** This method preserves your existing layout. This merges storage layout information from your source 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 a 7.x 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 `SI EBSRVR_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 as appropriate. If you have questions, contact Siebel Technical Support or Professional Services.

For Windows users, screen refers to a standard, application window. For UNIX users, the Upgrade Wizard displays in a shell. Screen refers to the pages of text that display in the shell.

NOTE: It is recommended that you enter all values in uppercase letters.

Table 10 lists the information that you must enter in the Database Server Configuration utility to extract the storage control file.

Table 10. Information Required for the Database Server Configuration Utility to Extract the Storage Control File

Screen Name	Required Information
Siebel Gateway Name Server Address	Windows only. The Siebel Gateway Name Server machine name and the Enterprise Server name.
Siebel Server Directory	The absolute path of the directory where the Siebel Server is installed. For UNIX, do <i>not</i> enter the string \$SIEBEL_ROOT—for example D: \sea_7x\si ebsrvr.
Siebel Database Server Directory	The absolute path of the directory where the Siebel Database Server is installed—for example D: \sea_7x\dbsrvr.
RDBMS Platform	Choose IBM DB2 UDB for z/OS.
Siebel Database Operation	Choose Run Database Utilities.
Database Configuration Options	Choose Extract Storage Control File to extract a storage control file.
Source Database ODBC Data Source Name	<p>Verify the ODBC name for connecting to the source Siebel Database for your upgrade. If it is not correct, enter the correct ODBC name.</p> <p>The ODBC data source <i>must</i> have the same name as the subsystem. When you set up the ODBC connection in DB2 Connect, use the actual subsystem name for the database alias.</p> <p>Windows: To find the name of your ODBC data source, navigate to Start > Settings > Control Panel > Administrative Tools > Data Source (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>
Source Database User Name	<p>Enter the source user name and password for the Siebel administrator of the source database for your upgrade.</p> <p>NOTE: The source database user name (user ID) needs to have authorization to set CURRENT SQLID.</p>
Source Siebel Schema Qualifier and Security Group ID.	<p>Source Schema Qualifier. Enter the eight-character identifier that designates the Siebel Schema for your source 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> <p>Source Security Group ID. Enter the user ID of the group to whom schema access is granted, for example, SSEROLE.</p>

Table 10. Information Required for the Database Server Configuration Utility to Extract the Storage Control File

Screen Name	Required Information
Extract Options	<p>Extract from Catalog. This option extracts the storage layout of your source database from the DB2 catalog as-is. The output is a new storage layout.</p> <p>Extract from Catalog and Merge with Template. This option preserves your existing layout. This option merges storage layout information from your source database 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>
Database Encoding	Indicate whether your DB2 subsystem is ASCII or EBCDIC.
Siebel Schema Layout	<p>Choose Siebel Schema without Partitioning if you want all tables only in segmented tablespaces.</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 tablespaces.</p>
Default Tablespace	Enter the name of the default tablespace.
Storage Group for Tablespaces	<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	<p>Indicate the values for the following parameters:</p> <p>4KB Buffer Pool Name. Enter the 4-KB buffer pool name for your tablespaces or accept the default name, BP1. This buffer pool should already be activated and have access to it granted by the DBA.</p> <p>16KB Buffer Pool Name. Enter the 16-KB buffer pool name for your tablespaces or accept the default name, BP16K1. This buffer pool should already be activated and have access to it granted by the DBA.</p>

Table 10. Information Required for the Database Server Configuration Utility to Extract the Storage Control File

Screen Name	Required Information
32KB Buffer Pool Name	<p>Indicate the values for the following parameters:</p> <p>32KB Buffer Pool Name. Enter the 32-KB buffer pool name for your tablespaces or accept the default name, BP32K1. This buffer pool should already be activated and have access to it granted by the DBA.</p> <p>Index Buffer Pool Name. Enter the buffer pool name for indexes or accept the default name, BP2. This buffer pool should already be activated and have access to it granted by the DBA.</p>
Database Name Prefix	<p>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>
Storage Control File	<p>Enter the directory path and name for the storage control file created by this process.</p> <ul style="list-style-type: none"> ■ Windows: Accept the default value displayed in the Storage Control File field (this is the <i>DBSRVR_ROOT\DB2390</i> directory, for example, D: \DBSRVR_ROOT\DB2390\storage.ctl) or use the Browse button to navigate to a different directory. ■ UNIX: Enter the path to your storage control file.
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> <p>CAUTION: If you are restarting or recovering a previous incomplete upgrade, do not change the Log Output Directory that you previously selected.</p>

Extracting the Storage Control File

This procedure extracts the storage control file.

To extract the storage control file

- 1 Run the Database Configuration Wizard using the procedure appropriate for your operating system.
 - “Running the Database Server Configuration Utility Under Windows” on page 163.
 - “Running the Database Server Configuration Utility Under UNIX” on page 164.

- 2 If you are using a UNIX operating system, go to [Step 3 on page 111](#).

If you are using a Windows operating system, on the runnow dialog box indicate whether you are ready to launch the Siebel Upgrade Wizard.

- To apply the configuration later, click Cancel.
- To apply the configuration now, click OK.

The Database Server Configuration Wizard validates that your repository has been renamed to Prior Customer Repository.

- If the repository name verification fails, you need to rename your repository to Prior Customer Repository before you proceed with the upgrade.
- If the repository name verification succeeds, the Configuration Parameter Review screen appears.

- 3 On the Configuration Parameter Review screen, review the configuration values you entered on the previous Configuration utility screens.

NOTE: Passwords are encrypted and do not appear in plain text either in the user interface or in the upgrade configuration files (UCF files). After a password is entered, it always appears in encrypted form. If you need to use another password, you must rerun the configuration utility.

If you need to change any values, do the following:

- Windows: Use the Previous and Next buttons to access the desired screens, then to return to this screen. When you are satisfied, click Finish to launch the Siebel Upgrade Wizard.
- UNIX: Enter N, and rerun the Database Server Configuration Wizard by entering `dbsrvr_config.ksh`. This allows you to reconfigure with different values. When you are satisfied, enter Y to launch the Siebel Upgrade Wizard (`srvrupgwi z`).

When the Siebel Upgrade Wizard appears, click OK to execute the Siebel Upgrade Wizard and create the storage control file in the directory that you specified in [Table 10 on page 108](#).

This process reads the database catalog and merges your prior custom database layout with one of the three 7.7 Siebel Database layout templates (located in the `DBSRVR_ROOT\db2390` directory). New objects take a layout from one of the three layout templates. Each layout template has its own storage control file.

By default, new tables are created as one-table-per-database to prevent concurrency and locking errors.

If you select a control file with partitioning, by default, the existing database catalog overrides the templates unless you specify table names in the override input file (`override.inp`). If your tablespace is not large enough to hold the merged schema, examine the `DDLIMP.log`. Look for the bufferpools for the offending tablespace, and increase as necessary. Bufferpools are likely to increase for some tables when you extract or validate the storage layout.

The addition of columns to certain tables may require these tables to be moved to a tablespace with a larger buffer pool. For more information, see [“Reviewing Tablespace Size” on page 115](#).

Validating the Storage Control File

You must validate the storage control file after you extract it or after you modify it. This process validates that the tables are the right length for the target schema.

- For development environment upgrades, this process validates the storage control file for the source database, so you must use the source tableowner.
- For production environment upgrades, this process validates the storage control file for the target database, so you must use the target tableowner.

The validation produces two output files in the `log` directory under `SI EBSRVR_ROOT`.

- `db_conf_i g_s i e b e l . l o g`
- `db_conf_i g_v a l i d a t i o n . l o g`

NOTE: When validating the extracted storage control file, the validation job calls for the `schema.ddl` file as an example. Make a copy of `ddl.ctl` and rename it `schema.ddl`, keeping the copy in the same location as `ddl.ctl`.

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.

NOTE: It is recommended that you enter all values in uppercase letters.

To validate the storage control file

- 1 Launch the Database Server Configuration Wizard as you did in [“Extracting the Storage Control File”](#) on page 110.

In the following steps, for Windows users, screen refers to a standard, application window. For UNIX users, the Siebel Upgrade Wizard displays in a shell. Screen refers to the pages of text that display in the shell.

- 2 Enter the information requested on each screen and click or press the appropriate button for your operating system to proceed.
 - Windows: Click Next.
 - UNIX: Press Enter.
- 3 On the Database Utility Selection screen, choose `Val i d a t e S t o r a g e C o n t r o l F i l e`.
- 4 On the Storage Control File Language screen, choose your Storage Control File language.
- 5 On the Configuration Parameter Review screen, review the configuration values you entered on the previous Configuration utility screens.

NOTE: Passwords are encrypted and do not appear in plain text either in the user interface or in the upgrade configuration files (UCF files). After a password is entered, it always appears in encrypted form. If you need to use another password, you must rerun the configuration utility.

If you need to change any values, do the following:

- Windows: Use the Previous and Next buttons to access the desired screens, then to return to this screen. When you are satisfied, click Finish to launch the Siebel Upgrade Wizard.
 - UNIX: Enter N, and rerun the Database Server Configuration Wizard by entering `dbsrvr_config.ksh`. This allows you to reconfigure with different values. When you are satisfied, enter Y to launch the Siebel Upgrade Wizard (`srvrupgwiz`).
- 6 To validate that `storage.ctl` works, connect to the catalog, and run `ddlmp`.
- NOTE:** If your tablespace is not large enough to hold the merged schema, examine the `DDLIMP.log`. Look for the bufferpools for the offending tablespace, and increase as necessary. Bufferpools are likely to increase for some tables when you extract or validate the storage layout.
- 7 Review the log file (`ddlmp.log`) that is generated in the log subdirectory under `SI EBSRVR_ROOT`. `SI EBSRVR_ROOT` refers to the directory that you specified when you set up the validation job.

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. 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 Storage Control File” on page 112.

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 = SI EBTS00

Stogroup = \$StogroupTables

IndexStogroup = \$StogroupIndexes

IndexBp = \$IndexBufferPool

Bufferpool = \$KBufferPool

Locksize = Page

SegSi ze = 32
LockMax = 0
Pri Qty = 7200
SecQty = 14400
PctFree = 17
FreePage = 0
Compress = No
Defi ne = No
Erase = No
CCSID = \$DbType

[Object 2]

Type = Database
Name = \$DbnamePrefi x0000
LockSi ze = Page

[Object 3]

Type = Tabl espace
Name = SI EBTS00
Database = \$DbnamePrefi x0000
Bufferpool = \$4KBufferPool
Stogroup = \$StogroupTabl es
LockSi ze = PAGE
LockMax = 0
SegSi ze = 32
Pri Qty = 7200
SecQty = 14400
PctFree = 17
FreePage = 0
Compress = No
Parti ti ons = 0
Defi ne = 0

Erase = 0

Reviewing Tablespace Size

Upgrades: All upgrades.

Environments: Development environment only.

Certain tables require larger tablespaces in Release 7.7 to accommodate additional columns. Review each table listed in [Table 11 on page 116](#) and verify that the page size is sufficient to accommodate new columns.

NOTE: It is recommended that you validate the storage control file particularly where you have schema changes.

Specify an entry in the storage control file for each obsolete table and extended table that you want to move to the 1:1:1 model. The 1:1:1 model was new in Release 7.7. For more information, see [“Planning Your Upgrade” on page 85](#).

NOTE: You must move S_EMPLOYEE to a 16-KB bufferpool before performing an upgrade because an additional column is added to this table by the upgrade. S_EMPLOYEE has no default tablespace name, because it is obsolete.

Drop all views (including custom views) on a table before you move it.

Table 11. Tables with Additional Columns

Table Name	Default Tablespace Name	New Page Size
S_ASSET	H0477000	16 KB
S_COLUMN	H0658000	16 KB
S_EMPLOYEE	H0997000	16 KB
S_CRSE_URL	H0738000	16 KB
S_DD_TRGT_GRP	H0875000	16 KB
S_I C_CALC_I T	H1158000	16 KB
S_PART_RPR	H1523000	16 KB
S_PERF_MGR_APPR	H1534000	16 KB
S_PROD_DEFECT	H1626000	16 KB
S_PROD_I NT	H1639000	16 KB
S_SCHMST_DBSCPT	H1808000	32 KB
S_SRC_PAYMENT	H1894000	16 KB
S_SRV_REQ1_FNX	F2689000	16 KB
S_SRV_REQ1_FNXM	F2690000	16 KB
S_TABLE	H1966000	16 KB
S_UPG_KI TWI Z	H2009000	16 KB

NOTE: Certain tables listed in Table 11 may not have existed in the schema during previous releases.

Table 12 lists the EIM tables that need to be evaluated for sufficient tablespace size.

Table 12. EIM Tables with Additional Columns

Table Name	Default Tablespace Name	New Page Size
EIM_ACCOUNT	H0004000	32 KB
EIM_ADDR_PER	H0019000	16 KB
EIM_AGREE_DTL	H0022000	16 KB
EIM_AGREEMENT1	H0021000	32 KB
EIM_ASSET_DTL	H0033000	16 KB
EIM_BASELN_DTL	H0041000	16 KB
EIM_CALL_LST	H0046000	32 KB

Table 12. EIM Tables with Additional Columns

Table Name	Default Tablespace Name	New Page Size
EIM_CONSUM_DTL	H0064000	16 KB
EIM_CONTACT	H0065000	32 KB
EIM_CONTACT1	H0066000	32 KB
EIM_COURSE	H0073000	32 KB
EIM_ENTLTMPL	H0121000	16 KB
EIM_FN_ASGNGRP	F2300000	16 KB
EIM_FN_PAYEE1	F2346000	16 KB
EIM_FN_PAYEE2	F2347000	16 KB
EIM_FN_REVN1	F2357000	32 KB
EIM_GROUP	H0147000	16 KB
EIM_I C_I NCENTV	H0153000	16 KB
EIM_MDF_DTL	H0181000	16 KB
EIM_ORDER	H0198000	32 KB
EIM_ORDER1	H0199000	16 KB
EIM_ORDER_I TEM	H0201000	32 KB
EIM_ORDER_SHI P	H0203000	16 KB
EIM_PAYMENT	H0215000	32 KB
EIM_PAYMNT_DTL	H0216000	16 KB
EIM_PDSHI P_DTL	H0219000	16 KB
EIM_PL_I TM_DTL	H0228000	16 KB
EIM_PRODI NVLOC	H0249000	16 KB
EIM_PRSP_CON	H0271000	32 KB
EIM_PT_RPR_DTL	H0276000	16 KB
EIM_QUO_I T_DTL	H0286000	16 KB
EIM_QUOTE1	H0281000	16 KB
EIM_QUOTE_I T	H0283000	32 KB
EIM_SRC1	H0301000	16 KB
EIM_SRC2	H0302000	16 KB
EIM_SRC_DBM	H0303000	16 KB
EIM_SRC_EVT	H0305000	32 KB
EIM_TARGET_DTL	H0317000	16 KB

Backing Up the Database

Upgrades: All upgrades.

Environments: Development environment only.

Perform a full backup of the database, using the utilities specific to your database platform. 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 (upgrep+upgphys)
- After the repository merge

Adding a Siebel User

Upgrades: All upgrades.

Environments: Development environment only.

Through Siebel Client application administration, add the Siebel user or users (with administrative privileges) who executes the Database Server Configuration Wizard. This is the person whose user ID is entered when prompted for Database User Name. See *Applications Administration Guide*.

7

Application Upgrade Planning

This chapter contains the following topics:

- “Upgrade Planning for Address Migration” on page 119
- “Upgrade Planning for Siebel Employee Relationship Management (ERM)” on page 120
- “Upgrade Planning for Siebel Marketing” on page 121
- “Upgrade Planning for Siebel Workflow Designer” on page 126
- “Upgrade Planning for Handheld Devices” on page 127
- “Upgrade Planning for Resonate Central Dispatch” on page 128
- “Upgrade Planning for String Translation” on page 128

Upgrade Planning for Address Migration

Upgrades:

- From Release 6.x Siebel Industrial applications (except Communications, Media, and Energy) that you are upgrading directly to Release 7.7 Siebel Industrial applications
- From Release 6.x and 7.0.x Siebel Financial Services applications that you are upgrading to Release 7.7 Siebel Industrial applications
- From Release 7.7 Siebel eBusiness applications that you are upgrading to Release 7.7 Siebel Industry applications
- From Release 6.2.1 Siebel Financial Services applications on IBM z/OS platforms that you are upgrading to Release 7.7.x Siebel Industry applications on IBM z/OS platforms

NOTE: This topic does not apply to Siebel eBusiness applications (HOR) that you are upgrading to a later release of Siebel eBusiness applications (HOR).

Release 7.7 changes the way address information is stored. You must perform several tasks to make sure address migration is handled correctly.

Previous Releases

In previous releases, address data was stored as follows:

- The relationship between person and address was $1:M$ and was stored in the table S_ADDR_PER.
- The relationship between account and address was $1:M$ and was stored in S_ADDR_ORG.
- Both tables included a column ADDR_NAME, which is a computed value based on other attributes in the address table.
- The user key for S_ADDR_PER included PER_ID and ADDR_NAME.

Release 7.7

In Release 7.7, the relationship between person and address and between account and address is M:M. Address information is stored in S_ADDR_PER. The relationship between addresses and contacts and between addresses and organization is stored in S_CON_ADDR.

The upgrade process revises storage of address data as follows:

- Inserts data into S_CON_ADDR from S_ADDR_PER and S_ADDR_ORG.
- Migrates data from S_ADDR_ORG to S_ADDR_PER. The table S_ADDR_ORG is obsolete.
- Sets S_ADDR_PER.PER_ID to null. It is no longer used as part of the user key.
- The table S_CON_ADDR becomes the intersection table for data stored in S_ADDR_PER and account or contact data stored in S_ORG_EXT or S_CONTACT respectively.
- Uniqueness of addresses in S_ADDR_PER is enforced only on ADDR_NAME.

How Address Data Is Preserved

Because PER_ID is no longer part of the user key for S_ADDR_PER, the ADDR_NAME must be unique for all records.

It is possible that records within or across S_ADDR_ORG and S_ADDR_PER could have the same ADDR_NAME. If this occurs, the ADDR_NAME for one of the records is preserved, and the upgrade process appends the ROW_ID to ADDR_NAME for the others. This prevents records from being deleted and preserves all records from both tables.

How to Manage Address Migration

You must perform three tasks to migrate address data:

- Before upgrading the database, you must run a script to identify records that have the same ROW_ID between S_ADDR_PER and S_ADDR_ORG. You must eliminate duplicate row IDs.
- You must evaluate whether to modify upgrade scripts to migrate address data in custom extension columns in S_ADDR_PER and S_ADDR_ORG. During the database upprep, you do this after running the Database Server Configuration utility but before running the Siebel Upgrade Wizard.
- After the upgrade is complete, review the records in S_ADDR_PER and eliminate duplicate and obsolete records.

To manage address migration, follow the steps in [“Preparing Address Data for Upgrade” on page 135](#). Each of the address migration tasks is included as a step in this process. Each step refers you to a procedure for performing the task.

Upgrade Planning for Siebel Employee Relationship Management (ERM)

Upgrades: Releases 7.0.x, 7.5.x.

Release 7.7 of ERM introduced important changes to workflows, Siebel Training, Microsite management, and Group News.

Workflows

Release 7.5.3 included approval business process workflows in the Sample database. If you activated these workflows, or created workflows containing approval steps, you must manually upgrade these workflows.

Siebel Training

In previous releases, the product catalogs containing Siebel Training courses were of type Buying. Release 7.7 provides a new catalog type called Training. If you have catalogs that contain both Siebel Training products and other kinds of products, revise these catalogs so that they contain only Siebel Training products. This prevents nontraining products from being moved to the training catalog during upgrade.

Microsite and Group News

If you have created customized microsite or Group News page sections, you must revise the associated business component and applet definition.

Upgrade Planning for Siebel Marketing

Upgrades: Releases 7.0.x, 7.5.x.

Release 7.7 introduced three architectural changes to Siebel Marketing:

- **Simplified hierarchy.** Program occurrences and campaign occurrences have been removed. This simplifies the Siebel Marketing hierarchy.
- **Merged Business Objects.** In previous releases, there were two business objects that supported similar campaign functions, DBM Campaign and Campaign. Release 7.7 merged all campaign administration functions into the Campaign business object.
- **Marketing Server migrated to Siebel Analytics.** The Marketing Server is obsolete. Its functions have been migrated to Siebel Analytics and are implemented on the Siebel Analytics Server.

Simplified Hierarchy

The Program Plan > Campaign Plan > Campaign Occurrences hierarchy has been simplified. The new hierarchy is Programs > Campaigns > Waves.

During the upgrade, objects that were related to the campaign occurrence are re-parented to the surviving campaign. Child objects are re-parented from a campaign occurrence to a campaign, as shown in Figure 4.

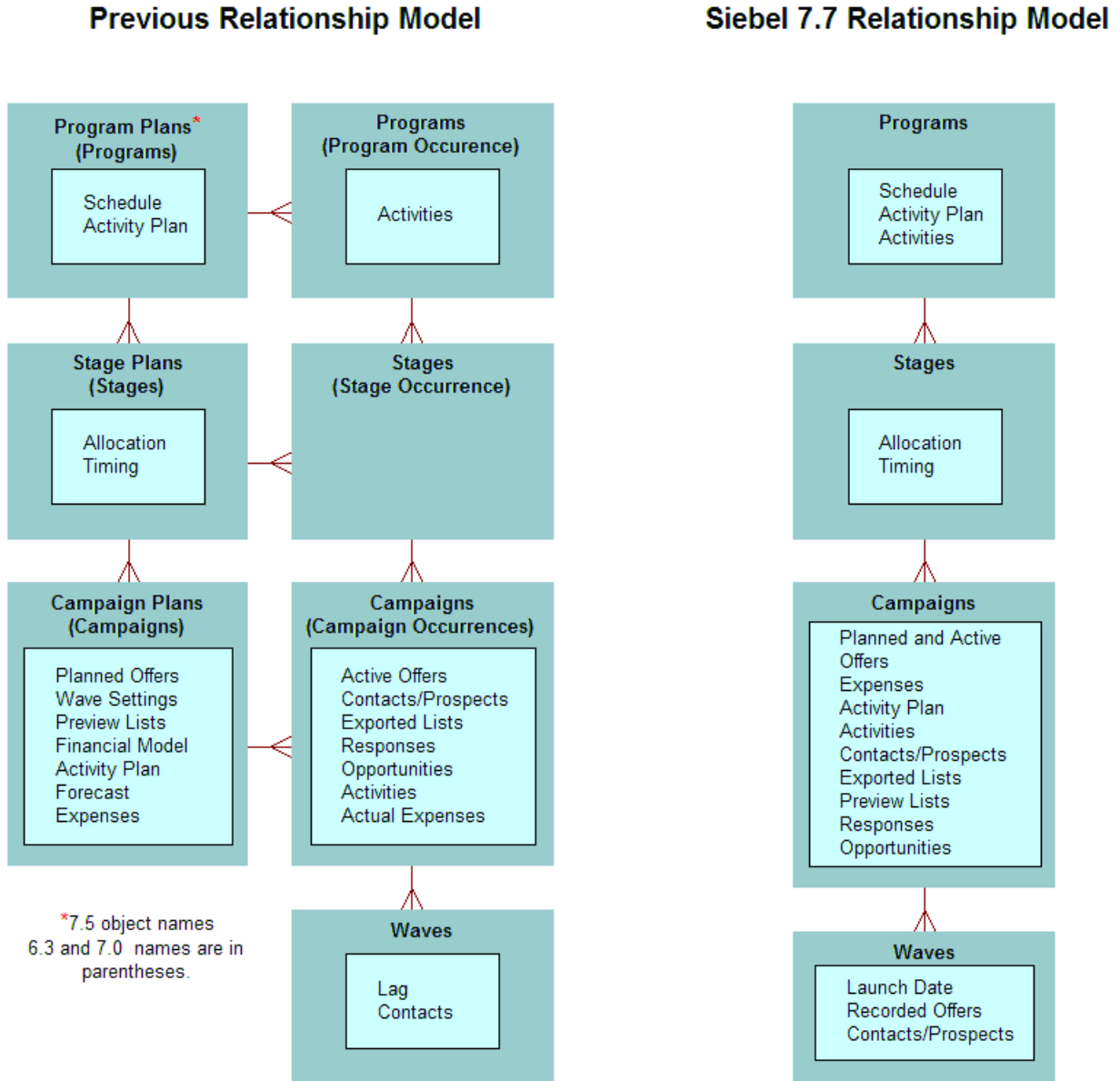


Figure 4. Relationship Model for Siebel Marketing

Merged Campaign Business Objects

In previous releases, campaign functions were handled by two business objects—DBM Campaign and Campaign. These objects shared similar functions, but many related elements such as views and child objects were not shared. The upgrade merges all campaign functions and views into a single business object—the Campaign business object.

During the upgrade, the following changes are performed to merge campaign plans and campaigns:

- Campaign Plans from the previous release are converted into Campaigns.
- The campaign occurrences that were children of the Campaign Plan are made obsolete.
- A new Load record is created for each previous occurrence.

Any child objects that were related to the obsolete campaign occurrence are re-parented to the surviving campaign.

Marketing Server Migrated to Siebel Analytics

The upgrade obsoletes the Data Dictionary Manager and Marketing Server components in the Siebel Enterprise Server. In previous releases, the Marketing Server component group performed segmentation and list export. Segmentation and list export are handled by subsystems in the Siebel Analytics Web server.

Due to the obsolete server components, a set of business components and other objects are also made obsolete. These objects remain in the repository but are no longer used by the application.

The upgrade does not obsolete the List Import Service Manager, Marketing AOM (SObjMgr), eMarketing AOM (eMarketObjMgr), or the eEvents AOM (eEventsObjMgr). They are actively used in the application.

[Appendix B, “Siebel Marketing Upgrade Reference”](#) lists the following obsolete items:

- Server components
- Business objects and business components
- Screens and views
- Data that is obsolete

About Siebel Marketing Customizations

Release 7.7 introduced the Incorporate Custom Layouts (ICL) feature. When you select this feature in the Upgrade Wizard, customized view and applet layouts are preserved, consistent with new features and schema changes.

Because of the large number of user interface changes in Siebel Marketing, the views and applets for Siebel Marketing have been excluded from ICL support. (The Upgrade Behavior property for Siebel Marketing objects in the Siebel Tools Repository is set to ADMIN.)

This means that the Repository upgrade will not preserve customizations to Siebel Marketing views and applets. If you select ICL mode for your upgrade, then other views and applets (except those for Siebel Marketing) will be preserved under ICL.

Summary of Siebel Marketing Database Changes

Table 13 summarizes Siebel Database changes that apply to Siebel Marketing.

Table 13. Summary of Siebel Database Changes at Siebel 7.7

Upgrade Step	Description
Updates S_SRC and S_SRC_REL	Updates marketing plan tactics to use the S_SRC. MKTG_PLAN_ID to indicate the parent marketing plan, rather than S_SRC_REL.
Updates S_SRC	Updates Status values (STATUS_CD) for campaigns and programs. Updates campaign plans and stand-alone campaigns to use the same CAMP_TYPE_CD. Updates the values for the S_SRC. RESPNSE_TYPE.
Inserts into S_DD_CAMP_WAVE and S_CAMP_LD_WAVE	Creates a Load record for each campaign occurrence. Load numbers are assigned based on date of execution. Creates one load wave for each stand-alone campaign. Sets the status of the load wave to the corresponding execution status from the original campaign occurrence.
Inserts into S_CAMP_WAVE_DCP	Records the offer history for load wave records (preupgrade occurrences).
Index change and update to S_CAMP_CON	Modifies the user key to use Campaign ID + Contact ID + Load Number + Token Number. When campaign occurrences are converted to Load records, S_CAMP_CON is updated to the appropriate Load number for the campaign contact.
Updates S_SRC_COST	Updates TYPE_CD to the new values for Fixed and Per Unit expenses.
Updates S_SRC_GOAL	Updates goal type codes (GOAL_TYPE_CD).
Updates S_EVT_ACT	Re-parents activities from campaign occurrences to campaigns. Re-parents activities from program occurrences to programs.
Updates S_SRC for Stages	Updates CAMP_TYPE_CD = "STAGE" for stage records.
Updates S_OPTY_SRC	Re-parents opportunities from campaign occurrences to campaigns.

Table 13. Summary of Siebel Database Changes at Siebel 7.7

Upgrade Step	Description
Updates S_SRC_DCP	Re-parents offers from campaign occurrences to campaigns.
Updates S_ORDER	Re-parents orders from campaign occurrences to campaigns.
Updates S_CS_PATH_SRC	Re-parents SmartScripts from campaign occurrences to campaigns.
Updates S_SRC_POSTN	Re-parents team members from campaign occurrences to campaigns.
Updates S_COMMUNICATION	Re-parents responses from campaign occurrences to campaigns.
Updates S_CAMP_SKILL	Re-parents Assignment Skills from campaign occurrences to campaigns.
Updates S_QTA_PLAN	Re-parents Quota Plans from campaign occurrences to campaigns.
Updates S_PROD_INT_SRC and S_PROD_LN_SRC	Re-parents related products and product lines from campaign occurrences to campaigns.
Updates S_CALL_LST	Re-parents internal lists from campaign occurrences to campaigns.
Inserts into S_SRC_GOAL (for Campaign Plan)	Updates actual and forecast values for Goals based on Financial Modeler columns in S_SRC (Revenue, #Leads, Response Rate, Conversion Rate, Avg. Revenue per Sale, Avg. Contribution Margin, ROI Amount).

Summary of How Marketing Data Will Be Migrated

Table 14 describes how Siebel Marketing data will be migrated.

Data that is obsolete as of Release 7.7 is listed in [Appendix B, “Siebel Marketing Upgrade Reference.”](#)

Table 14. Siebel Marketing Data Migration at Siebel 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
Marketing Plans	Preserved	N/A	N/A	
Program Plans	Preserved	N/A	N/A	
Stage Plans	Preserved	N/A	N/A	

Table 14. Siebel Marketing Data Migration at Siebel 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
Campaign Plans	Preserved	N/A	N/A	Campaign Plans are converted to Campaigns.
Campaign History	Preserved	N/A	N/A	Index on S_CAMP_CON is modified to preserve history for campaign and campaign load.
Stand-alone Campaigns (campaigns not related to a program)	Preserved	N/A	N/A	
Response History	Preserved	N/A	N/A	
Lists	Preserved	N/A	N/A	
Offers (except eNewsletter offers)	All offer types preserved, except for eNewsletters	See Obsolete Objects table	See Obsolete Objects table	
Segments	Usage history in programs is preserved Segment criteria are obsolete	S_CALL_LST_CRIT S_CALL_LST_DTL S_CALL_LST_QRY	Segment Detail-DD Segment Expression Campaign Segment Allocation Segment Campaign Allocation	Segments should be reconstructed in the Release 7.7 Segment Designer.

Upgrade Planning for Siebel Workflow Designer

Upgrades: Releases 7.0.x, 7.5.x.

In Release 7.7, the Workflow Process Designer was moved to Siebel Tools. Workflow components and definitions are defined as Siebel Tools objects and are stored in the repository.

The upgrade script migrates workflows from the main Siebel Database to the repository. Before upgrading your development or production environments, make sure that they include all of the necessary workflows.

The upgrade copies or moves all workflow definitions to the repository as follows:

- **Seed workflows.** The old seed workflows are overwritten by new seed workflows. Their status is Inactive.
- **Customer workflows, status Inactive.** These are converted to workflow definitions in the repository. They will not have a status.
- **Customer workflows, status In-Progress.** These are converted to workflow definitions in the repository. Their status remains In-Progress.
- **Customer workflows, status Active.** These are converted to workflow definitions in the repository. Their status is changed to Completed. These workflow definitions are not copied to the main Siebel Database. This means that after the upgrade, no workflows are deployed. You must manually deploy seed workflows and customer workflows after the upgrade.

You must deploy and activate repository workflows in order to use them. Workflow policy object and policy program data is upgraded normally. No data is changed or lost. Database triggers are not upgraded. After the upgrade, you must regenerate database triggers.

For information on how to deploy workflows, see *Siebel Business Process Designer Administration Guide*.

Schema Changes

The main Siebel Database tables that contain workflow definitions have changed. The new tables contain the workflow definitions for deployed workflows. The definitions of workflows that are Inactive or In-Progress are located in the repository. The tables are named as follows:

- Siebel Database tables containing workflow information begin S_WFA.
- Repository tables containing workflow information begin S_WFR.
- Siebel Database tables that contain workflow information for releases prior to Release 7.7 begin S_WF_ (note the underscore after WF). After the upgrade to Release 7.7, these tables are obsolete and are not referenced by applications.

Upgrade Planning for Handheld Devices

Upgrades: Releases 7.0.x, 7.5.x.

Device operating system support, third-party product support, and application configuration management for handheld devices were revised in Release 7.7. Before the upgrade, verify that handheld devices are running an OS supported by Release 7.7. Also, verify that third-party software is the correct version.

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

Supported Device OS

The 2002 version of the Pocket PC operating system is no longer supported as of Release 7.7. Devices running Pocket PC 2002 must be upgraded to Pocket PC 2003. For a full description of supported handheld devices and operating systems, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

Active Sync Support

The third-party product Active Sync must be at version 3.7.1 or higher as of Release 7.7. For a complete description of supported third-party products, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

Handheld Device Administration

Release 7.7 introduced a handheld device administration screen. If you have modified the PDA portion of the Siebel Sales, Siebel Service, Siebel ePharma, or Siebel eConsumer application .cfg file, do the following before upgrading:

- Save the modified .cfg file under a new name so that it is not overwritten during the upgrade.
- After the upgrade, go to the Administration—Mobile screen and enter the changes from the saved .cfg file. For information on using this screen, refer to the Handheld guide for your application.

Handheld Application Upgrade

Handheld applications do not upgrade automatically. Users must uninstall the application and install Release 7.7.

Upgrade Planning for Resonate Central Dispatch

Upgrades: Releases 7.0.x, 7.5.x.

Support for Resonate Central Dispatch is discontinued as of Release 7.7. It has been replaced by a load balancing module that is included in the Siebel Web Server Extension. In addition, Siebel Systems has certified several third-party HTTP load balancers for use with the Siebel Web Server Extension.

For a description of the Siebel load balancing module, see *Siebel System Administration Guide* and *Siebel Installation Guide* for the operating system you are using. For a list of supported HTTP load balancers, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

Upgrade Planning for String Translation

Upgrades: Releases 7.0.x, 7.5.x.

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

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

CAUTION: If you have modified string mappings in a previous release, you must obtain Siebel assistance before upgrading. If you do not obtain assistance, you could lose or corrupt data during upgrade.

8

Preparing Application Data for Upgrade

This chapter contains the following topics:

- “Preparing Workflow Processes for Upgrade” on page 131
- “Preparing Mobile and Dedicated User Data for the Database Upgrade” on page 132
- “Setting Up Campaign Status Values for Siebel Marketing Upgrade” on page 133
- “Preparing Address Data for Upgrade” on page 135
- “Preparing Products for Upgrade” on page 136
- “Preserving Marketing Segment Descriptions” on page 136
- “Resolving Duplicate Row IDs” on page 137
- “Preparing Siebel eChannel Data for Upgrade” on page 137
- “Preparing Forecasting Data for Upgrade” on page 138
- “Preparing Products and Quotes for Upgrade” on page 138
- “Verifying Household Data Integrity” on page 139
- “Preparing Financial Services Application Tables for Upgrade” on page 141
- “Preparing Siebel Configurator Data for Upgrade” on page 141

Preparing Workflow Processes for Upgrade

Upgrades: All upgrades.

Environments: All environments.

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

Deactivate all workflow processes before doing the upgrade. This ensures that run-time event actions associated with old workflows are cleaned up properly.

Development Environment Upgrades

Before upgrading your development environment, make sure that all customized workflows and all seed workflows from the production database are present in the development environment Siebel Database. The upgrade will move them to the Siebel Repository and also update the seed workflows. The upgrade migrates custom workflows from S_WF_* tables to S_WFR_* tables in the Siebel Repository.

If the workflows from the production environment are not present, they will not be moved to the Siebel Tools Repository. You will have to manually move them using database tools.

Siebel Systems provides a script and an input file for migrating workflow data between your production database and your development database. See Siebel Alert 1016 on SupportWeb.

Preparing Mobile and Dedicated User Data for the Database Upgrade

Upgrades: All upgrades.

Environments: All environments.

This topic applies primarily to developers running the Mobile Web Client in the development environment and to end users in the production environment. This topic applies to the development environment only if it has Mobile Web Client users.

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

After synchronizing, mobile users must make no further changes to their local databases until the upgrade has been completed. Any changes made during the upgrade are lost when they are reinitialized following the upgrade.

Complete the following steps before beginning the upgrade of either a development environment or a production environment.

To prepare mobile users for the database upgrade

- 1** Perform a partial synchronization for mobile users, sending all transactions to the Siebel Database Server.
- 2** Verify that Mobile Web Clients have synchronized and that all changes have been merged into the Siebel Database Server as follows:
 - a** Check that no transaction files remain in the synchronization inbox and outbox for any mobile user. The synchronization inbox for each user is on the Siebel Server:

Windows: `SI EBEL_ROOT\docki ng\MOBI LEUSERNAME.`

UNIX: `$SI EBEL_ROOT/docki ng/MOBI LEUSERNAME.`

Transaction files are in the format *number.dx*; for example, 00000023. dx.
 - b** Log onto a Siebel eBusiness application, such as Call Center, as the Siebel administrator. Use the Server Administration - Server Tasks screen to make sure that each Transaction Merger task has successfully completed.
 - c** Verify that Workflow Monitor and Workflow Action agents have processed all pending requests. If Workflow Manager has completed successfully, the S_ESCL_REQ table should not have any rows.
- 3** To prevent synchronization of Mobile Web Clients with the Siebel Database Server, stop or disable all Siebel Remote components on all Siebel Servers, as described in *Siebel Remote and Replication Manager Administration Guide* and in *Siebel System Administration Guide*.

4 Disconnect all Web Clients from the Siebel Server by stopping the appropriate Application Object Managers, as described in *Siebel System Administration Guide*.

5 **Upgrades from Release 7.5x and earlier.** Make sure that Dedicated Web Clients have disconnected from the Siebel Database Server.

The method you use to do this depends on your database. For example, with an IBM RDBMS, you would stop the primary listener. However, all RDBMS types require starting the database in restricted mode. Refer to the documentation that you received from your RDBMS vendor for more information.

Setting Up Campaign Status Values for Siebel Marketing Upgrade

Upgrades: All upgrades.

Environments: Production environment only.

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

This optional step is not required for upgrade processing, but will simplify program and campaign reporting by making historical values equivalent with the values used as of Release 7.7.

The database upgrade process expects the standard Status values for the following applets. The process uses the standard values to convert records to the new campaign model during the upgrade:

- Campaigns (business component is Campaign)
- Campaign Plans (business component is DBM Campaign)
- Campaign Occurrences (business component is Campaign Occurrences)

If you customized the values for the Status field for these objects, convert the Status (S_SRC. STATUS_CD) of any existing records to the standard values before doing the upgrade.

Any campaign records that do not have standard Status values will not create Load Wave records during the upgrade and will not be displayed as launched campaigns in the Campaign Agent views.

Optionally, you can also migrate the status values for the following components:

- Program (business component is Program Container)
- Stage (business component is Program (DBM))

Table 15 lists the standard Status values for each object.

Table 15. Standard Status Values

Object	Standard Status Values
Program Plan	Planned Active Completed
Stage Plan	Planned Active Completed
Campaign Plan	Planned Active Completed
Campaign	Planned Active Completed
Campaign Occurrence	Pending Active Finished Manual Error Cancelled

Table 16 lists search specifications for locating records containing Status values. Release 7.5 terminology is used for the object names.

Table 16. Search Specifications for Locating Records

Object	Business Component	Base Table	Search Specification
Program Plan	Program Container	S_SRC	WHERE SUB_TYPE = 'MARKETING_CAMPAIGN' AND CAMP_TYPE_CD = 'PROGRAM_CONTAINER'
Stage Plan	Program (DBM)	S_SRC	WHERE SUB_TYPE = 'MARKETING_CAMPAIGN' AND CAMP_TYPE_CD = 'PROGRAM'

Table 16. Search Specifications for Locating Records

Object	Business Component	Base Table	Search Specification
Campaigns	Campaign	S_SRC	WHERE SUB_TYPE = 'MARKETING_CAMPAIGN' AND CAMP_TYPE_CD IS NULL AND CUST_TRGT_METH_CD IS NULL AND TMPL_ID IS NULL
Campaign Plans	DBM Campaign	S_SRC	WHERE SUB_TYPE = 'MARKETING_CAMPAIGN' AND CAMP_TYPE_CD IS NULL AND CUST_TRGT_METH_CD IS NOT NULL
Campaigns (Occurrences)	Campaign Occurrences	S_SRC	WHERE SUB_TYPE = 'MARKETING_CAMPAIGN' AND CAMP_TYPE_CD IS NULL AND CUST_TRGT_METH_CD IS NULL

Preparing Address Data for Upgrade

Upgrades:

- From Release 6.x Siebel Industrial applications (except Communications, Media, and Energy) that you are upgrading directly to Release 7.7 Siebel Industrial applications
- From Release 6.x and 7.0.x Siebel Financial Services applications that you are upgrading to Release 7.7 Siebel Industrial applications
- From Release 6.x and 7.7 Siebel eBusiness applications that you are upgrading to Release 7.7 Siebel Industrial applications
- From Release 6.2.1 of Siebel Financial Services applications on IBM z/OS platforms that you are upgrading to Release 7.7.x of Siebel Industrial applications on IBM z/OS platforms

NOTE: This topic does not apply to Siebel eBusiness applications (HOR) that you are upgrading to a later release of Siebel eBusiness applications (HOR).

Environments: Production environment only.

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

The way address data is stored is changed at Release 7.7. To prepare for the revised storage scheme, you must verify that there are no records with the same row IDs across the tables S_ADDR_PER, S_ADDR_ORG, and S_CON_ADDR.

CAUTION: There must be no duplicate row IDs in these tables or the upgrade will fail.

To prepare address data for upgrade

- 1 Run `rpt_dup_addr_rowids.sql` against the Siebel Database. The script is located in the following directory:

Windows: `DBSRVR_ROOT\database_platform`

UNIX: `DBSRVR_ROOT/database_platform`

where

`database_platform` is the database type, for example DB2.

- 2 Review the output generated by the script.
- 3 If the output contains records with duplicate row IDs, revise the `ROW_ID` value of one of the records.
- 4 After addressing all the duplicate row IDs, rerun the script and verify there are no more duplicates.

Preparing Products for Upgrade

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

Environments: Production environment only.

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

Customizable Products in Work Spaces

The upgrade does not migrate unreleased customizable products in work spaces. If you want to migrate customizable products, you must release them prior to the upgrade. This includes products with components and products with attributes.

Class Products

Verify that the Orderable flag is not set for class products. With this flag unset, class products do not display as selectable products in quotes and orders after the upgrade.

The upgrade converts class products to a product and a product class. The upgrade sets the Product Class property for the product to Product Class.

Preserving Marketing Segment Descriptions

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

When upgrading from a prior release of Siebel Marketing to Release 7.7 or later, the database upgrade overwrites the Description field for all Segment records (S_CALL_LST.DESC_TEXT) with the string *DO NOT USE: Segment from previous release*. If you want to preserve segment descriptions, migrate the existing descriptions to another text extension column before running the upgrade.

Resolving Duplicate Row IDs

Upgrades: Release 6.x only.

Environments: All environments.

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

During the upgrade from Release 6.x to 7.7, 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 source 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 Server software, but before you upgrade, you must identify and resolve any duplicate ROW_IDs in your Siebel Database.

To resolve duplicate ROW_IDs across source tables

- 1 Run the script, `Find_DUP_S_PARTY_ROW_IDs.sql`, located in the following directory:

WINDOWS: `DBSRVR_ROOT\DATABASE_PLATFORM`

UNIX: `DBSRVR_ROOT/DATABASE_PLATFORM`

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.x only.

Environments: Production environment only.

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

For upgrades from Release 6.x to 7.7, 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 Siebel Professional Services.

Preparing Forecasting Data for Upgrade

Upgrades: Release 6.x only.

Environments: Production environment only.

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

For upgrades from Release 6.x to 7.7, 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 eBusiness 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.x only.

Environments: Production environment only.

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

Customizable Products

Determine which of your Release 6.x configuration solutions are to be upgraded to Customizable Product quote items in Release 7.7. 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.x 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 *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.x 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 7.7, 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
```

Verifying Household Data Integrity

Upgrades: Release 6.x only.

Environments: Production environment only.

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

If you choose to implement the new household design, you must run the household verification script (HH_MIG_populate.sql) before upgrading to Release 7.7.

If you choose not to upgrade to the new household design, do not run the script. Also, make sure that the temporary table TEMP_HH_OBJ_MIG does not exist in the database.

The household verification script verifies household data integrity. The script verifies that at least the same number of entities will belong to a household after the upgrade as belong to it before the upgrade. The script ensures that after the upgrade, you will see the same or more household data than before the upgrade.

The household verification script makes the following assumptions:

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

The script populates a temporary table with data, TEMP_HH_OBJ_MIG. The script generates a report based on an output file. Output is in the form of row IDs.

If there is no output, this means Siebel 6.x data integrity is good, and no action is required. If you receive output, this means that no contact is associated with a household for a particular entity.

To run the household verification utility

1 Type the following command:

- Windows:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /a /c REM /separator / /O
OutputFileLocation\HH_Mig_populate.txt
/L LogFileLocation\HH_Mig_populate.log ScriptLocation\HH_Mig_populate.sql /v y
```

- UNIX:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /a /c REM /separator / /O
OutputFileLocation/HH_Mig_populate.txt
/L LogFileLocation/HH_Mig_populate.log ScriptLocation/HH_Mig_populate.sql /v
```

where:

Tableowner = Tableowner

Password = Tableowner password

ODBCDataSource = Data source of the database

OutputFileLocation = Location of the output file:

- Windows: *SIEBEL_ROOT*\Log\HH_Mig_populate.txt

- UNIX: *\$SIEBEL_ROOT/Log/HH_Mig_populate.txt*

LogFileLocation = Location of the log file :

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

- ❑ UNIX: \$SIEBEL_ROOT/Log/HH_Mi g_populate. log
- ScriptLocation = Location of the script:
- ❑ Windows: *DBSRVR_ROOT\database_platform\HH_Mi g_populate. sql*
- ❑ UNIX: *DBSRVR_ROOT/database_platform/HH_Mi g_populate. sql*

Windows example:

```
odbcsql /U Tableowner /P Password /S ODBCDataSource /a /c REM /separator / /O  
C:\sea7xx\si ebsrvr\Log\HH_Mi g_populate. txt /L  
C:\sea7xx\si ebsrvr\Log\HH_Mi g_populate. log  
C:\sea7xx\dbsrvr\DB2UDB\HH_Mi g_populate. sql /v y
```

- 2 If you receive output, review the temporary table and verify the following for each contact. Make corrections as needed:
 - Contact is correct and household is incorrect.
 - Contact is incorrect and household is correct.
 - Contact is incorrect and household is incorrect.

At least one contact associated with an entity must also be associated with the household for that entity.

Preparing Financial Services Application Tables for Upgrade

Upgrades: Release 6.x only.

Environments: All environments.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform the 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.x to Release 7.7.

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

6.x upgrades only. This topic applies to upgrades from Release 6.x.

Environments: Development environment only.

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 other recommended preupgrade tasks, see *Siebel Interactive Selling Applications Upgrade Guide*.

9

Preparing a Development Environment for Upgrade

This chapter contains the following topics:

- “Prerequisites for a Development Upgrade” on page 143
- “About Moving Tables” on page 144
- “Checking In Development Repository Projects” on page 145
- “Saving Interface Table Data” on page 145
- “Upgrading Your DB2 Software” on page 146
- “Preparing EIM Tables for Upgrade” on page 146
- “Preparing for Intersection Table Maintenance and Deduplication” on page 148
- “Determining Which Template File was Used During an Extract or Merge” on page 149
- “Disabling Customized Triggers” on page 150
- “Dropping Customized Views” on page 150
- “Preparing the Development Environment Repositories for Upgrade” on page 150

Prerequisites 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 Server Configuration” on page 103](#), 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 *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 C, “Upgrade Planning Worksheet,”](#) and fill out the appropriate fields with the following information.

- **ODBC Data Source Name.** This name must be the same as the name of the database qualifier. If you create a new datasource to use for your upgrade or installation other than the default ODBC datasource created by Siebel Systems, you must add the TXNISOLATION=1 parameter in the db2cli.ini file for the new datasource.

- **Database User Name.** Type the ID allowed to log into the Siebel Database. This user ID should have authorization to set CURRENT SQLID and must be set up as a Siebel Systems employee. The employee record 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 prior to proceeding with the upgrade.

This login must be the same as the database user name that you specify in the upgrade configuration utility.
- **Database Password.** Type the password for the ID used to log into the Siebel Database.
- **Schema Qualifier.** Type the up to eight-character identifier that designates Siebel Schema. This is also an authorization ID. The schema qualifier must start with a letter, cannot contain special characters, and must be entered in uppercase.
- **Group Authorization ID.** The user ID of the group to whom schema access is granted, for example, SSEROLE.
- **Storage Group for Indexes.** The name of the storage group used for temporary indexes. The default value is SYSDEFLT.

About Moving Tables

Upgrades: All upgrades.

Environments: Development environment only.

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

Some of the tables are going to 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 tablespace or bufferpool. Moving the identified tables to a new, larger and differently named tablespace allows you to maintain the paradigm of multiple tables per tablespace which is present for all pre-7.7 schemas.

Before you move the tables, you need to drop any standard or custom views. 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 tablespace within the same database as the older, smaller tablespace.

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 = 'CQ10K003'
AND BNAME IN ('S_ASSET', 'S_PROD_INT');
```


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

```
SELECT DISTINCT 'DROP VIEW ' || 'CQ10K003.' || V.DNAME || ' ';
FROM SYSIBM.SYSVIEWDEP V
WHERE V.BNAME IN ('S_ASSET', 'S_PROD_INT')
WHERE V.BNAME = 'S_ASSET'
AND BCREATOR = 'CQ10K003';
```

In order to process these drop statements on the zSeries 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 tablespace 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 7. The S_ESCL_REQ table should not have any rows if Workflow Manager has completed successfully.

Saving Interface Table Data

Upgrades: All upgrades.

Environments: Development environment only.

Use the appropriate tool to export any data in your interface tables that you want to retain. During the upgrade process, your interface tables are dropped and then recreated. After you have upgraded, you can import this data using the appropriate tools.

NOTE: During the upgrade, all custom indexes on interface tables are dropped from both logical and physical schema.

Upgrading Your DB2 Software

Upgrades: All upgrades.

Environments: Development environment only.

Before you upgrade, carefully review *System Requirements and Supported Platforms* on Siebel SupportWeb to verify that you are using the currently supported versions of DB2 software.

- **DB2 Version 7.** 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 7.7. You should perform your DB2 version upgrade on a separate subsystem from the one that you plan to use for Siebel 7.7. Plan to allow at least two weeks of testing to verify that your DB2 upgrade was successful before you proceed with the upgrade of your Siebel Database.
- **DB2 Connect.** Siebel Dedicated 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 7.7.

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

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 your DDL output directory and open the predevftp.txt file.
 - Windows: The default DDL output directory is `SIEBEL_ROOT\dbsrvr\db2390\`
 - UNIX: The default DDL output directory is `$SIEBEL_ROOT/dbsrvr/db2390/`
- 2 Edit the predevftp.txt file using the following information:
 - a Change &IP into the IP address of your DB2 host, for example, ZM01.
 - b Change &Username into the your own user name, for example, sadmin.
 - c Change the constant SIEBELQ1 to your own dataset high-level qualifier (*DSNHLO*), for example, sadmin, using a command like Replace All.

Use the same high-level qualifier for all zSeries upgrade datasets. The *DSNHLO* 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 *DSNHLO* can be the TSO ID.
 - d Change &SiebelRoot to your `SIEBEL_ROOT` directory, for example, `D:\7.7.`
- 3 After modifying the predevftp.txt file, save it, and then close the file.

- 4 Double-click ftppredev.bat (located in *SI EBEL_ROOT\dfsrvr\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.
All files generated to this point of the upgrade process transfer from the midtier to the zSeries.
- 6 After the transfer is successful, log on to the mainframe, and navigate to the *DSNHLO.PREDEVO.JCL* file, for example, *SI EBELQ1.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:

```
//###PREDEV JOB ACCNT#, CLASS=S, MSGCLASS=X, MSGLEVEL=(1, 1),
//          TIME=1440, REGION=OM, NOTIFY=&SYSUID, LINES=9999
//*
/*JOBPARM S=QA01
```

- 8 From the command line, execute the following commands:

- C *SI EBELQ1 DSNHLO* all

Where:

DSNHLO is the high-level qualifier you specified in [Step 2 on page 146](#).

- C <SCHEMA> *YOUR_DB2_SCHEMA*

For example,

```
C <SCHEMA> CQ10K003 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 eBusiness applications:

```
DSNHLO. PRE. HOR621. JCLLIB
```

```
DSNHLO. PRE. HOR704. JCLLIB
```

```
DSNHLO. PRE. HOR752. JCLLIB
```

□ For upgrades of Siebel Industry applications:

```
DSNHLO. PRE. SI A621. JCLLIB
```

```
DSNHLO. PRE. SI A752. JCLLIB
```

```
DSNHLO. PRE. SI S63. JCLLIB
```

```
DSNHLO. PRE. SI S704. JCLLIB
```

11 Edit the dataset from 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 DB2390 directory under *DBSRVR_ROOT* 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 DB2390 directory under *DBSRVR_ROOT* 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, SIEBDB, with your database name prefix. Your database name prefix must be a maximum of four characters, for example *nnnn*, where *nnnn*. 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* under the DB2390 directory under *DBSRVR_ROOT*.

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 the 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 upgwiz.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\strgxrct.exe /u QADMIN /p ***** /c Q10K /
d CQ10K029 /i D:\18025\si_ebsrvr\log\dbconfi_g_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/override.inp /8 D029 /9 SIEBTS /1 SYSDEFLT
/2 SYSDEFLT /3 BP1 /4 BP2 /7 EBCDIC
```

Disabling Customized Triggers

Upgrades: All upgrades.

Environments: Development environment only.

Release 7.7 does not support customized triggers. If you have created customized triggers on your Siebel base tables, disable them before you perform the upgrade. You must recreate them after the upgrade is finished.

Dropping Customized Views

Upgrades: All upgrades.

Environments: Development environment only.

If you created customized views on Siebel base tables, you must drop them before you perform the upgrade. If they are still applicable after the upgrade, you must recreate them after the upgrade is finished.

Preparing the Development Environment Repositories for Upgrade

Upgrades: All upgrades.

Environments: Development environment only.

Four separate repositories are used during the development upgrade process.

Your existing development repository. To prevent a naming conflict, before you run the upgrade, rename your existing development repository (*Siebel Repository*) to *Prior Customer Repository*. After the upgrade, your new Release 7.7 development repository is given the name *Siebel Repository*.

Three new repositories. These are automatically loaded when you run the Siebel Upgrade Wizard:

- Prior standard repository
- New standard repository (Release 7.7)
- New customer repository (Release 7.7) which, after the merge, becomes the customized 7.7 repository.

Before these repositories are loaded, verify that your existing repositories do not use the names reserved for the upgrade process:

- **New Customer Repository.** Your new custom repository, which results from the merge, and includes your custom configurations.
- **New Siebel Repository.** The new Siebel standard repository.
- **Prior vX.x Siebel Repository.** The prior Siebel Repository.

To avoid running out of space when importing these repositories, make sure you correctly size your tables. The following tables have a high number of inserts during imprep:

- S_APPLET
- S_APPLET_I NTL
- S_APPLET_METH_MI
- S_APPL_WEB_TMPL
- S_APPL_WTMPL_I T
- S_BOCOMP
- S_BUSCOMP_UPROP
- S_COLUMN
- S_CONTROL_I NTL
- S_CONTROL_UPROP
- S_DDOWN_OBJECT
- S_EI M_FK_MAPCOL
- S_FI ELD
- S_I NDEX
- S_I NDEX_COLUMN
- S_I NTFLD_UPROP
- S_I NTFLD_CKEY_FLD
- S_I NTFLD_COMP
- S_I NTFLD_FI ELD

- S_JOIN
- S_JOIN_SPEC
- S_LIST
- S_LIST_COLUMN
- S_PICKMAP
- S_SCREEN_VIEW
- S_UK_ATTJOIN
- S_LIST_COL_INTL
- S_USER_KEY_ATT
- S_VIEW_WTMPL_IT
- S_CONTROL
- S_EIM_ATT_MAP

Use the version of Siebel Tools that matches your old version of Siebel eBusiness applications (for example, Siebel Tools Release 6.x for upgrades from Release 6.x). Connect directly to the development Siebel Database Server, and then perform the following steps to make sure that your existing repositories follow the correct naming conventions:

To rename the repository

- 1** Using the appropriate prior version of Siebel Tools, connect to the Siebel Database Server.
- 2** Change the name of the existing repository, as described below:
 - a** In the Object Explorer, click the Types tab.
 - b** Click Repository.
 - c** In the Repository view, click Name.
 - d** Locate the appropriate repository in the list applet and rename it Prior Customer Repository.
- 3** Step off the list to commit the record to the database.

NOTE: The upgrade process verifies the repository names. If no repository is named Prior Customer Repository, the Siebel Repository is renamed to Prior Customer Repository in the target database so that the upgrade executes properly.

If you need more information about renaming repositories, refer to *Configuring Siebel eBusiness Applications*.

10 Preparing a Production Environment for Upgrade

This chapter contains the following topics:

- “Prerequisites for a Production Upgrade” on page 153
- “Importing the Development Repository and Schema Definition” on page 155
- “Saving Interface Table Data” on page 155
- “Preparing Production Environment Repositories for Upgrade” on page 156

Prerequisites for a Production Upgrade

Upgrades: All upgrades.

Environments: Production environment only.

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 development 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 [“Verifying Database Server Configuration” on page 103](#).

NOTE: Verify that your production database uses binary sort order.

About Subsystems and Tableowners

[Table 17](#) describes the subsystem and tableowner requirements for the production upgrade.

Table 17. Production Upgrade Subsystem and Tableowner Requirements

Subsystem	Tableowner
One subsystem	Two tableowners
Two subsystems	One tableowner
Two subsystems	Two tableowners

If you have not already done so, contact your database administrator and get the following information about your source database and your target database.

Source Database Values

For the production environment upgrade, you need the following values for your source database:

- **ODBC Data Source Name.** This name must be the same as the name of the database qualifier.

NOTE: On Microsoft Windows, the Siebel Server installation process automatically creates the data source, using the format `Si ebSrvr_EnterpriseName`. 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.

- **Database User Name.** The ID allowed to log into the Siebel Database. This user ID should have authorization to set CURRENT SQLID and must be set up as an employee within your Siebel eBusiness application. The employee record must have the Siebel administrator responsibility. SADMIN is the default administrator user name and password, but you can use any name that has SADMIN credentials. 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 prior to proceeding with the upgrade. For more information, see [“Adding a Siebel User” on page 118](#).

This login must be the same as the target database user name that you will specify in the upgrade configuration utility.

- **Database Password.** The password for the ID used to log into the Siebel Database.
- **Schema Qualifier.** The up to eight-character identifier that designates Siebel Schema. This is also an authorization ID. The schema qualifier must start with a letter, cannot contain special characters, and must be entered in uppercase.
- **Group Authorization ID.** The user ID of the group to whom schema access is granted, for example, SSEROLE.

Target Database Values

For the production environment upgrade, you need the following values for your target database:

- **Schema Qualifier.** The up to eight-character identifier that designates Siebel Schema. This is also an authorization ID. The schema qualifier must start with a letter, cannot contain special characters, and must be entered in uppercase.
- **DB2 Subsystem.** Verify that there are no objects in the target DB2 subsystem before you start the upgrade.
- **Database User Name.** The ID allowed to log into the Siebel Database, for example, SBLDBA. This user ID should have authorization to set CURRENT SQLID and must be set up as an employee within your Siebel eBusiness application. The employee record must have the Siebel administrator responsibility. SADMIN is the default administrator user name and password, but you can use any name that has SADMIN credentials. 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 prior to proceeding with the upgrade. For more information, see [“Adding a Siebel User” on page 118](#).

This login must be the same as the target database user name that you will specify in the upgrade configuration utility.

- **Database Password.** The password for the ID used to log into the Siebel Database.

- **Group Authorization ID.** The user ID of the group to whom schema access is granted, for example, SSEROLE.
- **Storage Group for Indexes.** The name of the storage group used for temporary indexes. The default value is SYSDEFLT.
- **Storage Group for Tables.** The name of the storage group used for temporary tables. The default value is SYSDEFLT.
- **ODBC Data Source Name.** This name must be the same as the name of the database qualifier.

The remaining information required for upgrade will be derived from your prior source database layout.

Importing the Development Repository and Schema Definition

Upgrades: All upgrades.

Environments: Production environment only.

To import the development repository and schema definition

- 1 Navigate to your development environment's `DBSRVR_ROOT\PLATFORM` directory (this may be either DB2 UDB or DB2 UDB for z/OS).
- 2 Copy the following files that were created during your development environment upgrade:
 - `custrep.dat`
 - `schema.ddl`

- 3 Place these files in the `DBSRVR_ROOT\PLATFORM` directory for your production environment.

The `custrep.dat` file will be used by the Siebel Upgrade Wizard to import the New Customer Repository. The `schema.ddl` file will be used by the Siebel Upgrade Wizard to create the new database schema.

NOTE: If you made any changes after performing your development environment upgrade and before your production environment upgrade, you need to export your modified repository (`custrep.dat`) and your modified schema definition (`schema.ddl`) to your production environment.

Saving Interface Table Data

Upgrades: All upgrades.

Environments: Production environment only.

Use the appropriate tools for your business to unload any data from your interface tables that you want to retain. During the upgrade process, your interface tables will be dropped and then recreated. After you have upgraded, you can then import this data using your standard tools.

NOTE: During the upgrade, all custom indexes on interface tables are dropped from both logical and physical schema.

Preparing Production Environment Repositories for Upgrade

Upgrades: All upgrades.

Environments: Production environment only.

Two separate repositories are used during the production upgrade process:

- **Your existing production repository.** To prevent a naming conflict, before you run the upgrade, rename your existing production repository (*Siebel Repository*) to *Prior Customer Repository*. After the upgrade, your new Release 7 production repository will be given the name *Siebel Repository*.
- **New Customer Repository.** This repository will be loaded when you run the Siebel Upgrade Wizard.

CAUTION: Your upgrade will encounter errors if you have more than one existing repository for a production upgrade. Export and archive any redundant repositories from your environment before you upgrade your production environment.

Renaming Your Existing Production Repository

You must rename your existing production repository so that it is recognized by the upgrade process.

To rename your existing production repository

- 1 Use the version of Siebel Tools that matches your prior version of Siebel eBusiness applications (for example, Siebel Tools Release 6.x for upgrades from Release 6.x) to connect to the Siebel Database Server.
- 2 Change the name of the existing repository:
 - a In the Object Explorer, click the Types tab.
 - b Click Repository.
 - c In the Repository view, click Name.
 - d Locate the appropriate repository in the list applet and rename it *Prior Customer Repository*.

TIP: Differentiate this repository from your development environment *Prior Customer Repository* by adding a prefix to the name; for example, 20031202 Prod Pri or v6. x Customer Reposi tory.

- 3 Step off the list to commit the record to the database.

NOTE: The upgrade process will verify the repository names. If no repository is named *Prior Customer Repository*, the Siebel Repository will be renamed to *Prior Customer Repository* in the target database so that the upgrade will execute properly.

If you need more information about renaming repositories, refer to *Configuring Siebel eBusiness Applications*.

11 Upgrading the Database

This chapter contains the following topics:

- [“Changing the Database Server Configuration Utility Language” on page 159](#)
- [“Preparing to Run the Database Server Configuration Utility” on page 160](#)
- [“Running the Database Server Configuration Utility Under Windows” on page 163](#)
- [“Running the Database Server Configuration Utility Under UNIX” on page 164](#)
- [“Running the Siebel Upgrade Wizard” on page 165](#)
- [“Deploying Stored Procedures and User-Defined Functions” on page 167](#)
- [“Migrating Address Data from Custom Extension Columns” on page 172](#)
- [“Preparing for a No-Development-Environment Upgrade” on page 175](#)
- [“Installing New License Keys After Upgrade” on page 176](#)

Changing the Database Server Configuration Utility Language

Upgrades: All upgrades.

Environments: All environments.

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

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

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

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

For more information about the Database Server Configuration utility, see [“About the Database Server Configuration Utility” on page 34.](#)

Preparing to Run the Database Server Configuration Utility

Upgrades: All upgrades.

Environments: All environments.

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

Use this topic to identify the information you must enter when running the Database Server Configuration utility. Collect this information and verify it before running the utility.

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

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

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

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

[Table 18](#) lists the information that you must enter in the Database Server Configuration utility.

[Table 19 on page 163](#) lists additional information the utility requires when you perform a production environment or production upgrade.

For more information on the Database Server Configuration utility, see [“About the Database Server Configuration Utility” on page 34](#).

Table 18. Information Required for the Database Server Configuration Utility

Screen Name	Required Information
Siebel Gateway Name Server Address	Windows only. The Siebel Gateway Name Server machine name and the Enterprise Server name.
Siebel Server Directory	The absolute path of the directory where the Siebel Server is installed. For UNIX, do <i>not</i> enter the string \$SIEBEL_ROOT—for example D:\sea_7x\si ebsrvr.
Siebel Database Server Directory	The absolute path of the directory where the Siebel Database Server is installed—for example D:\sea_7x\dbsrvr.
RDBMS Platform	Choose IBM DB2 UDB for z/OS.
Siebel Database Operation	Choose Upgrade Database. The other menu choices are for database installation and administration.

Table 18. Information Required for the Database Server Configuration Utility

Screen Name	Required Information
Environment Type	<p>Choose Development for development environment upgrades.</p> <p>Choose Production for production environment upgrades.</p>
Upgrade Options	Choose Upgrade Siebel Database Schema (upgrep + upgphys).
Siebel Industry Application	<p>This displays 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 eBusiness application as part of upgrading to the new Siebel Industry application release, choose Siebel Horizontal application.</p>
Current Siebel Version	Choose the application version you are upgrading from.
Language	Select the appropriate language for your upgrade.
Source Database ODBC Data Source Name	<p>Verify the ODBC name for connecting to the source Siebel Database for your upgrade. If it is not correct, enter the correct ODBC name.</p> <p>The ODBC data source <i>must</i> have the same name as the subsystem. When you set up the ODBC connection in DB2 Connect, use the actual subsystem name for the database alias.</p> <p>Windows: To find the name of your ODBC data source, navigate to Start > Settings > Control Panel > Administrative Tools > Data Source (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>
Source Database User Name	<p>Enter the source user name and password for the Siebel administrator of the source database for your upgrade.</p> <p>NOTE: The source database user name (user ID) needs to have authorization to set CURRENT SQLID.</p>
Source Siebel Schema Qualifier	Enter the eight-character identifier that designates the Siebel Schema for your source 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.
Target Database ODBC Data Source Name	<p>Verify the ODBC name for connecting to the target Siebel Database for your upgrade. If it is not correct, enter the correct ODBC name.</p> <p>The ODBC data source <i>must</i> have the same name as the subsystem. When you set up the ODBC connection in DB2 Connect, use the actual subsystem name for the database alias.</p>

Table 18. Information Required for the Database Server Configuration Utility

Screen Name	Required Information
Target Database User Name	<p>Enter the target user name and password for the Siebel administrator of the target database for your upgrade.</p> <p>NOTE: The target database user name (user ID) needs to have authorization to set CURRENT SQLID.</p>
Target Siebel Schema Qualifier and Target Security Group ID	<p>Enter the following parameter values:</p> <p>Target Schema Qualifier. Enter the up to eight-character identifier that designates 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.</p> <p>Target Security Group ID. Enter the user ID of the group to whom schema access is granted, for example, SSEROLE.</p>
Storage Group for Temporary Indexes	<p>Enter the following parameter values:</p> <p>Storage Group for Temporary Indexes. Enter the name of the storage group provided by the database administrator (the default value is SYSDEFAULT).</p> <p>Storage Control File. Use either of the following methods to select the storage control file (that you prepared in “Preparing the Storage Layout of the Schema” on page 104) that contains the previously customized database storage layout:</p> <ul style="list-style-type: none"> ■ Windows: Accept the default value displayed in the Storage Control File field (this is the <i>DBSRVR_ROOT\DB2390</i> directory. For example, <i>D:\DBSRVR_ROOT\DB2390\storage.ctl</i>) or use the Browse button to navigate to a different directory. ■ UNIX: Enter the path to your storage control file.
Primary Quantity for Temporary Index Space	<p>Enter the primary and secondary quantities for temporary index space.</p>
DDL Commit Frequency	<p>Enter the DDL commit frequency for your upgrade.</p>
DDL 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 Siebel Database schema. These files must be manually applied by the database administrator.</p>
Log Output Directory	<p>Accept the default or enter the directory name. If the directory does not exist, it will be created. Do not use special characters such as spaces, slashes, or symbols in the name of the log output.</p> <p>CAUTION: If you are restarting or recovering a previous incomplete upgrade, do not change the Log Output Directory that you previously selected.</p>

Additional Information Required for Production Upgrades

When you perform a production environment upgrade, the additional information shown in [Table 19](#) is required when you run the Database Server Configuration utility in your production environment.

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

Table 19. 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.
Database Table Owner for Development Database	Account name and password for the Siebel Database table owner in the development environment.
Import Repository Name	Enter the name of the upgraded Siebel Tools repository in the development environment database. Typically, this is <i>Siebel Repository</i> .

Running the Database Server Configuration Utility 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 the Upgrade.”](#)

Run the Database Server Configuration utility to upgrade the Siebel Database. The utility 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 upgrade the Siebel Database.

For more information on the Database Server Configuration utility, see [“About the Database Server Configuration Utility” on page 34.](#)

Prerequisite: Collect the information that the Database Server Configuration utility requires. See [“Preparing to Run the Database Server Configuration Utility” on page 160.](#)

To run the Database Server Configuration utility under Windows

- 1** Verify that no server tasks are running in the background.
If necessary, stop Siebel Servers and Siebel Gateway Name Server service by navigating to Start > Settings > Control Panel > Services.
- 2** Select Start > Programs > Siebel Enterprise Servers 7.x > Configure DB Server.
The first window of the Database Server Configuration utility appears.
- 3** Enter the information requested in each window and click Next.
To return to previous windows, click Previous.
- 4** When the runnow window displays, do the following:
 - **Upgrep mode:** Answer No. Do not start the Siebel Upgrade Wizard.
 - **Upophys mode:** Answer Yes to start the Siebel Upgrade Wizard.Just before displaying the runnow window, the Database Server Configuration utility calls the SQL generator to create or populate SQL scripts.
To manually start the Siebel Upgrade Wizard, see [“Running the Siebel Upgrade Wizard” on page 165](#).

Running the Database Server Configuration Utility 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 the Upgrade.”](#)

Run the Database Server Configuration utility to upgrade the Siebel Database. The utility 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 upgrade the Siebel Database.

For more information on the Database Server Configuration utility, see [“About the Database Server Configuration Utility” on page 34](#).

Prerequisite: Collect the information that the Database Server Configuration utility requires. See [“Preparing to Run the Database Server Configuration Utility” on page 160](#).

To run the Database Server Configuration utility under UNIX

- 1** Verify that all servers are stopped:
 - Stop all Siebel Servers by typing: stop_server all
 - Stop the Siebel Gateway Name Server by typing: stop_ns

- 2 Depending on your shell, source environment variables:

Korn or Bourne shell

```
. si ebenv. sh
```

C shell

```
source si ebenv. csh
```

- 3 Navigate to `$SIEBEL_ROOT/bin` and run the following script:

```
dbsrvr_config. ksh
```

This launches the Database Server Configuration utility.

- 4 Review the values of the following environment variables and confirm whether the settings are correct by entering either Y or N.

- `SIEBEL_ROOT`

- `LANGUAGE` (This is the language in which the Database Server Configuration utility runs.)

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

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

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

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

- **Upprep mode:** Answer No. Do not start the Siebel Upgrade Wizard.

- **Uppphys mode:** Enter Y to start the Siebel Upgrade Wizard.

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

To manually start the Siebel Upgrade Wizard, see [“Running the Siebel Upgrade Wizard.”](#)

Running the Siebel Upgrade Wizard

Upgrades: All upgrades.

Environments: All environments.

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

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

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

For more information on the Siebel Upgrade Wizard, see [“About the Siebel Upgrade Wizard” on page 38](#).

Prerequisites for Restarting the Siebel Upgrade Wizard

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

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

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

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

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

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

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

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

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

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

To manually start the Siebel Upgrade Wizard

- 1 Navigate to the following directory:

Windows: `SI EBEL_ROOT\bin`

UNIX: `$SI EBEL_ROOT/bin`

- 2 Enter the following command:

Windows: `si ebug /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`

UNIX: `srvrupgwi z /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.ucf`

where:

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

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

`si ebug /m master_upgrep_dev_704.ucf.`

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

A check mark appears beside each item as it is completed.

- 4 The Siebel Upgrade Wizard stops after the step, Export schema definition, and the following message appears:

Pause #1

Please make sure your database administrator performs operations based on the files in the DDL Output Directory to complete this step. Once the operations are completed, please select Yes. To stop now and continue upgrade later, please select No. (To resume, please start the Upgrade Wizard from the command line using `option /m master_upgrep_<env>_<path>_mf.ucf.`)

- 5 Do not click either Yes or No when the preceding message appears. Depending on whether you are upgrading from a midtier environment or a mainframe environment, continue your upgrade by completing the appropriate tasks:
 - If you are upgrading from a midtier environment, see [“Executing SQL Files After the First Pause” on page 177](#).
 - If you are upgrading from a mainframe environment, see [“Resuming the Siebel Upgrade Wizard After the First Pause” on page 199](#).

Deploying Stored Procedures and User-Defined Functions

Upgrades: All upgrades.

Environments: Development environment only.

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

You must transfer and install a user-defined function (UDF) and several stored procedures onto the Siebel Database Server to support the Siebel product. Any method that transfers the necessary files to the correct location on the Database Server is acceptable.

NOTE: When deploying stored procedures and user-defined functions, you need to update the jobcard and job parameter information to conform to your specific standards.

Perform the following tasks to transfer the stored procedure code to the zSeries host.

Prerequisites: The following prerequisites apply to creating and deploying stored procedures:

- You must have upgraded your schema to the new schema structure introduced in Release 7.7.
- You must have set up your WLM environment *before* running the stored procedures.
- You cannot create stored procedures until *after* the second pause in the development environment upgrade process.
- For development environment upgrades from the midtier, see [“Resuming the Siebel Upgrade Wizard After the Second Pause” on page 182](#).
- For development environment upgrades from the mainframe, see [“Resuming the Siebel Upgrade Wizard on the Midtier After the Second Pause” on page 222](#).

To transfer the stored procedure files to the zSeries host

- 1 Navigate to your DDL output directory and open the dev0ftp.txt file.

Windows: The default DDL output directory is `SIEBEL_ROOT\dbsrvr\db2390\dboutput\upgrade`.

UNIX: The default DDL output directory is `$SIEBEL_ROOT/dbsrvr/db2390/dboutput/upgrade`.

- 2 Edit the dev0ftp.txt file using the following information:

- a Change &IP into the IP address of your DB2 host, for example ZM01.
- b Change &Username to an authorized TSO ID.
- c Change the constant SIEBELQ1 to your own dataset high-level qualifier (*DSNHLO*), for example, SADMIN, using a command like Replace All.

Use the same high-level qualifier for all zSeries upgrade datasets. The *DSNHLO* 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 *DSNHLO* can be the TSO ID.

- 3 After modifying the dev0ftp.txt file, save it and close the file.

- 4 Double-click ftpdev0.bat on Windows (from UNIX issue the following command: `Ftp -i > ftpfile.bat`)

You are prompted to enter the password for the user name that you entered in dev0ftp.txt to replace &Username.

- 5 Enter the password, and then press Enter.

All files generated to this point of the upgrade process transfer from the midtier to the zSeries host.

To submit the stored procedure job on the DB2 host

- 1 After the transfer is successful, log on to the mainframe, and navigate to the *DSNHLO.SIEBEL.DEVO* file, for example, *SIEBLQ1.SIEBEL.DEVO*.
- 2 Edit *DSNHLO.SIEBEL.DEVO*, and then submit the job.
 - a In the job card set the following values:

```
CLASS=S
MSGCLASS=X
JOBPARAM S=YOUR_HOST_NAME
```

For example, if your Host server is QA01, the job card should look like the following:

```
#####DEVOD JOB ACCNT#, CLASS=S, MSGCLASS=X, MSGLEVEL=(1, 1),
//          TIME=1440, REGION=OM, NOTIFY=&SYSUID, LINES=9999
//*
/*JOBPARM S=QA01
```

Use uppercase letters only. As indicated in the Comments section, change all references of *SIEBELHLO* to your TSO ID or *DSNHLO* using the following command:

```
C SIEBELQ1 DSNHLO ALL
```

Where *DSNHLO* is the dataset name high-level qualifier.

- b On the command line, enter the following commands:

```
C <DB2LIB1> your_DB2_Load_DSN ALL
```

For example, C <DB2LIB1> DSN710.SDSNLOAD ALL

```
C <DB2LIB2> your_second_DB2_Load_DSN ALL
```

For example, C <DB2LIB2> DSN710.RUNLIB.LOAD ALL

```
C &TSUBSYS your_subsystem ALL
```

For example, C &TSUBSYS Q10K ALL

```
C &SCHEMAQUALIFIER your_schema_qualifier ALL
```

For example, C &SCHEMAQUALIFIER CQ10Knnn ALL

Change the <DB2WLM> parameter to your isolated WLM environment name. This must include eight characters.

For example, if your isolated WLMENV is WLMxyz1, then you should issue the following command:

```
C <DB2WLM> 'WLMxyz1 ' ALL
```

NOTE: A trailing space was added in the preceding example to make the string eight characters long.

- c Enter submit and then press Enter.
 - d After submitting the job, enter cancel on the command line or press PF3 to save changes.
 - e Verify that the job completed successfully:

- ❑ Verify that the RC=0.
 - ❑ Verify that dataset *DSNHLQ.DEV.JCL* was created.
- 3** Edit the RECEIVE member of *DSNHLQ.DEV.JCL* as appropriate, then submit the job.
- a** In the job card set the following values:
- ```
CLASS=S
MSGCLASS=X
JOBPARAM S=YOUR_HOST_NAME
```
- b** Comment the line that begins with //LIBS by replacing L with \*.
- For example, if your Host server is QA01, the job card should look like the following:
- ```
000001 //###SPRV   JOB ACCNT#, CLASS=S, MSGCLASS=X, MSGLEVEL=(1, 1),
000002 //          TIME=1440, REGION=OM, NOTIFY=&SYSUID, LINES=9999
000003 //*
000004 /*JOBPARAM S=QA01
000005 //*
000006 //*LIBS   JCLLIB ORDER=(DSNHLQ.JCLLIB)
000007 //*
```
- c** Verify that RC equals 00 to signify that the job completed successfully.
- 4** Execute a rexx exec to alter the received DDL library symbolics.
- Enter the following command from the command line:
- ```
TSO EX ' DSNHLQ.DEV.JCL(DDLMODR)'
```
- For example, if your *DSNHLQ* is XYZ123.AA, then enter the following command after you run the RECEIVE member:
- ```
TSO EX ' XYZ123.AA.DEV.JCL(DDLMODR)'
```
- 5** Edit the IBCOPY member of *DSNHLQ.DEV.JCL* as appropriate, then submit the job.
- a** In the job card, set the following values:
- ```
CLASS=S
MSGCLASS=X
JOBPARAM S=YOUR_HOST_NAME
```
- b** Comment the line that begins with //LIBS by replacing L with \*.
- c** In the line beginning //OUTPDS DD DSN=*load\_module\_dsname\_space*
- Specify the WLM environment where:
- load\_module\_dsname\_space* corresponds to your isolated WLM environment.
- For example, your isolated WLM environment is WLMxyz1:
- ```
DB2WLM.WLMxyz1.LOAD
```
- d** Verify the job card looks like the following (your Host server is QA01) and submit the job:

```
000001 //###SPRV   JOB ACCNT#, CLASS=S, MSGCLASS=X, MSGLEVEL=(1, 1),
000002 //          TIME=1440, REGION=OM, NOTIFY=&SYSUID, LINES=9999
000003 //*
000004 /*JOBPARM S=QA01
000005 //*
000006 /*LIBS   JCLLIB ORDER=(DSNHLQ.JCLLIB)
000007 /*
```

e Verify that RC equals 00 to signify that the job completed successfully.

6 Edit the BINDSIA member of *DSNHLQ.DEV.JCL*.

a In the job card, set the following values:

```
CLASS=S
MSGCLASS=X
JOBPARM S=<YOUR HOST NAME>
```

b Comment the line that begins with //LIBS by replacing L with *.

c Verify the job card looks as follows (your host server is QA01):

```
000001 //###BNDS   JOB ACCNT#, CLASS=S, MSGCLASS=X, MSGLEVEL=(1, 1),
000002 //          TIME=1440, REGION=OM, NOTIFY=&SYSUID, LINES=9999,
000003 //          USER=SBLDBA
000004 //*
000005 /*JOBPARM S=QA01
000006 //*
000007 /*LIBS   JCLLIB ORDER=(DSNHLQ.JCLLIB)
```

d Execute the following change command: to replace &SCHEMAQUALIFIER with the target schema qualifier you are using:

```
C &SCHEMAQUALIFIER TABLEOWNER_NAME ALL
```

e Execute the following change command, replacing *ASCEBC* with either ASCII or EBCDIC as appropriate:

- For ASCII databases, type the following command: C 'ASCEBC' ASCII ALL
- For EBCDIC databases, type the following command: C 'ASCEBC' EBCDIC ALL

f Execute the following change command to replace &TSUBSYS with the subsystem you are using:

```
C &TSUBSYS SUBSYSTEM ALL
```

g Submit the job. Verify that RC equals 00 to signify that the job completed successfully.

h Press PF3 to save your changes.

7 After the stored procedure creates and binds, refresh the Workload Manager (WLM) environment. To do this, run WLMREFSH.

Interpreting Error Returns

The BINDSIA job sometimes fails with an RC8 with the following error in SDSF output:

```
DSNT408I  SQLCODE = -454, ERROR:  THE SIGNATURE PROVIDED IN THE CREATE FUNCTION  
STATEMENT FOR CQ10K006.NEXTSEQ MATCHES THE SIGNATURE OF SOME OTHER FUNCTION ALREADY  
EXISTING IN THE SCHEMA
```

```
DSNT418I  SQLSTATE   = 42723 SQLSTATE RETURN CODE  
DSNT415I  SQLERRP    = DSNXCRT SQL PROCEDURE DETECTING ERROR
```

This will cause a job status return code of 8 for CREATE and 4 for BIND. Error messages of this type are acceptable. Any other message types are unacceptable and mean the job has failed.

Migrating Address Data from Custom Extension Columns

Upgrades:

- From Release 6.x Siebel Industrial applications (except Communications, Media, and Energy) that you are upgrading directly to Release 7.7 Siebel Industrial applications
- From Release 6.x and 7.0.x Siebel Financial Services applications that you are upgrading to Release 7.7 Siebel Industrial applications
- From Release 6.x and 7.7 Siebel eBusiness applications that you are upgrading to Release 7.7 Siebel Industrial applications
- From Release 6.2.1 of Siebel Financial Services applications on IBM z/OS platforms that you are upgrading to Release 7.7.x of Siebel Industrial applications on IBM z/OS platforms

NOTE: This topic does not apply to Siebel eBusiness applications (HOR) that you are upgrading to a later release of Siebel eBusiness applications (HOR).

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

If you have custom extension columns on the Address table S_ADDR_ORG, you must manually edit your upgrade scripts to prevent loss of data during an upgrade. Perform the following task, and contact Siebel Technical Support or Siebel Expert Services if you need assistance.

Prerequisite: You must have run the Database Server Configuration utility to pause # 1. This creates the files required for this procedure.

To migrate Address data from custom extension columns

- 1 In the tmptable.ctl file, modify the definitions of S_ADDR_ORG_TMP and S_ADDR_PER_TMP to include your custom extension columns from the development environment.

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

Windows: *DBSRVR_ROOT\DB2390*

UNIX: *DBSRVR_ROOT/DB2390*

For example:

[Object 2]

```
Type = Table
Name = S_ADDR_ORG_TMP
Space = USERSPACE1
Column 1 = ROW_ID VARCHAR(15) NOTNULL
Column 2 = CREATED_TIMESTAMP NOTNULL DEFAULT %NOW%
Column 3 = CREATED_BY VARCHAR(15) NOTNULL
Column 4 = LAST_UPD_TIMESTAMP NOTNULL DEFAULT %NOW%
Column 5 = LAST_UPD_BY VARCHAR(15) NOTNULL
Column 6 = DCKING_NUM NUMERIC(22, 7) DEFAULT 0
...
Column 46 = RURAL_ROUTE_NUM VARCHAR(10)
Column 47 = START_DT_TIMESTAMP
Column 48 = X_CUSTOM_EXTENSION_COLUMN1 VARCHAR(15)
Column 49 = X_CUSTOM_EXTENSION_COLUMN2 VARCHAR(15)
```

- 2 Add all custom extension columns from S_ADDR_PER and S_ADDR_ORG to the S_CON_ADDR definition.
 - For development environment upgrades, use the ddl.ctf file to add these columns
 - For production environment upgrades, use the schema.ddl file to add these columns

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

Windows: *DBSRVR_ROOT\DB2390*

UNIX: *DBSRVR_ROOT/DB2390*

- 3 In the preschm_sia.sql file, add the custom extension columns for S_ADDR_PER and S_ADDR_ORG to the migration scripts.

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

Windows: *DBSRVR_ROOT\DB2390\upgrade\version*

UNIX: *DBSRVR_ROOT/DB2390/upgrade/version*

For example:

```
insert into S_ADDR_ORG_TMP (
ROW_ID, CREATED, CREATED_BY, LAST_UPD, LAST_UPD_BY,
DCKING_NUM, MODIFICATION_NUM, CONFLICT_ID, ACTIVE_FLG, ADDR_NAME,
BL_ADDR_FLG, MAIN_ADDR_FLG, DISA_CLEANSER_FLG, NAME_LOCK_FLG, OU_ID,
SHIP_ADDR_FLG, LATITUDE, LONGITUDE, ADDR, ADDR_LINE_2,
```

```

ADDR_LINE_3, ADDR_NUM, ADDR_TYPE_CD, AREA_ID, CITY,
COMMENTS, COUNTRY, COUNTY, DESCRIPTOR, DFLT_SHIP_PRIO_CD,
EMAIL_ADDR, FAX_PH_NUM, PH_NUM, INTEGRATION_ID, INTEGRATION2_ID,
INTEGRATION3_ID, PROVINCE, STATE, TRANSPRT_ZONE_CD, ZIPCODE,
ADDR_MAIL_CD, ADDR_SUB_CD, BU_ID, END_DT, RURAL_ROUTE_NUM, START_DT
, X_CUSTOM_EXTENSION_COLUMN1 , X_CUSTOM_EXTENSION_COLUMN2 )
select ROW_ID, CREATED, CREATED_BY, LAST_UPD, LAST_UPD_BY,
DCKING_NUM, MODIFICATION_NUM, CONFLICT_ID, ACTIVE_FLG, ADDR_NAME,
BL_ADDR_FLG, MAIN_ADDR_FLG, DISACLEANSE_FLG, NAME_LOCK_FLG, OU_ID,
SHIP_ADDR_FLG, LATITUDE, LONGITUDE, ADDR, ADDR_LINE_2,
ADDR_LINE_3, ADDR_NUM, ADDR_TYPE_CD, AREA_ID, CITY,
COMMENTS, COUNTRY, COUNTY, DESCRIPTOR, DFLT_SHIP_PRIO_CD,
EMAIL_ADDR, FAX_PH_NUM, PH_NUM, INTEGRATION_ID, INTEGRATION2_ID,
INTEGRATION3_ID, PROVINCE, STATE, TRANSPRT_ZONE_CD, ZIPCODE,
ADDR_MAIL_CD, ADDR_SUB_CD, BU_ID, END_DT, RURAL_ROUTE_NUM, START_DT
, X_CUSTOM_EXTENSION_COLUMN1 , X_CUSTOM_EXTENSION_COLUMN2
from S_ADDR_ORG

```

- 4 Edit the preschm_sia.sql file to populate S_CON_ADDR from the temporary tables with the newly added custom extension columns.

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

Windows: *DBSRVR_ROOT\DB2390\upgrade\version*

UNIX: *DBSRVR_ROOT/DB2390/upgrade/version*

For example:

```

insert into S_CON_ADDR
(ROW_ID,
CREATED, CREATED_BY, LAST_UPD, LAST_UPD_BY, MODIFICATION_NUM,
CONFLICT_ID, ADDR_PER_ID, ADDR_TYPE_CD, ACCNT_ID,
EMAIL_ADDR, FAX_PH_NUM, PH_NUM, DFLT_SHIP_PRIO_CD,
TRANSPRT_ZONE_CD, BRICK_ID,
SHIP_ADDR_FLG, ADDR_MAIL_CD, BL_ADDR_FLG, END_DT, MAIN_ADDR_FLG, START_DT,
X_CUSTOM_EXTENSION_COLUMN1 , X_CUSTOM_EXTENSION_COLUMN2 )

```

```

select
s_sequence_pkg.get_next_rowid,
CREATED, CREATED_BY, LAST_UPD, LAST_UPD_BY, MODIFICATION_NUM,
CONFLICT_ID, NEW_ROW_ID, ADDR_TYPE_CD, OU_ID,
EMAIL_ADDR, FAX_PH_NUM, PH_NUM, DFLT_SHIP_PRIO_CD,
TRANSPRT_ZONE_CD , AREA_ID,
SHIP_ADDR_FLG, ADDR_MAIL_CD, BL_ADDR_FLG, END_DT, MAIN_ADDR_FLG, START_DT
, X_CUSTOM_EXTENSION_COLUMN1 , X_CUSTOM_EXTENSION_COLUMN2
FROM S_ADDR_ORG_TMP SAOT
WHERE
NOT EXISTS
        (SELECT 'x' FROM S_CON_ADDR SCA1 , S_ADDR_PER_TMP
        WHERE SCA1.ACCNT_ID = SAOT.OU_ID
        AND SCA1.ADDR_PER_ID = TMP.ROW_ID
        AND TMP.ADDR_NAME = SAOT.ADDR_NAME
        )
AND SAOT.OU_ID IS NOT NULL

```

To view the data in your custom extension columns, you must configure your application to expose the custom extension columns added to S_CON_ADDR.

- 5 Development environment upgrades only.** In Siebel Tools, add the custom extension columns defined in the previous steps to S_CON_ADDR and S_ADDR_PER.

NOTE: Confirm that the preschema_sia.jcl file includes all necessary extension columns for the data migration before ftping the file to the zSeries host.

Preparing for a No-Development-Environment Upgrade

Upgrades: All upgrades.

Environments: Production environment only.

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

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

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

To prepare for a no-development-environment upgrade

- 1 Navigate to *DBSRVR_ROOT\common* (UNIX: *DBSRVR_ROOT/common*) and locate the *mstrep.dat* file.
- 2 Copy the *mstrep.dat* file and rename it *custrep.dat*.
- 3 Place the *custrep.dat* file in the *DBSRVR_ROOT\Platform* (UNIX: *DBSRVR_ROOT/Platform*) directory, where *Platform* is the database platform, for example *DBSRVR_ROOT\DB2UDB*.
- 4 In the *Platform* directory, copy the *ddl.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 tablespace, separate from the Siebel Database. Install the Siebel Database from the new release in the new tablespace. Do not migrate any data to the new database.

This database is called the reference database.
- 7 Define an ODBC for the reference database.

Installing New License Keys After Upgrade

Upgrades: All upgrades.

Environments: All environments.

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

With the new release of Siebel eBusiness applications, you received one or more license keys. You must add all of the new license keys to enable the new release of the Siebel eBusiness applications, which you use in the next steps of the upgrade process.

- For development environment upgrades, perform this task after upgrading the Siebel Database schema (*upgprep*).
- For production environment upgrades, perform this task after upgrading the custom database schema (*Upgphys*).

To add new license keys

- 1 Start Siebel Tools (current release version) from a production workstation and log on to the Siebel Database Server as the Siebel administrator.
- 2 Add your new license keys.

12 Upgrading a Database from the Midtier

This chapter contains the following topics:

- “Executing SQL Files After the First Pause” on page 177
- “Resuming the Siebel Upgrade Wizard After the First Pause” on page 178
- “Executing SQL Output Files After the Second Pause” on page 180
- “Resuming the Siebel Upgrade Wizard After the Second Pause” on page 182
- “Executing SQL Output Files After the Third Pause” on page 184
- “Resuming the Siebel Upgrade Wizard After the Third Pause” on page 185
- “Upgrading the Physical Schema” on page 185
- “Synchronizing the Schema” on page 186
- “Transferring Dedup Files” on page 187
- “Updating Statistics” on page 187
- “Migrating Custom Business Component Configurations” on page 188

If you have not customized the Siebel eBusiness applications, and therefore do not have a development environment, proceed directly to [Chapter 10, “Preparing a Production Environment for Upgrade.”](#)

NOTE: Be aware that when performing a development environment upgrade from the midtier, there is only a *source* system name (that is, there is no *target* system name).

Executing SQL Files After the First Pause

Upgrades: All upgrades.

Environments: Development (midtier-centric) environment.

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

Execute the development output files from the DB2 command window in the following order:

- **Drop_view.sql.** This file drops all views related to the schema.
- **Droptrig.sql.** This file drops all triggers related to the schema. This file is generated at run-time.
- **Pret.sql.** This static file (consisting of five update statements) is run on the source tables so that unloads from them can load, compatible with the target tables. This affects source tables with columns that changed from NULLABLE to NOT NULL (not accepting null values).
- **Pret_sia.sql.** This file is only generated for upgrades of Siebel Industry applications.

To execute each SQL file

- 1 From the DB2 command window, go to the output directory that you specified (this is where the SQL files were generated) and type:

```
db2 connect to DATA_SOURCE user DATABASE_USER_NAME using PASSWORD
```

```
db2 set current sqlid='SCHEMA_QUALIFIER'
```

The following message appears if the command is successful:

```
DB20000I The SQL command completed successfully.
```

- 2 Open the SQL file in the specified output directory and check the SQL delimiter. This can be either ";" or "/".
- 3 Execute each SQL file, in the order in which they are listed above, by entering the command that is appropriate for the SQL delimiter.

- If the delimiter is ";" then the command to execute the sql file is:

```
Db2 -tvf SQL_FILENAME.sql -z LOG_FILENAME.log
```

Where:

SQL_FILENAME.sql = name of the SQL file

LOG_FILENAME.log = name of the log file in which the results of this command are stored. You can specify any name.

To execute *drop_view.sql* the command is:

```
Db2 -tvf drop_view.sql -z drop_view.log
```

- If the delimiter is "/" (or any character other than ";"), then the command to execute the sql file is:

```
Db2 -td/ -vf SQL_FILENAME.sql -z LOG_FILENAME.log
```

Where:

SQL_FILENAME.sql = the name of the .SQL file

LOG_FILENAME.log = the name you specify for the log file in which the results of this command are stored (you can specify any name)

To execute *schema.sql* the command is:

```
Db2 -td/ -vf schema.sql -z schema.log
```

- 4 Repeat [Step 3 on page 178](#) for each SQL file.

Resuming the Siebel Upgrade Wizard After the First Pause

Upgrades: All upgrades.

Environments: Development (midtier-centric) environment.

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

After the schema has been manually created (after you have performed the task, “[Executing SQL Files After the First Pause](#)” on page 177), continue the upgrade from the point at which it paused.

To continue upgrade of the Siebel Database schema on the workstation

- 1 Resume the development environment upgrade using the appropriate method for your operating system.

Windows:

- If the dialog box is still open from [Step 4 on page 167](#), click Yes in the dialog box.
- If you closed the Siebel Upgrade Wizard screen or it is no longer open, restart the upgrade by typing the following command at the Windows DOS command prompt from the `$IEBEL_ROOT\si_ebsrvr_root\BIN` directory:

```
si_ebug.exe /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.mf.ucf
```

To restart a development upgrade from Release 7.0.4, type the following command:

```
si_ebug.exe /m master_upgrep_dev_704.mf.ucf
```

UNIX:

- Navigate to `$IEBEL_ROOT/bin` and type the following command:

```
srvrupgwi z /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.mf.ucf
```

To restart a development environment upgrade from Release 7.0.4, type the following command:

```
srvrupgwi z /m master_upgrep_dev_704.mf.ucf
```

- 2 In the Siebel Upgrade Wizard screen, click OK to resume the upgrade.

A check mark appears beside each item as it is completed.

CAUTION: The upgrade process executes in a DOS command window. Do not click inside this window while the upgrade is running. If you accidentally click inside the DOS command window, the process pauses and the word *Select* precedes the title of the command window. To continue, press the space bar or escape.

The Siebel Upgrade Wizard stops again and the following message appears:

Pause #2

Please make sure your database administrator performs operations based on the files in the DDL Output Directory to complete this step. Once the operations are completed, please select Yes. To stop now and continue upgrade later, please select No. (To resume, please start the Upgrade Wizard from the command line using option /m master_upgrep_dev_<path>.mf.ucf.)

- 3 Do not click either Yes or No when the second pause message appears. Leave this dialog box open until after your DBA has executed the SQL output files as described in [“Executing SQL Output Files After the Second Pause”](#).

After successfully executing the SQL files and reviewing the log files for unacceptable errors, continue with the upgrade.

Executing SQL Output Files After the Second Pause

Upgrades: All upgrades.

Environments: Development (midtier-centric) environment.

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

The output files created by the Siebel Upgrade Wizard before the second pause message are generated into the default output directory or into the output directory that you specified in [Table 18 on page 160](#). You execute the output files in two stages:

■ Stage 1

In stage 1 you execute a sub-set of the output files created by the Siebel Upgrade Wizard before the second pause message. Then you deploy stored procedures and user-defined functions as described in [“Deploying Stored Procedures and User-Defined Functions” on page 167](#).

■ Stage 2

After you deploy the stored procedures and user-defined functions as described in [“Resuming the Siebel Upgrade Wizard After the Second Pause” on page 182](#), you execute the remainder of the output files created by the Siebel Upgrade Wizard before the second pause message.

Windows: The default DDL output directory is `SI EBEL_ROOT\dsrsvr\db2390\dboutput\upgrade`.

UNIX: The default DDL output directory is `$SI EBEL_ROOT/dsrsvr/db2390/dboutput/upgrade`.

Executing SQL Output Files After the Second Pause - Stage 1 of 2

This section describes stage 1 of how to execute the output files created by the Siebel Upgrade Wizard before the second pause message.

To execute the SQL output files after the second pause

- Execute the SQL output files in the following order from the DB2 command window. See [“Resuming the Siebel Upgrade Wizard After the Second Pause” on page 182](#) for execution procedures.

NOTE: You may have a subset of these files, because they are produced according to upgrade path and Siebel eBusiness application.

- schema.sql
- tmptable.sql

NOTE: You only need to run the following echannel SQL files if you are upgrading from a Siebel 6.x application.

- echannel_merge_contact.sql
- echannel_merge_org_int.sql

To create and populate the temporary tables used by echannel_merge_contact.sql and echannel_merge_org_int.sql, go to the dbsrvr\db2390 directory and run the following jobs:

- eChannelAccountDivMatch.sql
- eChannelEmpContactMatch.sql

If you are a Financial Services customer, run the following jobs:

- eChannelAccountDivMatch_fins.sql
- eChannelEmpContactMatch_fins.sql.
- echannel_merge_contact_sia.sql
- echannel_merge_org_int_sia.sql
- HHMigPop.sql

NOTE: Change the HHMigPop.sql file to reflect your site-specific storage details (such as, storage group and schemaqualifier). Also, since HHMigPop.sql creates objects that are required for Household_Mig_Fins.sql, make sure you run HHMigPop.sql *before* running Houshold_Mig_Fins.sql.

- Household_Mig_Fins.sql
- devOfftp.txt

Execute this file as part of the procedure described in [“Resuming the Siebel Upgrade Wizard After the Second Pause” on page 182.](#)

- ftpdev0.bat

Execute this file as part of the procedure described in [“Resuming the Siebel Upgrade Wizard After the Second Pause” on page 182.](#)

TIP: To stop the process after every error when executing the output files, add the parameter -s to the command. This will prevent the need to restore your database and rerun the upgrade if you encounter errors due to data dependencies in these jobs.

After executing the SQL files successfully and reviewing the log files for unacceptable errors, you deploy the stored procedures and user-defined functions as described in [“Deploying Stored Procedures and User-Defined Functions” on page 167.](#)

Executing SQL Output Files After the Second Pause - Stage 2 of 2

This section describes stage 2 of how to execute the output files created by the Siebel Upgrade Wizard before the second pause message. This section assumes that you have deployed stored procedures and user-defined functions as described in [“Resuming the Siebel Upgrade Wizard After the Second Pause” on page 182](#).

To execute the SQL output files after the second pause

- Execute the SQL output files in the following order from the DB2 command window. See [“Executing SQL Files After the First Pause” on page 177](#) for execution procedures.

NOTE: You may have a subset of these files, because they are produced according to upgrade path and Siebel eBusiness application.

- preschm.sql
- preschm_sia.sql
- prod_configurator.sql
- prod_configurator_sia.sql
- upg_iss.sql

After executing the SQL files successfully and reviewing the log files for unacceptable errors, you may continue with the upgrade.

Resuming the Siebel Upgrade Wizard After the Second Pause

Upgrades: All upgrades.

Environments: Development (midtier-centric) environment.

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

After you perform the task, [“Executing SQL Output Files After the Second Pause” on page 180](#), continue the upgrade from the point at which it paused.

CAUTION: Do not proceed unless you installed stored procedures and user-defined functions as described in [“Resuming the Siebel Upgrade Wizard After the Second Pause” on page 182](#).

To continue upgrade of the Siebel Database schema on the workstation

- 1 Resume the development environment upgrade using the appropriate method for your operating system.

Windows:

- If the dialog box is still open from [Step 2 on page 179](#), click Yes in the dialog box for pause #2.

- If you closed the Siebel Upgrade Wizard screen or it is no longer open, restart the upgrade by typing the following command at the Windows DOS command prompt from the `$IEBEL_ROOT\si_ebsrvr_root\BIN` directory:

```
si_ebugg.exe /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.mf.ucf
```

To restart a development upgrade from Release 7.0.4, type the following command:

```
si_ebugg.exe /m master_upgdev_704.mf.ucf
```

UNIX:

- Navigate to `$IEBEL_ROOT/bin` and type the following command:

```
srvrupgwiz /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.mf.ucf
```

To restart a development environment upgrade from Release 7.0.4, type the following command:

```
srvrupgwiz /m master_upgdev_704.mf.ucf
```

- 2 In the Siebel Upgrade Wizard screen, click OK to resume the upgrade.

A check mark appears beside each item as it is completed.

CAUTION: The upgrade process executes in a DOS command window. Do not click inside this window while the upgrade is running. If you accidentally click inside the DOS command window, the process pauses and the word *Select* precedes the title of the command window. To continue, press the space bar or escape.

The Siebel Upgrade Wizard stops again and the following message appears:

Pause #3.

Your database administrator must create Siebel Indexes using `sciidx.sql` located in the Upgrade Output Directory. Once the Siebel Indexes are created, please continue by selecting Yes. To stop now and resubmit the Upgrade Wizard later, please select No. (To resume, please start the Upgrade Wizard from the command line using option `/m master_upgdev_<path>.mf.ucf`.)

- 3 Do not click either Yes or No when the third pause message appears. Leave this dialog box open until after your DBA has executed the following SQL files on the zSeries host:

- `indexes.sql`
- `gen_primary1.sql`
- `gen_primary2.sql`
- `gen_primary3.sql`

- gen_primary4.sql

After successfully executing the SQL files and reviewing the log files for unacceptable errors, you may continue with the upgrade.

NOTE: When you are performing a midtier-centric development environment upgrade, the Siebel Upgrade Wizard may stop responding during the Repository Data Upgrade step. If this happens, cancel the Siebel Upgrade Wizard and from the command prompt, navigate to \$SIEBEL_ROOT\bin and execute the RUNSTATS commands on tables S_COLUMN and S_UK_ATTJOIN using the following syntax:

```
RSTAT390 /u <database_username> /p <database_password> /c <ODBC_datasource> /d  
<Tableowner> /a Y /l <logfile> /T S_COLUMN
```

```
RSTAT390 /u <database_username> /p <database_password> /c <ODBC_datasource> /d  
<Tableowner> /a Y /l <logfile> /T S_UK_ATTJOIN
```

After executing these commands, restart the Siebel Upgrade Wizard from the command prompt by entering siebug.exe /m master_upgrep_dev_<version>_mf.ucf.

Executing SQL Output Files After the Third Pause

Upgrades: All upgrades.

Environments: Development (midtier-centric) environment.

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

The output files created by the Siebel Upgrade Wizard before the third pause message are generated into the default output directory or into the output directory that you specified in [Table 18 on page 160](#).

Windows: The default DDL output directory is \$SIEBEL_ROOT\dbsrvr\db2390\dboutput\upgrade.

UNIX: The default DDL output directory is \$SIEBEL_ROOT/dbsrvr/db2390/dboutput/upgrade.

To execute the SQL output files after the third pause

- Execute the SQL output files in the following order from the DB2 command window. See [“Executing SQL Files After the First Pause” on page 177](#) for execution procedures.
 - indexes.sql
 - gen_primary1.sql
 - gen_primary2.sql
 - gen_primary3.sql
 - gen_primary4.sql

After successfully executing the SQL files and reviewing the log files for unacceptable errors, continue with the upgrade from the point at which it paused.

Resuming the Siebel Upgrade Wizard After the Third Pause

Upgrades: All upgrades.

Environments: Development (midtier-centric) environment.

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

After your DBA has performed the task, “[Executing SQL Output Files After the Third Pause](#)” on page 184 and updated statistics, you may continue the upgrade from the point at which it paused.

CAUTION: Do not proceed unless you installed stored procedures and user defined functions as described in “[Deploying Stored Procedures and User-Defined Functions](#)” on page 167.

To continue upgrade of the Siebel Database schema on the workstation

- 1 Resume the development environment upgrade using the appropriate method for your operating system.

Windows:

- If the dialog box is still open from [Step 2 on page 183](#), click Yes in the dialog box for pause #3.
- If you closed the Siebel Upgrade Wizard screen or it is no longer open, restart the upgrade by typing the following command at the Windows DOS command prompt from the `SIEBEL_ROOT\si_ebsrvr_root\BIN` directory:

```
si_ebugg.exe /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.mf.ucf
```

To restart a development upgrade from Release 7.0.4, type the following command:

```
si_ebugg.exe /m master_upgrep_dev_704.mf.ucf
```

UNIX:

- Navigate to `$SIEBEL_ROOT/bin` and type the following command:

```
srvrupgwi z /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.mf.ucf
```

To restart a development environment upgrade from Release 7.0.4, type the following command:

```
srvrupgwi z /m master_upgrep_dev_704.mf.ucf
```

- 2 In the Siebel Upgrade Wizard screen, click OK to resume the upgrade.

A check mark appears beside each item as it is completed.

Upgrading the Physical Schema

Upgrades: All upgrades.

Environments: Development (midtier-centric) environment.

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

The configuration utility that you ran in [“Preparing to Run the Database Server Configuration Utility” on page 160](#) updated the Siebel Database Schema. You must now use the Database Server Configuration Wizard to apply these changes to the physical database schema to upgrade it to the new version.

The repository on which you run the Physical Schema upgrade must be the same repository that you prepared for the merge.

To launch the Database Configuration Wizard, refer to one of the following procedures as appropriate for your operating system:

- [“Running the Database Server Configuration Utility Under Windows” on page 163](#)
- [“Running the Database Server Configuration Utility Under UNIX” on page 164](#)

Synchronizing the Schema

Upgrades: All upgrades.

Environments: Development (midtier-centric) environment.

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

The output files created by the Siebel Upgrade Wizard during the procedure [“Upgrading the Physical Schema” on page 185](#) are generated into the default output directory or into the output directory that you specified in [Table 18 on page 160](#).

Windows: The default DDL output directory is `SI EBEL_ROOT\dbsrvr\db2390\dboutput\upgrade`.

UNIX: The default DDL output directory is `$SI EBEL_ROOT/dbsrvr/db2390/dboutput/upgrade`.

To synchronize the schema

- 1 Execute the SQL output files generated by [“Running the Siebel Upgrade Wizard” on page 165](#) in the following order from the DB2 command window. See [“Executing SQL Files After the First Pause” on page 177](#) for execution procedures.
 - `Synctab.sql`
 - `Syncidx.sql`
 - `Tmpdeduptable.sql`
 - `Ddlview.sql`
- 2 After executing each SQL file, review the corresponding log file.

After successfully executing the SQL files and reviewing the log files for unacceptable errors, you may continue the upgrade from the point at which it paused.

Transferring Dedup Files

Upgrades: All upgrades.

Environments: Development (midtier-centric) environment.

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

During upgphys processing, one of the last steps (Generate dedup jcl files for 390) is generating the dedup files (dedup.jcl and dedup_prod.jcl) that you are going to use for your production environment upgrade. Once upgrep has generated these dedup files (by default, in the *DBSRVR/DB2390/DBOUTPUT/UPGRADE* directory), transfer them to the same directory on the Siebel Enterprise Server that you are going to use for your production environment upgrade.

NOTE: These dedup.jcl and dedup_prod.jcl files are similar to files that you generate during the Prepare for Production upgrade process, which is also run before starting the production environment upgrade. Since both processes use the same information when generating the files, you can use the dedup files from either the upgphys process or Prepare for Production process.

Updating Statistics

Upgrades: All upgrades.

Environments: Development (midtier-centric) environment.

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

This section describes how to update statistics. Perform this task on tables that receive a large number of inserts on the repository imports.

To update statistics

- 1 Navigate to the following directory:

SI EBEL_ROOT\si ebsrvr\bin

- 2 Execute the file RSTAT390.exe

The following table describes the parameters for this executable.

Argument	Description
/u	Username. For example, sbladm. This argument is required.
/p	Password. This argument is required.
/c	ODBC Data Source. This argument is required.
/d	Siebel Table Owner. This argument is required.
/e	Specify Y to execute RUNSTATS on entire schema. Default = N.
/t	Specify the table to RUNSTATS on.
/a	Specify Y to RUNSTATS on all tables in table space. Default = N.

Argument	Description
/i	Specify Y to ignore errors. This applies to the entire schema only. Default = N.
/l	Specify directory and filename for log file. Default filename = rstat390.log.
/s	Sample value. Default = 25.

Migrating Custom Business Component Configurations

Upgrades: All upgrades.

Environments: Development (midtier-centric) environment.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform the 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 buscomps 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 buscomps that require manual migration, review the upgcust.l log file, located in the *SIEBSRVRL* log directory.

Critical obsolete tables are listed in [Table 20](#).

Table 20. 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 buscomp migration utility is run on the Prior Customer Repository by the Siebel Upgrade Wizard during upgrade of the Siebel Database schema.

Several fields, buscomps, 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.

13 Upgrading a Database from the Mainframe

This chapter contains the following topics:

- “Transferring the Customized Repository to z/OS Host” on page 189
- “Preparing the zSeries Host Environment” on page 190
- “Resuming the Siebel Upgrade Wizard After the First Pause” on page 199
- “Transferring Control Cards and Schema DDL Files to the zSeries Host” on page 200
- “Preparing the Schema and JCL Files on the zSeries Host” on page 201
- “Installing Stored Procedures on the zSeries Host” on page 205
- “Preparing Siebel-Scheduled Jobs on the zSeries Host” on page 206
- “Optimizing the Source Database Job Stream” on page 210
- “Loading Log Tables on the Source Database” on page 210
- “Loading Log Tables on the Target Database” on page 211
- “Executing the Upgrade on the zSeries Host (Phase 1 of 2)” on page 212
- “Resuming the Siebel Upgrade Wizard on the Midtier After the Second Pause” on page 222
- “Transferring the Index Schema to the zSeries Host” on page 224
- “Preparing Siebel-Scheduled JCL” on page 225
- “Executing the Upgrade on the zSeries Host (Phase 2 of 2)” on page 228
- “Generating RUNSTATS” on page 234
- “Performing Intersection Table Maintenance” on page 234

Transferring the Customized Repository to z/OS Host

Upgrades: All upgrades.

Environments: Production environment only.

When you upgraded the custom database schema (upgphys) in your development environment, the new customized repository was exported to a file called *custrep.dat*, located in the *DBSRVR_PLATFORM_ROOT* directory on the Siebel Server client machine from which you ran the upgrade.

You must now make a copy of *custrep.dat* and transfer it from your DB2 UDB for UNIX or Windows root directory to your DB2 UDB for z/OS and OS/390 directory on your client machine.

Preparing the zSeries Host Environment

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

The results of the Siebel Upgrade Wizard are stored in the output directory that you specified in [Table 18 on page 160](#).

The first step in preparing the zSeries host environment is to transfer the output files generated by the Siebel Upgrade Wizard from the output directory on the midtier into datasets on the zSeries host. Once you have transferred these files, you need to create and populate the target datasets. The last step in preparing the zSeries host environment is to modify the populated target datasets to meet your job execution requirements. To accomplish these tasks, complete the following procedures:

- [“Prerequisites for Upgrade Procedures on the zSeries” on page 190](#)
- [“Transferring JCL, SQL, and Binary Files to the zSeries Host” on page 191](#)
- [“Defining and Allocating zSeries Setup Datasets” on page 191](#)
- [“Setting Up the Upgrade Environment on the zSeries Host” on page 193](#)
- [“Preparing for Table Creation \(PRET\) on the Source Database” on page 196](#)

Prerequisites for Upgrade Procedures on the zSeries

The user who performs upgrade procedures on the zSeries 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 zSeries host using the high-level qualifier specified in the FTP script.
- Access to the DB2 source and target system.
- Access to datasets.

To provide a DBA with access to your datasets, issue the following command on the TSO command line:

```
TSO PERMIT ** ID(SBLDBA) ACC(ALTER)
```

Where:

SBLDBA is the person to whom you want to grant the authority to alter dataset access.

NOTE: Only issue this command once.

- DB2 authorities to create DB2 objects and create DB2 VSAM datasets.
- Grant and Bind authority.
- DB2 Workload Manager refresh authority.

This user must have the following information:

- The dataset name high-level qualifier (*DSNHLO*) that you replaced SIEBELQ1 with in the FTP script. The *DSNHLO* must be less than or equal to 18 characters in length and may contain multiple nodes separated by periods. Each node can be between one and eight characters in length. A node more than eight characters long will cause the job to fail.
- The default JOBCLASS=Q. Make sure you use the correct job class in the static jobcards.

Transferring JCL, SQL, and Binary Files to the zSeries Host

The following procedure transfers necessary files to the zSeries host.

To transfer files to the zSeries host

- 1 Navigate to your DDL output directory and open the pretftp.txt file.

Windows: The default DDL output directory is *SI EBEL_ROOT\dfsrvr\db2390\dboutput\upgrade*.

UNIX: The default DDL output directory is *\$SI EBEL_ROOT/dfsrvr/db2390/dboutput/upgrade*.

- 2 Edit the pretftp.txt file using the following information:

- a Replace &IP with the IP address or name of your zSeries host, for example ZM01.
- b Replace &Username with the your TSO ID, for example sadmi n.
- c Replace the constant SIEBELQ1 with your own dataset high-level qualifier (*DSNHLO*), for example, to your TSO ID or to a *DSNHLO* that you have authority to allocate datasets as (such as SK. SBL) using a command like Replace All.

Use the same high-level qualifier for all zSeries upgrade datasets. The *DSNHLO* 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 *DSNHLO* can be the TSO ID.

- 3 After modifying the pretftp.txt file, save it, and then close the file.
- 4 Double-click pretftp.bat (from UNIX issue the following command: `Ftp -i > ftpfile.bat`).
- 5 At the prompt, enter the password for the user name that you entered in pretftp.txt to replace &Username, and then press Enter.

All files generated to this point of the upgrade process transfer from the midtier to the zSeries host.

The pretftp.log file is created in *dfsrvr\db2390\dboutput\upgrade*.

Defining and Allocating zSeries Setup Datasets

Complete the following procedure to define and allocate zSeries setup datasets.

To define and allocate zSeries setup datasets

- 1 After the transfer is successful, log on to the mainframe, and navigate to the *DSNHLO.SIEBEL.JOB0* dataset, for example, SK.SBL.SIEBEL.JOB0.
- 2 Go to Edit mode on the dataset.
- 3 Follow the procedures inside the *DSNHLO.SIEBEL.JOB0* file. Use uppercase letters only. As indicated in the Comments section, change all references of SIEBELHLQ to *DSNHLO* using the following command:

```
c SIEBELHLQ DSNHLO ALL
```

CAUTION: The variables *DSNHLO*, *SIEBELQ1*, and *SIEBELHLQ* all represent the same value, and this value *must* be the same throughout your deployment.

- 4 Submit the job using the JCL in dataset *DSNHLO.SIEBEL.JOB0*.
- 5 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 6 Verify that the job ran successfully.
 - a Verify that the RC=0.
 - b Verify that three datasets were created:
 - ❑ *DSNHLO.SIEBEL.EXEC*
 - ❑ *DSNHLO.SIEBEL.JCLLIB*
 - ❑ *DSNHLO.SIEBEL.SP.CNTL*
 - c Verify that the following PDS members contain information that you defined in the Database Server Configuration Wizard on the midtier:
 - ❑ *DSNHLO.SIEBEL.EXEC(@TBOSRC)* contains the value for the source schema qualifier.
 - ❑ *DSNHLO.SIEBEL.EXEC(@TBOTAR)* contains the value for the target schema qualifier.
- 7 Enter 6 on the TSO command line to go to the TSO Command Line Processor panel.
- 8 Enter the following command to allocate the CLIST/EXEC library *DSNHLO.SIEBEL.EXEC*:


```
al loc f(sysexec) da(DSNHLO.SIEBEL.EXEC) shr reuse
```

NOTE: To make sure that you only reference Siebel executables, 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 TSO command I SRDDN.

The EXEC library *DSNHLO.SIEBEL.EXEC* is used to execute all Rexx execs. The panel and programs are located in this library.

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 *DSNHLO.SIEBEL.EXEC* library with each new TSO logon.

Setting Up the Upgrade Environment on the zSeries Host

Complete the following procedure to set up the upgrade environment on the zSeries host.

To set up the upgrade environment on the zSeries host

- 1 Enter the following command:

```
TSO SBLUPG77
```

The Siebel Upgrade 7.7 Main Menu appears. The panel ID is SBLUPG7P. You can find the panel ID in the upper 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 Main Menu, select option 0: RECEIVE - LOAD LIBRARY AND STORED PROCEDURE XMIT FILES.
- 3 Submit the displayed job.

NOTE: The displayed job depends on your language and upgrade path.

- 4 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.SI EBEL. LOAD*
- *DSNHLO.SI EBEL. SP. SPDDL*
- *DSNHLO.SI EBEL. SP. DBRMLI B*

- 5 Verify that the job ran successfully.

a Review the output in SDSF. Verify that RC=0 and JCLTEST RC=FLUSH.

b Review the output to verify that all three datasets (and members) were received properly. If receipt was successful, you will see messages 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 ["Running the Siebel Upgrade Wizard" on page 165](#).

- 6 On the Siebel Upgrade 7.7 Main Menu, select option 1: SETUP - ENVIRONMENT AND DATASET SETUP.

The Environment/Dataset menu appears. The panel ID is SBLUTLP.

- 7 On the Environment/Dataset menu, select option 0: SET SYSTEM ENVIRONMENT VARIABLES. The System Variables Definitions screen appears. The panel ID is SBLSETVP.

- 8 On the System Variables menu (Option 0), enter the following information and then press Enter:

- **DSN high level qualifier name.** The DSN high-level qualifier name is displayed automatically.

NOTE: If the *DSNHLQ* specified in Step 2 on page 191 (job0) is greater than 18 characters, you must rerun job0 using a *DSNHLQ* less than or equal to 18 characters.

- **Item A.** The HOST/LPAR name is displayed automatically.
- **Item B.** Enter DB2 Workload Manager (WLM) name, for example, DB27WLM.
- **Item C.** Enter the DB2 CCSID (encoding) type:

- A = ASCII
- E = EBCDIC

The DB2 CCSID variable is used to modify the bind jobs which are stored in *DSNHLQ*. SIEBEL. SP. CNTL and the WLM parameter is updated in this dataset.

- **Item D.** The DB2 SUBSYS for source and target subsystems appears automatically.
- **Item E.** Enter the DB2 LOAD LIBRARIES for the source and target subsystems. (These are the libraries where the DSN, DSNTDP2, and DSNTIAUL programs are located.)

NOTE: The libraries you enter must exist (that is, they must be cataloged).

- 9 Press Enter.

Messages appear indicating that the *DSNHLQ* and DB2 load library information was written to individual PDS members in the *DSNHLQ*. SIEBEL. JCLLIB library.

- 10 At the prompt, enter the job name prefixes for all upgrade job types.

- 11 Press Enter when you see the message: . . . PRESS ENTER TO CONTINUE. . .

The JOBNAME PREFIX/PARM DEFINITIONS panel appears. The panel ID is SBLJPFXP.

- 12 Change the NOTIFY value from &SYSUID to your TSO ID or leave it as &SYSUID.

NOTE: If you want to remove the notify parameter from the job card, replace the symbolic parameter &sysuid with spaces.

- 13** For items number 1 through 16, enter a three-character job name prefix for each upgrade job type.

All job type prefixes must be specified.

The three-character prefix should be unique, 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 does not appear and cannot be modified) are defined by Siebel Systems and are unique across all upgrade jobs.

- 14** Review the job card parameters and make any necessary changes.

Verify that you are using the correct accounting, job class, and message class.

- 15** To cancel panel SBLJPFXP and return to the Environment/Dataset menu, press PF3.

- 16** Press Enter after entering the job name prefix and parameter definitions.

The JCL template files are updated. Messages indicate that the DB2 Workload Manager (WLM) was set and that all references to SIEBELQ1 (midtier parameter in JCL template files) were replaced by the *DSNHLO* that you specified in [Step 2 on page 191](#).

The final message indicates that upgrade system environment variables have been set.

- 17** Press Enter to return to the Environment/Dataset menu.

- 18** On the Environment/Dataset menu, select option 1: CREATE UPGRADE JCL LIBRARY (VSTG00nn).

You are in edit mode for dataset *DSNHLO.SIEBEL.VSTG00nn*.

NOTE: If you want to change the job card, do so at this time.

- 19** Run the job using the JCL in dataset *DSNHLO.SIEBEL.VSTG00nn*.

- 20** Verify that the job ran successfully.

a Review the output in SDSF. Verify that the RC=0 and that the JCLTEST RC=FLUSH.

b Verify that this job allocated and populated dataset *DSNHLO.SIEBEL.INSTALL.JCL*.

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

- 22** On the Environment/Dataset menu, select option 2: ALLOCATE PDS DATASETS (CREATEDS).

- 23** Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(CREATEDS)*.

- 24** Verify that the job ran successfully.

a Review the output in SDSF. Verify that the RC=0 and that the JCLTEST RC=FLUSH.

b Verify that this job allocated datasets.

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

- 26** On the Environment/Dataset menu, select option 3: BUILD/POPULATE PDS DATASETS(UNPACK01).

- 27** Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(UNPACK01)*.

- 28** Verify that the job ran successfully.

- a Review the output in SDSF. Verify that the RC=0. A RC=4 is acceptable if the dataset is empty.
- b Verify that the JCLTEST RC=FLUSH.
- 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 IEBGENER.

29 Press PF3 to return to the Siebel Upgrade Main Menu.

Preparing for Table Creation (PRET) on the Source Database

Complete the following procedures to prepare the source database to generate unload, load, and schema files:

- [“Modifying the Source Database” on page 196](#)
- [“Adding Jobcards to All PRET Jobs” on page 196](#)
- [“Generating and Dropping Interface Tables” on page 197](#)
- [“Running Table Edit Maintenance Jobs” on page 198](#)
- [“Running Table Creation Maintenance Jobs” on page 198](#)

Modifying the Source Database

Complete the following procedure to modify the source database.

To modify the source database

- 1 On the Siebel Upgrade Main Menu, select option 2: PRET UPGRADE PROCESSES.
The PRET Upgrade Processes menu appears. The panel ID is SBLPRETP.
- 2 Press PF3 to return to the Pret Upgrade Processes menu.
- 3 (Optional.) On the Pret Upgrade Process menu, select option 1: UNLOAD/BACKUP "SOURCE" PRET TABLES THAT ARE DROPPED/ALTERED - EACH PDS JOB MEMBER MUST BE RUN.

NOTE: You can use option 1 to unload selected tables prior to PRET script execution (steps 5 and 6).

Adding Jobcards to All PRET Jobs

Perform the following procedure to add jobcards to all PRET jobs for the source database.

To add jobcards to all jobs

- 1 On the Pret Upgrade Process menu, select option 2: ADD JOBCARDS TO PRET TYPE JOBS (OPTIONS 5-6).
- 2 When you are prompted to enter the type of scheduling for the Pret processes, enter 1: Siebel Scheduled or 2: Vendor Scheduled. For information about choosing a scheduler, see [“Executions of Jobs Using Siebel-Scheduled Mode or Vendor-Scheduled Mode”](#) on page 88.

CAUTION: Choose your scheduling mode carefully. Once you select a scheduling mode for PRET, you cannot change it.

This option builds jobcards for the pret and pretfins job types. 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 zSeries host, household jobcards are also built (no messages are displayed).

During the Schema/JCL (JCL-Prep) phase you will again be prompted to select Siebel- or Vendor-scheduled Mode. You can select a different scheduling mode than what you selected during the PRET process.

Generating and Dropping Interface Tables

Perform the following procedure to generate and drop interface tables from the source database.

NOTE: Perform this procedure only during the production environment upgrade. For information on generating and dropping interface tables during a development environment upgrade, see [“Preparing EIM Tables for Upgrade”](#) on page 146.

To generate and drop interface tables

- 1 On the Pret Upgrade Process menu, select option 3: GENERATE/DROP INTERFACE TABLES FROM "SOURCE" SYSTEM.

This places you in edit mode on JCL member PRETI DRP for dataset *DSNHLO.SIEBEL.INSTALL.JCL*.

- 2 Enter sub on the command line and press Enter to submit the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(PRETI DRP)*.
- 3 After submitting the job, review the output in SDSF to verify that the job ran successfully.
 - a Verify that all steps in this job contain an acceptable return code:
 - RC=0 or RC=4 indicates the job was successful
 - RC=98 indicates that this job has already been run, so there is nothing to drop. RC=98 may be acceptable, but you should verify that all EIM interface tables do not exist under the source tableowner.
 - b Verify that the JCLTEST RC=FLUSH.
- 4 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Running Table Edit Maintenance Jobs

Perform the following procedure to run the table edit maintenance jobs called PRETEDIT.

To run PRETEDIT jobs

- 1 On the Pret Upgrade Process menu, select option 4: PRETEDIT JOBS - EACH PDS MEMBER MUST BE RUN.
This places you in edit mode for the PDS dataset *DSNHLQ.SIEBEL.PRETEDIT.JCL*.
- 2 Submit each PDS member job using the JCL in dataset *DSNHLQ.SIEBEL.PRETEDIT.JCL*. Jobs from this step can run in parallel.
- 3 After all jobs are submitted, review the output in SDSF to verify that the job ran successfully.
 - All steps must contain an acceptable return code: RC=0 or RC=4.
 - The JCLTEST return code should be FLUSH.
- 4 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Running Table Creation Maintenance Jobs

Perform the following procedure to run the table creation maintenance jobs called PRET.

To run the table creation maintenance jobs

- 1 On the Siebel Upgrade PRET menu, select option 5: PRET - PRET JOBS.
This places you in edit mode for the PDS dataset *DSNHLQ.SIEBEL.INSTALL.JCL(SUBPRET)*.
NOTE: If you are upgrading a Siebel Financial Services (FINS-SIA) application, you must first complete option 5: PRET - PRET JOBS. Once you have completed this option, you need to select option 6: PRETFINS - PRETFINS JOBS. This option places you in edit mode for the PDS dataset *DSNHLQ.SIEBEL.INSTALL.JCL(SUBPRETF)*.
This job submits the first PRET job. If the first job is successful, it submits the dependent jobs. This process continues until *all* PRET jobs are run.
CAUTION: Each job must run serially due to dependencies. If a single job fails, the process stops until the failed job is corrected and run successfully.
- 2 After *all* jobs have run, review the output in SDSF to verify that the job ran successfully.
All steps must contain an acceptable return code: RC=0 or RC=8.
The JCLTEST or JCLTEST8 return code should be FLUSH.
- 3 After submitting the job, enter cancel on the command line or press PF3 to save changes.
NOTE: After successfully completing this task, continue with the upgrade on the midtier.

Resuming the Siebel Upgrade Wizard After the First Pause

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

After your DBA has applied the PRET processes on the zSeries host, (see [“Preparing the zSeries Host Environment” on page 190](#)), continue the upgrade from the midtier.

To continue upgrade of the Siebel database schema from the midtier

- 1 Resume the upgrade using the appropriate method for your operating system.

Windows:

- If the dialog box is still open from Pause #1 ([Step 4 on page 167](#)), click Yes.
- If you closed the Siebel Upgrade Wizard screen or it is no longer open, restart the upgrade by typing the following command at the DOS command prompt from the `SIEBEL_ROOT\si_ebsrvr_root\BIN` directory:

```
si_ebug.exe /m master_UPGRADEOPTION_ENVIRONMENT_VERSION_mf.ucf
```

For example, to restart a development environment upgrade from Release 7.0.4, enter the following command:

```
si_ebug.exe /m master_upgrep_dev_704_mf.ucf
```

To restart a production environment upgrade from Release 7.0.4, enter the following command:

```
si_ebug.exe /m master_upgrep_prod_704_mf.ucf
```

UNIX:

- Navigate to `$SIEBEL_ROOT/bin` and enter the following command:

```
srvrupgwi z /m master_UPGRADEOPTION_ENVIRONMENT_VERSION_mf.ucf
```

For example, to restart a development environment upgrade from Release 7.0.4, enter the following command:

```
srvrupgwi z /m master_upgrep_dev_704_mf.ucf
```

To restart a production environment upgrade from Release 7.0.4, enter the following command:

```
srvrupgwi z /m master_upgrep_prod_704_mf.ucf
```

- 2 In the Siebel Upgrade Wizard screen, click OK to resume the upgrade.

A check mark appears beside each item as it is completed.

CAUTION: The upgrade process executes in a DOS command window. Do not click inside this window while the upgrade is running. If you accidentally click inside the DOS command window, the process pauses and the word *Select* precedes the title of the command window. To continue, press the space bar or Escape.

The Siebel Upgrade Wizard stops again and the following message appears:

Pause #2

Please make sure your database administrator performs operations based on the files in the DDL Output Directory to complete this step. Once the operations are completed, please select Yes. To stop now and continue upgrade later, please select No. (To resume, please start the Upgrade Wizard from the command line using option /m master_upgrep_<env>_<path>_mf.ucf.)

- 3 Do not click either Yes or No when the second pause message appears. Leave this dialog box open until after your DBA has performed the following tasks on the zSeries host.

After the DBA successfully performs the tasks on the zSeries host, continue with the upgrade.

Transferring Control Cards and Schema DDL Files to the zSeries Host

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

The following procedure transfers control cards and schema DDL files to the zSeries host.

To transfer files to the zSeries host

- 1 Navigate to your DDL output directory and open the ftpfile.txt file.

Windows: The default DDL output directory is `SI EBEL_ROOT\dfsrvr\db2390\dboutput\upgrade`.

UNIX: The default DDL output directory is `$SI EBEL_ROOT/dfsrvr/db2390/dboutput/upgrade`.

- 2 Edit the ftpfile.txt file using the following information:

- a Replace &IP with the IP address or name of your zSeries host, for example ZM01.
- b Replace &Username with the your TSO ID, for example sadmi n.

- c Replace the variable `SI EBELQ1` with the dataset high-level qualifier (*DSNHLO*) that you used for all upgrade datasets.

CAUTION: This *DSNHLO* must match the *DSNHLO* specified in your FTP script in “Transferring JCL, SQL, and Binary Files to the zSeries Host” on page 191.

You must use the same high-level qualifier for all zSeries upgrade datasets. The *DSNHLO* 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 *DSNHLO* can be the TSO ID.

- 3 After modifying the `ftpfi le.txt` file, save it, then close the file.
- 4 To run the FTP script, locate and double-click the batch file `ftpfi le.bat` (from UNIX issue the following command: `Ftp -i > ftpfi le.bat`).
- 5 At the prompt, enter the password for the TSO ID that you entered in `ftpfi le.txt`, and then press Enter.

All files generated to this point of the upgrade process transfer from the midtier to the zSeries host.

- 6 The `ftpfi le.log` file is created in `SIEBEL_ROOT\dbsrvr\db2390\dboutput\upgrade`.

Preparing the Schema and JCL Files on the zSeries Host

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

Perform the following procedures to prepare the schema and JCL files on the zSeries:

- [“Applying DSNHLO to FTP DDL, Unload Cards, and Load Cards”](#)
- [“Unpacking the Unload and Load Control Cards and Executing the DDL” on page 202](#)

Applying *DSNHLO* to FTP DDL, Unload Cards, and Load Cards

Perform the following task to apply the dataset name high-level qualifier (*DSNHLO*) to your FTP DDL, unload cards, and load cards.

To apply *DSNHLO* to DDL and unload/load files

- 1 On the Siebel Upgrade Main Menu, select option 3: SCHEMA/JCL - BUILD DDL AND JCL-PREP FOR UPGRADE.

The SCHEMA/JCL MENU appears. The panel ID is SBLUPGP.

- 2 On the Schema/JCL Menu, select option 0 (CHANGE UNLOAD/LOAD JCL TEMPLATES - SET *DSNHLQ*), and then press Enter.

NOTE: This option must be run *first* and can be run only *once*. If you attempt to rerun this option, the following message appears: CHANGE UNLOAD/LOAD JCL TEMPLATES - SET *DSNHLQ* HAS ALREADY BEEN RUN/APPLIED. OPTION 0 CANNOT BE RERUN.

The message STAGING dataset/TEMPLATE MODIFICATIONS IN PROGRESS appears.

Additional messages that appear:

- DROP STATEMENTS WILL BE BUILT FOR OPTIONAL USE:
 - BUILDING - DROP DATABASE STATEMENTS
 - BUILDING - DROP TABLESPACE STATEMENTS
 - BUILDING - DROP TABLE STATEMENTS
 - BUILDING - DROP INDEX (UNIQUE ONLY) STATEMENTS
 - BUILDING - DROP INDEX (NON-UNIQUE ONLY) STATEMENTS
 - BUILDING - DROP INDEX (OBSOLETE ONLY) STATEMENTS
- UNLOAD/LOAD DEFAULT *DSNHLQ* SUCCESSFULLY CHANGED TO: *DSNHLQ* ; PROCEED TO THE PDS/SCHEMA/VERIFICATION UTILITIES PANEL; RUN OPTION 1: BUILD/POPULATE UNLOAD/LOAD PDS DATASETS; (UNPACK02) ; MESSAGE: SBLSETV-0007

- 3 Press Enter to return to the Schema/JCL Menu.

The Schema/JCL Menu appears.

Unpacking the Unload and Load Control Cards and Executing the DDL

Complete the following procedure to unpack the unload and load cards and execute the DDL.

To unpack, unload/load, and execute the DDL

- 1 On the Siebel Upgrade Main Menu, select option 3: SCHEMA/JCL - BUILD DDL AND JCL-PREP FOR UPGRADE.

The SCHEMA/JCL MENU appears. The panel ID is SBLUPGP.

- 2 On the Schema/JCL Menu, select option 1: PDS/SCHEMA/VERIFICATION UTILITIES.

The Utilities Menu appears.

- 3 On the Utilities Menu, select option 0: BUILD/POPULATE UNLOAD/LOAD PDS DATASETS (UNPACK02).

This places you in edit mode for PDS dataset member: *DSNHLQ.SIEBEL.INSTALL.JCL(UNPACK02)*.

- 4 Run the job using the JCL in dataset: *DSNHLO.SIEBEL.INSTALL.JCL(UNPACK02)*.

This job populates (unpacks, using IEBUPDTE) the unload and load control cards into members in their corresponding PDS dataset. The DDL files are output (using IEBGENER) to sequential files.

- 5 Verify that the job ran successfully, RC=0.

The JCLTEST return code should be FLUSH.

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

- 7 After the batch unpack02 (option 0) job has run successfully, select option 1: CUSTOMIZE UNLOAD/LOAD FOR TABLES W/CLOB COLUMNS (CLOBCOPY).

If the source system has no CLOB columns on any tables, one of two message types appears:

- A message box that indicates that no CLOB tables exist.
- UTL0033: CLOB CUSTOMIZATION NOT REQUIRED FOR *upgtype* UPGRADE PATH.

If CLOBs exist on the source 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)
```

- 8 Submit the job and verify that it ran successfully, RC=0.

- 9 On the Utilities Menu, select option 2: CREATE DB2 SCHEMA (DB, TABLESPACE, TABLE, UNIQUE INDEXES, VIEW).

This places you in edit mode for PDS dataset member: *DSNHLO.SIEBEL.INSTALL.JCL(SCHEMA01)*.

NOTE: You need to verify the DDL before running the following job.

- 10 Run the job using the JCL in dataset: *DSNHLO.SIEBEL.INSTALL.JCL(SCHEMA01)*.

- 11 Verify that the job ran successfully, RC=0.

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

NOTE: Verify that the DDL creation (the job submitted in Step 9) was successful before continuing with the upgrade.

- 13 Select option 3: CREATE TEMP/WORK TABLE SCHEMA (3 SETS: TMPTAB00, 01 AND 02).

A message that describes the process for creating the Temp-Tables appears. This process involves being placed in edit mode three times on different PDS TMPTAB nn members in the INSTALL.JCL library:

- TMPTAB00
- TMPTAB01

■ TMPTAB02

You receive an edit-mode message each time this process occurs. After accepting the message, run a batch job.

The messages have the following format:

1. YOU WILL NOW BE PLACED IN EDIT MODE FOR TMPTAB00 . . . PRESS ENTER TO CONTINUE.

14 At the edit-mode message for TMPTAB00, press Enter.

This places you in edit mode for PDS dataset and member
DSNHLO.SI EBEL. I NSTALL. JCL(TMPTAB00).

a Run this job using the JCL in dataset *DSNHLO.SI EBEL. I NSTALL. JCL(TMPTAB00)*.

This job creates COMMON temp tables used by upgrade data migration SQL scripts.

b Verify that the job ran successfully, RC=0.

The JCLTEST return code should be FLUSH.

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

15 At the edit-mode message for TMPTAB01, do the following:

a Press Enter.

This places you in edit mode for PDS dataset and member:
DSNHLO.SI EBEL. I NSTALL. JCL(TMPTAB01).

b Run this job using the JCL in dataset *DSNHLO.SI EBEL. I NSTALL. JCL(TMPTAB01)*.

This job creates a log table used by the upgrade process to log Unload jobs on the source subsystem. This table is used *only* if you are using Siebel Logging and Scheduling.

c Verify that the job ran successfully, RC=0.

The JCLTEST return code should be FLUSH.

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

16 At the edit-mode message for TMPTAB02, do the following:

a Press Enter.

This places you in edit mode for PDS dataset and member:
DSNHLO.SI EBEL. I NSTALL. JCL(TMPTAB02).

b Run this job using the JCL in dataset *DSNHLO.SI EBEL. I NSTALL. JCL(TMPTAB02)*.

This job creates a log table used by the upgrade process to log selected target subsystem jobs. This table is used *only* if you are using Siebel Logging and Scheduling.

c Verify that the job ran successfully, RC=0.

The JCLTEST return code should be FLUSH.

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

Installing Stored Procedures on the zSeries Host

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

Perform the following task to unpack the unload and load cards and execute the DDL.

To install and verify stored procedures

- 1 On the Siebel Upgrade Main Menu, select option 3: SCHEMA/JCL - BUILD DDL AND JCL-PREP FOR UPGRADE.

The SCHEMA/JCL MENU appears. The panel ID is SBLUPGP.

- 2 On the Schema/JCL Menu, select option 1: PDS/SCHEMA/VERIFICATION UTILITIES.

The Utilities Menu appears. The panel ID is SBLUTL2P.

- 3 On the Utilities Menu, select option 4: INSTALL/DEPLOY STORED PROCEDURES AND UDF.

This places you in edit mode for PDS dataset *DSNHLO.SI EBEL.SP.CNTL*.

- 4 Select the *DSNHLO.SI EBEL.SP.CNTL(@README)* PDS member.

This places you in edit mode for PDS dataset and member *DSNHLO.SI EBEL.SP.CNTL(@README)*.

- 5 Review the @README file.

- 6 Select the IEBCOPY PDS member.

This places you in edit mode for PDS dataset and member *DSNHLO.SI EBEL.SP.CNTL(IEBCOPY)*.

- 7 Follow the procedures in the JCL.

The WLM library to be used will vary based on the DB2 subsystem being used. You must replace *LOAD_MODULE_DSNAME_SPAS* with your WLM load library name.

This job moves Stored Procedure load modules to the WLMSPAS.

NOTE: Do *not* change the @DB2WLM member. This value was set on the SBLSETV panel as part of the environment/dataset setup process.

- 8 Verify that the job ran successfully, RC=0.

The JCLTEST return code should be FLUSH.

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

- 9 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 SP 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.

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

The JCLTEST return code should be FLUSH.

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

- 12** Select option 5: RUN STORED PROCEDURE INSTALL VERIFICATION.

This places you in edit mode for one of the following PDS datasets and members:

- For Siebel eBusiness application upgrades: *DSNHLO.SIEBEL.INSTALL.JCL(SPVHOR)*
- For upgrades of Siebel Industry applications: *DSNHLO.SIEBEL.INSTALL.JCL(SPVSIA)*

- 13** Run required JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SPVSIA)*.

This job executes each SP against an empty target schema. This process is to verify that the SPs have been installed and can be executed.

- 14** Verify that the job ran successfully, RC=0.

The JCLTEST return code should be FLUSH.

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

- 16** On the Utilities Menu, Press PF3.

The Schema/JCL Menu for your upgrade path appears. The panel ID is SBLUPGP.

You install the stored procedures and UDFs to facilitate upgrade processing. You can view the SQL in the dataset *DSNHLO.SIEBEL.SP.SPDDL*.

Preparing Siebel-Scheduled Jobs on the zSeries Host

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

NOTE: The procedure in this section uses Siebel-Scheduled job execution. For information about choosing a scheduler, see ["Executions of Jobs Using Siebel-Scheduled Mode or Vendor-Scheduled Mode" on page 88](#). Choose your scheduling method carefully, because once you begin an upgrade process under a selected mode, you cannot change your scheduling mode or reverse this decision.

To complete JCL preparation (Part 1)

- 1 On the Siebel Upgrade Main Menu, select option 3: SCHEMA/JCL - BUILD DDL AND JCL-PREP FOR UPGRADE.

The SCHEMA/JCL MENU appears. The panel ID is SBLUPGP.

- 2 On the Schema/JCL Menu, select option 2: SIEBEL SCHEDULED - JCL-PREP TO BUILD/EXECUTE/MANAGE JOBS VIA SIEBEL.

The JCL-PREP: Siebel Logging/Scheduling Menu appears. The panel ID is SBLJCL1P.

- 3 Select option 1: ADD JOBCARDS, BUILD SIEBEL JOB LOGGING CAPABILITIES.

A message appears before each part of the JCL preparation runs. The following messages are examples of the messages you receive.

NOTE: The messages you receive depend on your upgrade path.

```
SIEBEL JCL MODIFICATIONS TO BE APPLIED BASED ON RULES THAT SIEBEL LOGGING/SCHEDULING
WILL BE USED. MESSAGE: SBLJCL-0012.
```

```
JOB LOG LOADFILE BEING WRITTEN TO: DSN= DSNHLQ.SIEBEL.JOBLOG.LOADFILE.
```

```
UNLOAD JCL - ADDING JOBCARDS/LOGGING DSN= DSNHLQ.SIEBEL.DDLIMP.UNLOAD.CNTL.
```

```
LOAD JCL - ADDING JOBCARDS/LOGGING DSN= DSNHLQ.SIEBEL.DDLIMP.LOAD.CNTL.
```

```
DATA MI GR-PRESCHM JCL - ADDING JOBCARDS/LOGGING DSN= DSNHLQ.SIEBEL.PRESCHM.JCL.
```

```
DATA MI GR-PRDCFG JCL - ADDING JOBCARDS/LOGGING DSN= DSNHLQ.SIEBEL.PRDCFG.JCL.
```

```
DATA MI GR-GEN PRIMARY JCL - ADDING JOBCARDS/LOGGING DSN= DSNHLQ.SIEBEL.GENPRIM.JCL.
```

```
DATA MI GR-UPGISS JCL - ADDING JOBCARDS/LOGGING DSN= DSNHLQ.SIEBEL.UPGISS.JCL.
```

```
PROCESSING COMPLETE. . .
```

```
***
```

When three asterisks (***) appear, the process is complete.

- 4 Press Enter.

The JCL-PREP: SIEBEL LOGGING/SCHEDULING Menu appears. The panel ID is SBLUPGP.

- 5 Select option 2: ADD SIEBEL JOB LOGGING, JCL COND-CODE CHECKS.

A series of messages displays as each part of the JCL preparation runs:

NOTE: The messages you receive depend on your upgrade path.

- SIEBEL JCL MODIFICATIONS TO BE APPLIED BASED ON RULES THAT SIEBEL LOGGING/SCHEDULING WILL BE USED. MESSAGE: SBLJCL-0012.

- MODIFY UNLOAD JCL - ADDING SIEBEL LOGGING DSN= DSNHLQ.SIEBEL.DDLIMP.UNLOAD.CNTL.

- MODIFY LOAD JCL - ADDING SIEBEL LOGGING DSN= DSNHLQ.SIEBEL.DDLIMP.LOAD.CNTL.

- MODIFY DATA MI GR-PRESCHM JCL - ADDING SIEBEL LOGGING DSN= DSNHLQ.SIEBEL.PRESCHM.JCL.

- MODIFY DATA MIGR-GEN PRIMARY JCL - ADDING SIEBEL LOGGING DSN=
DSNHLQ.SIEBEL.GENPRIM.JCL.
- PROCESSING COMPLETE. . .

When three asterisks (***) appear, the process is complete.

6 Press Enter.

The JCL-PREP: SIEBEL LOGGING/SCHEDULING Menu appears. The panel ID is SBLUPGP.

7 Select option 3: ADD JOBSTEP TO EACH DATA MIGRATION JOBSTREAM THAT AUTO-SUBMITS DEPENDENT JOB TO JES INTRDR. DEPENDENT JOB SUBMITTED ONLY WHEN PREDECESSOR JOB RAN SUCCESSFULLY.

A series of messages displays as each part of the JCL preparation runs:

NOTE: The messages you receive depend on your upgrade path.

SIEBEL JCL MODIFICATIONS TO BE APPLIED BASED ON RULES THAT SIEBEL LOGGING/SCHEDULING WILL BE USED. MESSAGE: SBLJCL-0012.

PLEASE NOTE: THE FOLLOWING CHANGES WILL BE APPLIED TO SOURCE TABLE UNLOAD AND DATA MIGRATION JCL. (EXAMPLES: PRESCHM, PRDCFG, UPGISS).

MODIFY DATA MIGR-PRESCHM JCL - ADD DEPENDENT JOB SUBMIT JCL DSN=
DSNHLQ.SIEBEL.PRESCHM.JCL.

PROCESSING COMPLETE. . .

NOTE: The type and number of messages you receive depends on your upgrade.

8 Press Enter.

The JCL-PREP: SIEBEL LOGGING/SCHEDULING Menu appears. The panel ID is SBLUPGP.

9 On the Schema/JCL Menu for your upgrade path, select option 2: SIEBEL SCHEDULED - JCL-PREP TO BUILD/EXECUTE/MANAGE JOBS VIA SIEBEL.

The Schema/JCL Menu for your upgrade path appears. The panel ID is SBLUPGP.

10 Select option 4: GENERATE INDEX REBUILDS.

The Index Menu appears. The panel ID is SBLIDXP.

11 Select option 1: NON-UNIQUE INDEX.

The following message appears:

SIEBEL JCL MODIFICATIONS TO BE APPLIED BASED ON RULES THAT SIEBEL LOGGING/SCHEDULING WILL BE USED. MESSAGE: SBLJCL-0014.

12 When the following message appears, specify the number of indexes to be included in each rebuild job:

ENTER THE MAXIMUM NUMBER OF INDEXES TO BE INCLUDED IN A SINGLE REBUILD JOB. (PRESS ENTER TO ACCEPT DEFAULT OF 3, MAXIMUM IS 10)

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.

During the Non-Unique Index Rebuilds process, the following series of message appears as the process runs.

```
GENERATING INDEX REBUILD JOBS FOR NON-UNIQUE INDEXES OLD-SCHEMA BUILDING NON-UNIQUE-  
INDEX REBUILDS (DDLNUIND) DSN= DSNHLQ.SIEBEL.DDLNUIND.JCL  
DSN= DSNHLQ.SIEBEL.DDLNUIND.SQL.
```

```
NUMBER OF INDEX REBUILD JOBS BUILT = 951.
```

Index count varies according to your upgrade path and the number of indexes included in each job.

```
GENERATING INDEX REBUILD JOBS FOR OBSOLETE INDEXES OLD-SCHEMA - BUILDING OBSOLETE-  
INDEX REBUILDS (DDLIND) DSN= DSNHLQ.SIEBEL.DDLIND.JCL  
DSN= DSNHLQ.SIEBEL.DDLIND.SQL.
```

```
NUMBER OF INDEX REBUILD JOBS BUILT = 101.
```

Index count varies according to your upgrade path and the number of indexes included in each job.

The preceding example used TSO foreground mode. The procedure can be performed in either batch or foreground mode.

NOTE: It is recommended that you perform the procedure in batch mode. Performing the procedure in batch mode executes options 1-3 on the foreground screen.

- 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: If you perform the procedure in foreground mode, make sure your logon region size is at least 7092.

- To perform the procedure in batch mode, tab to the screen location named: RUN OPTIONS 1-3 IN BATCH? N (Y/N). Replace the N with a Y.

NOTE: Before performing the procedure in batch mode, make sure you modify the ispf batch procedure, *dsnhlq.siebel.proc(ispbat)*, to your installation standards.

The options on the JCL-PREP: Siebel Logging/Scheduling Menu *must* be performed in the order in which they are listed. For example, you must complete step 1 before performing step 2, and so on.

Optimizing the Source Database Job Stream

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

To improve performance and reduce the number of jobs managed in a one-table-per-tablespace (including partitioned tablespaces) database configuration, you can use an optional feature called Job Stream Optimization (JSO) to remove unload, load, and index rebuild jobs for tables that contain no data.

CAUTION: Do not perform this task if your environment has multiple tables per tablespace.

NOTE: This process is irreversible unless you make a copy of your datasets before deleting jobs that act on tables containing no data.

The jobs are located in `DSNHLQ.SIEBEL.INSTALL.JCL`.

To remove unload, load, and index rebuild jobs for tables that contain no data

1 After preparing Siebel-Scheduled jobs on the zSeries host, verify the following JCL members:

- JSOSRCQ
- JSOSRCD

These members are new in Release 7.7, and substitutions were applied to them on the midtier for source and target DB2 subsystems and tableowners.

2 Run the JCL member JSOSRCQ.

This member produces a source candidate list of tables that are empty.

3 Review the SDSF output for unload, load, and old schema index rebuild jobs (non-unique and obsolete).

4 Run the JCL member JSOSRCD.

This member physically deletes the JCL for every empty table, unload, load, and index rebuild jobs. The unload and load control cards are not deleted.

Loading Log Tables on the Source Database

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

Perform the following procedure to load log tables on the source database.

To load the Source Log table

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:
TSO SBLUPG77
- 2 Select option 4: RUN UPGRADE / POST-UPGRADE MAINTENANCE.
The Siebel Upgrade Job Submission Menu for your upgrade path appears.
 - For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
 - For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.
- 3 Select option 0: LOAD SOURCE SIEBEL LOG.
This loads the source Siebel log table using the *dsnhlq.siebel.joblog.loadfile*.
- 4 Verify that the job class is appropriate for the source subsystem.
- 5 Run the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(LSRCLOG)*.
NOTE: After running this job, verify that it was successful by reviewing the sysout in SDSF and making sure the RC=0.
- 6 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Loading Log Tables on the Target Database

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

Perform the following steps to load log tables on the target database.

To load the Target Log table

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:
TSO SBLUPG77
- 2 Select option 4: RUN UPGRADE / POST-UPGRADE MAINTENANCE.
The Siebel Upgrade Job Submission Menu for your upgrade path appears.
 - For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
 - For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.
- 3 Select option 1: LOAD TARGET SIEBEL LOG (1 OF 2).

- 4 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(LTARLOG1)*.
This loads the target Siebel log table using the *dsnhlq.siebel.joblog.loadfile*.

NOTE: After running this job, verify that it was successful by reviewing the sysout in SDSF and making sure the RC=0.

- 5 Review sysout output, rows loaded will vary, depending on your upgrade path.
- 6 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Executing the Upgrade on the zSeries Host (Phase 1 of 2)

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

To execute the upgrade process on the zSeries host, complete the following procedures:

- ["Running Unload Jobs on the Source Database" on page 212](#)
- ["Running Load Jobs on the Target Database" on page 213](#)
- ["Executing Index DDL and Executing Rebuild Indexes" on page 214](#)
- (Optional.) ["Migrating eChannel Data" on page 215](#)
- (Optional.) ["Migrating Household Data" on page 216](#)
- ["Migrating Preschm Data" on page 217](#)
- ["Migrating Address Data" on page 218](#)
- ["Migrating Product Configurator Data" on page 219](#)
- ["Migrating ISS Data" on page 221](#)

Siebel Financial Services Upgrades

Complete the following procedures if you are upgrading a Siebel Financial Services application:

- ["Migrating Preschm Data for FINS" on page 218](#)
- ["Migrating Product Configurator Data for FINS" on page 220](#)

Running Unload Jobs on the Source Database

Perform the following procedure to run unload jobs on the source database.

To run unload jobs on the source database

- 1** If you are not on the Siebel Upgrade Main Menu, enter the following command:
TSO SBLUPG77
- 2** Select option 4: RUN UPGRADE / POST-UPGRADE MAINTENANCE.
The Siebel Upgrade Job Submission Menu for your upgrade path appears.
 - For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
 - For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.
- 3** Select option 2: UNLOAD SOURCE.
This places you in edit mode for dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBUNLD)*.
- 4** Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBUNLD)*.
This job automatically submits *all* the unload job streams referenced in dataset *DSNHLO.SIEBEL.EXEC(@DDLIMPU)*. The number of Unload jobs varies according to upgrade path.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the source Siebel log table: *SOURCE_TABLEOWNER.TMP_SBLLOG_SRC*.
- 5** Verify that the job ran successfully, RC=0 or RC=4.
Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).
- 6** After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 7** View the job status log.

If you selected Siebel-Scheduled Mode, you can view the job status log by completing the procedure described in [“Viewing the Siebel Job Log Status” on page 240](#).

Running Load Jobs on the Target Database

Perform the following procedure to run load jobs on the target database.

To run loads to the target database

- 1** If you are not on the Siebel Upgrade Main Menu, enter the following command:
TSO SBLUPG77
- 2** Select option 4: RUN UPGRADE / POST-UPGRADE MAINTENANCE.
The Siebel Upgrade Job Submission Menu for your upgrade path appears.
 - For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
 - For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.

3 Select option 3: LOAD TARGET.

This places you in edit mode for dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBLOAD)*.

4 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBLOAD)*.

This job automatically submits *all* the unload job streams referenced in dataset *DSNHLO.SIEBEL.EXEC(@DDLIMPL)*.

This submits the Load job streams. The number of upgrade Load jobs varies by upgrade path. If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: *TARGET_TABLEOWNER.TMP_SBLLOG_TAR*.

5 Verify job ran successfully, RC=0 or RC=4.

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see ["Running the Siebel Upgrade Wizard" on page 165](#).

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

7 View the job status.

For information on how to view job status if you selected Siebel-Scheduled Mode, see ["Viewing the Siebel Job Log Status" on page 240](#).

Once all Load jobs have completed successfully, continue with the upgrade.

Executing Index DDL and Executing Rebuild Indexes

Perform the following procedure to execute index DDL and execute rebuild indexes.

To execute the index DDL and rebuild indexes

1 If you are not on the Siebel Upgrade Main Menu, enter the following command:

```
TSO SBLUPG77
```

2 Select option 4: RUN UPGRADE / POST-UPGRADE MAINTENANCE.

The Siebel Upgrade Job Submission Menu for your upgrade path appears.

■ For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.

■ For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.

3 Select option 4: "OLD SCHEMA INDEX DDL".

This places you in edit mode for dataset *DSNHLO.SIEBEL.INSTALL.JCL(SCHEMA02)*.

4 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SCHEMA02)*.

This job runs DDL Create Index statements to build the old-schema non-unique and obsolete indexes.

5 Verify that the job ran successfully, RC=0 or RC=4 in steps DDLNUIND and DDLOIND.

- 6 After submitting the job, enter `cancel` on the command line or press PF3 to save changes.
The Siebel Upgrade Job Submission Menu for your upgrade path appears. The panel ID is SBLRUNFP.
- 7 Select option 5: "OLD SCHEMA INDEX REBUILDS".
This places you in edit mode for dataset `DSNHLQ.SIEBEL.INSTALL.JCL(SUBIX)`.
- 8 Run the job using the JCL in dataset `DSNHLQ.SIEBEL.INSTALL.JCL(SUBIX)`.
This job automatically submits *all* the Non-Unique and Obsolete Index Rebuild jobs referenced in datasets: `DSNHLQ.SIEBEL.EXEC(@DDLNUIX)` and `DSNHLQ.SIEBEL.EXEC(@DDL0IX)`.
Non-unique job names range from `###N0001` to `###Nnnnn`. Obsolete job names range from `###O0001` to `###Onnnn`, where the placeholder `###` represents your job prefix and `nnnn` represents a sequential number from 0001 to *n*.
- 9 Verify that the job ran successfully, RC=0 or RC=4.
Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.
If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: `TARGET_TABLEOWNER.TMP_SBLLOG_TAR`.
NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see ["Running the Siebel Upgrade Wizard" on page 165](#).
- 10 After submitting the job, enter `cancel` on the command line or press PF3 to save changes.
- 11 View the job status.
For information on how to view job status if you selected Siebel-Scheduled Mode, see ["Viewing the Siebel Job Log Status" on page 240](#).

Migrating eChannel Data

Complete the following procedure to run corresponding eChannel SQL scripts against the target system.

NOTE: This topic applies to upgrades from Release SIA 6.2.1 applications. Perform this procedure *only* if you ran corresponding eChannel scripts on the source system during the development environment upgrade. (See ["Executing SQL Output Files After the Second Pause" on page 180](#).)

To run the data migration eChannel SQL scripts

- 1 Locate the `DSNHLQ.SIEBEL.INSTALL.JCL` library in your dataset list.
This is a PDS, so all PDS members are listed.

- 2 At the command line, locate PDS members that begin with ECHM by entering the following command:

```
L ECHM
```

The following PDS members appear in the list:

```
ECHMCON
```

```
ECHMORG
```

- 3 Enter E next to the PDS members that apply to your upgrade path.

This places you in edit mode on individual PDS members.

- 4 Run this job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(ECHM*)*.

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

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

NOTE: These jobs are not logged. You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

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

This places you in edit mode on the next PDS member.

Migrating Household Data

Complete the following procedure to run corresponding Household SQL scripts against the target system.

NOTE: This topic applies to upgrades from SIA 6.2.1, 7.0.4, and 7.5.2 applications. Perform this task *only* if you ran the corresponding Household script (Household_Mig_Fins.sql) on the source system during the development environment upgrade. (See [“Executing SQL Output Files After the Second Pause” on page 180](#).)

To run the data migration Household SQL scripts

- 1 Locate the *DSNHLQ.SIEBEL.INSTALL.JCL* library in your dataset list.

This is a PDS, so all PDS members are listed.

- 2 At the command line, locate PDS members that begin with HHMIG by using the following command:

```
L HHMIGFIN
```

- 3 Edit the HHMIGFIN member of *DSNHLQ.SIEBEL.INSTALL.JCL*.

This places you in edit mode on PDS member *DSNHLQ.SIEBEL.INSTALL.JCL(HHMIGFIN)*.

- 4 Run the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(HHMIGFIN)*.

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

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

NOTE: These jobs are not logged. You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

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

Migrating Preschm Data

Complete the following procedure to migrate preschm data by executing preschm jobs. The preschm jobs update columns, populate tables, set flags, and create temporary indexes for data migration.

To migrate preschm data

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:

```
TSO SBLUPG77
```

- 2 Select option 4: RUN UPGRADE / POST-UPGRADE MAINTENANCE.

The Siebel Upgrade Job Submission Menu for your upgrade path appears.

- For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
- For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.

- 3 Select option 6: PRESCHM.

This places you in edit mode for dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBPSH)*.

- 4 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBPSH)*.

NOTE: This job automatically submits the *first* PRESCHM job referenced in dataset *DSNHLO.SIEBEL.EXEC(@PRESCHM)*.

- If this job runs successfully, it automatically submits the next dependent job, and so on.
- This process continues for *all* preschm jobs. *If a job fails, the succeeding dependent job does not get submitted.* The automatic job submission sequence terminates.

- 5 After *all* preschm jobs are run, verify that each job ran successfully.

RC=0, RC=4, or RC=8 indicates that the job was successful. RC=8 is associated with jobs that have an X in position 3 of the member name.

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: *TARGET_TABLEOWNER.TMP_SBLLOG_TAR*.

NOTE: These jobs are not logged. You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

- 6 Enter cancel on the command line or press PF3 to save changes.
- 7 View the job status.

For information on how to view job status if you selected Siebel-Scheduled Mode, see [“Viewing the Siebel Job Log Status” on page 240](#).

Migrating Address Data

The Address Migration job corresponds to scripts run previously on the midtier against the Source system. This means that you *only* need to run the following 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.

NOTE: This topic applies to upgrades from SIA 6.2.1 or FINS 7.0.4 (*not* SIS 7.0.4) applications.

To run address data migration SQL scripts

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:
TSO SBLUPG77
- 2 Select option 4: RUN UPGRADE / POST-UPGRADE MAINTENANCE.
The panel ID is SBLRUNFP.
- 3 Select option 7: ADDRESS MIGRATION.
This places you in edit mode for dataset *DSNHLO.SIEBEL.INSTALL.JCL(ADMIN)*.
- 4 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(ADMIN)*.
- 5 Submit the job and then press Enter.
- 6 Verify that the job ran successfully, RC=0 or RC=4.
Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.
- 7 After submitting the job, enter cancel on the command line or press PF3 to save changes.

Migrating Preschm Data for FINS

Complete the following procedure to migrate Preschm data for Siebel Financial Services (FINS) applications.

Prerequisite task: [“Migrating Preschm Data” on page 217](#).

NOTE: This topic applies to upgrades from Siebel Financial Services Industry 6.2.1, 7.0.4, 7.5.2, and 7.7 applications.

To migrate preschm data for FINS applications

- 1** If you are not on the Siebel Upgrade Main Menu, enter the following command:
TSO SBLUPG77
- 2** Select option 4: RUN UPGRADE / POST-UPGRADE MAINTENANCE.
The panel ID is SBLRUNFP.
- 3** Select option 8: PRESCHM-FINS.
This places you in edit mode for dataset *DSNHLO.SI EBEL. I NSTALL. JCL(SUBPSF)*.
- 4** Run the job using the JCL in dataset *DSNHLO.SI EBEL. I NSTALL. JCL(SUBPSF)*. This job automatically submits the *first* PRESCHM job referenced in dataset *DSNHLO.SI EBEL. EXEC(@PRESCHF)*.
 - If this job runs successfully, it automatically submits the next dependent job, and so on.
 - This process continues for *all* PRESCHM-FINS jobs. *If a job fails, the succeeding dependent job does not get submitted.* The automatic job submission sequence terminates.
- 5** After *all* PRESCHM FINS jobs have run, verify that each job ran successfully.
RC=0, RC=4, or RC=8 indicates that the job was successful. RC=8 is associated with jobs that have an X in position 3 of the member name.

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: *TARGET_TABLEOWNER.TMP_SBLLOG_TAR*.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).
- 6** After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 7** View the job status.

For information on how to view job status if you selected Siebel-Scheduled Mode, see [“Viewing the Siebel Job Log Status” on page 240](#).

Migrating Product Configurator Data

Complete the following procedure to migrate Product Configurator data.

NOTE: This topic applies to upgrades from SIA 6.2.1 applications.

To migrate Product Configurator data

- 1** If you are not on the Siebel Upgrade Main Menu, enter the following command:
TSO SBLUPG77

- 2 Select option 4: RUN UPGRADE / POST-UPGRADE MAINTENANCE.

The panel ID is SBLRUNFP.

- 3 Select option 9: PROD CONFIG.

This places you in edit mode for dataset *DSNHLQ.SIEBEL.INSTALL.JCL(SUBPC)*.

- 4 Run the job using the JCL in dataset *DSNHLQ.SIEBEL.INSTALL.JCL(SUBPC)*.

This job automatically submits the *first* PRODCFG job stream. If this job runs successfully, it automatically submits the next dependent job, and so on. This process continues for *all* PRODCFG jobs. *If a job fails, the succeeding dependent job does not get submitted.* The automatic job submission sequence terminates.

- 5 After the job is run, verify that the job ran successfully.

RC=0, RC=4, or RC=8 indicates that the job was successful. RC=8 is associated with jobs that have an X in position 3 of the member name.

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, each job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: *TARGET_TABLEOWNER.TMP_SBLLOG_TAR*.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

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

- 7 View the job status.

For information on how to view job status if you selected Siebel-Scheduled Mode, see [“Viewing the Siebel Job Log Status” on page 240](#).

Migrating Product Configurator Data for FINS

Complete the following procedure to migrate Product Configurator data for Siebel Financial Services (FINS) applications.

Prerequisite task: [“Migrating Product Configurator Data” on page 219](#).

NOTE: This topic applies to upgrades from Siebel Financial Services Industry (FINS) 6.2.1 and 7.0.4 (*not* SIS 7.0.4) applications.

To migrate Product Configurator data for FINS applications

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:

```
TSO SBLUPG77
```

- 2 Select option 4: RUN UPGRADE / POST-UPGRADE MAINTENANCE.

The Siebel Upgrade Job Submission Menu for your upgrade path appears.

The panel ID is SBLRUNFP.

- 3 Select option 10: PROD CONFIG-FINS.

This places you in edit mode for dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBPCF)*.

- 4 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBPCF)*.

This job automatically submits the *first* PRODCFGF job stream. If this job runs successfully, it automatically submits the next dependent job, and so on. This process continues for *all* PRODCFGF jobs. *If a job fails, the succeeding dependent job does not get submitted.* The automatic job submission sequence terminates.

- 5 After the job is run, verify that the job ran successfully.

RC=0, RC=4, or RC=8 indicates that the job was successful. RC=8 is associated with jobs that have an X in position 3 of the member name.

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: *TARGET_TABLEOWNER.TMP_SBLLOG_TAR*.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

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

- 7 View the job status.

For information on how to view job status if you selected Siebel-Scheduled Mode, see [“Viewing the Siebel Job Log Status” on page 240](#).

Migrating ISS Data

Complete the following procedure to migrate ISS data.

Prerequisite: [“Migrating Product Configurator Data for FINS” on page 220](#)

To migrate UPGISS data

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:

```
TSO SBLUPG77
```

- 2 Select option 4: RUN UPGRADE / POST-UPGRADE MAINTENANCE.

The Siebel Upgrade Job Submission Menu for your upgrade path appears.

- For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
- For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.

- 3 Select the option for the application you are upgrading.

- For Siebel eBusiness application upgrades, select option 8: UPGISS.

- For Siebel Industry application upgrades, select option 11: UPGISS.

This places you in edit mode for dataset `DSNHLO.SIEBEL.INSTALL.JCL(SUBUPGIS)`.

- 4 Run the job using the JCL in dataset `DSNHLO.SIEBEL.INSTALL.JCL(SUBUPGIS)`.

This job automatically submits the *first* UPGISS job stream. If this job runs successfully, it automatically submits the next dependent job, and so on. This process continues for *all* UPGISS jobs. *If a job fails, the succeeding dependent job does not get submitted.* The automatic job submission sequence terminates.

- 5 After the job is run, verify that the job ran successfully.

RC=0, RC=4, or RC=8 indicates that the job was successful. RC=8 is associated with jobs that have an X in position 3 of the member name.

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: `TARGET_TABLEOWNER.TMP_SBLLOG_TAR`.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

- 6 After submitting the job, enter cancel on the command line or press PF3 to save changes.
- 7 View the job status.

For information on how to view job status if you selected Siebel-Scheduled Mode, see [“Viewing the Siebel Job Log Status” on page 240](#).

Resuming the Siebel Upgrade Wizard on the Midtier After the Second Pause

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

Once you have completed the task, [“Executing the Upgrade on the zSeries Host \(Phase 1 of 2\)” on page 212](#), continue the upgrade from the point at which it paused.

CAUTION: Do not proceed unless you installed stored procedures and user defined functions as described in [“Installing Stored Procedures on the zSeries Host” on page 205](#).

To resume the Siebel Upgrade Wizard on the midtier after the second pause

- 1 Resume the upgrade using the appropriate method for your operating system.

Windows:

- If the dialog box is still open from [Step 2 on page 200](#), click Yes in the dialog box for Pause #2.

- If you closed the Siebel Upgrade Wizard screen or it is no longer open, restart the upgrade by typing the following command at the Windows DOS command prompt from the `SIEBEL_ROOT\si_ebsrvr_root\BIN` directory:

```
si_ebugg.exe /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.mf.ucf
```

For example, to restart a development environment upgrade from Release 7.0.4, enter the following command:

```
si_ebugg.exe /m master_upgrep_dev_704.mf.ucf
```

To restart a production environment upgrade from Release 7.0.4, enter the following command:

```
si_ebugg.exe /m master_upgrep_prod_704.mf.ucf
```

UNIX:

- Navigate to `$SIEBEL_ROOT/bin` and enter the following command:

```
srvrupgwiz /m master_UPGRADEOPTION_ENVIRONMENT_VERSION.mf.ucf
```

For example, to restart a development environment upgrade from Release 7.0.4, enter the following command:

```
srvrupgwiz /m master_upgrep_dev_704.mf.ucf
```

To restart a production environment upgrade from Release 7.0.4, enter the following command:

```
srvrupgwiz /m master_upgrep_prod_704.mf.ucf
```

- 2 In the Siebel Upgrade Wizard screen, click OK to resume the upgrade.

A check mark appears beside each item as it is completed.

CAUTION: The upgrade process executes in a DOS command window. Do not click inside this window while the upgrade is running. If you accidentally click inside the DOS command window, the process pauses and the word *Select* precedes the title of the command window. To continue, press the space bar or Escape.

The Siebel Upgrade Wizard stops again and the following message appears:

Pause #3

Please create Siebel Indexes using `sciidx.sql` located in the DDL Output Directory. Once the Siebel Indexes are created, please select Yes. To stop now and continue upgrade later, please select No. (To resume, please start the Upgrade Wizard from the command line using option `/m master_upgrep_<env>_<path>.mf.ucf`.)

- 3 Do not click either Yes or No when the third pause message appears. Leave this dialog box open until after you have performed the following tasks on the zSeries host.

Once you complete the tasks on the zSeries host, continue with the upgrade.

NOTE: When you are performing a Mainframe-centric development environment upgrade, the Siebel Upgrade Wizard may stop responding during the Repository Data Upgrade step. If this happens, cancel the Siebel Upgrade Wizard and from the command prompt, navigate to \$Siebsrvr\Bin and execute the RUNSTATS commands on tables S_COLUMN and S_UK_ATTJOIN using the following syntax:

```
RSTAT390 /u <database_username> /p <database_password> /c <ODBC_datasource> /d  
<Tableowner> /a Y /l <logfile> /T S_COLUMN
```

```
RSTAT390 /u <database_username> /p <database_password> /c <ODBC_datasource> /d  
<Tableowner> /a Y /l <logfile> /T S_UK_ATTJOIN
```

After executing these commands, restart the Siebel Upgrade Wizard from the command prompt by entering siebug.exe /m master_upgrep_dev_<version>_mf.ucf.

Transferring the Index Schema to the zSeries Host

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

The following procedure transfers the index schema to the zSeries host.

To transfer the index schema to the zSeries host

- 1 Go to the `SIEBEL_ROOT\DBSRVR\DB2390\DBOUTPUT\UPGRADE` directory.
 - a Open the `ftpsci.ndx.txt` file.
 - b Replace `&IP` with the IP address or name of your zSeries host, for example `ZM01`.
 - c Replace `&Username` with your TSO, for example `sadmi n`.
 - d Replace the constant `SIEBELQ1` with the dataset high-level qualifier (*DSNHLQ*) to be used for all upgrade datasets. The *DSNHLQ* must be 18 characters or less and can have multiple nodes, separated by a period. A node must be 1-8 characters in length.

The *DSNHLQ* can be the TSO ID.

NOTE: The *DSNHLQ* you use *must* match the *DSNHLQ* you used in [Step 2 on page 191](#).
 - e After modifying the `ftpsci.ndx.txt` file, save it, and then close the file.
- 2 To run the FTP script, double-click the `ftpsci.ndx.bat` file.

- At the prompt, enter the password for the TSO ID specified in [Step 1](#), and then press Enter. This transfers all files generated to this point in the process from the midtier to the zSeries host:

Midtier File Name	zSeries Host Staging Dataset Name
SCI NDX. SQL	DSNHLO.SIEBEL.VSTG0110
dedup. j cl	DSNHLO.SIEBEL.VSTG0080
dedup_prod. j cl	DSNHLO.SIEBEL.VSTG0081

- Review the ftpsci ndx. log file to verify that all files transferred successfully.
- Run the Database Configuration Wizard.

Preparing Siebel-Scheduled JCL

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

Perform the following procedure to prepare the 7.7 index DDL, build index rebuild jobs, and build drop statements for old schema indexes.

NOTE: The following procedure applies *only* if you are using the Siebel-Scheduled Mode for executing jobs.

To prepare Siebel-scheduled JCL

- If you are not on the Siebel Upgrade Main Menu, enter the following command:
TSO SBLUPG77
- Select option 3: SCHEMA/JCL - BUILD DDL AND JCL-PREP FOR UPGRADE.
The Schema/JCL Menu appears. The panel ID is SBLUPGP.
- Select option 2: SIEBEL SCHEDULED - JCL-PREP TO BUILD/EXECUTE/MANAGE JOBS VIA SIEBEL.
- Select option 4: GENERATE INDEX REBUILDS. The Index Menu appears. The panel ID is SBLIDXP.
- The example text uses the TSO foreground mode (either batch or foreground mode can be used).

NOTE: It is recommended that you perform the procedure in batch mode.

Run option 2: GEN PRIMARY INDEX REBUILDS in batch mode by entering 2 in the option field, then tabbing to the RUN OPTION 1-OR-2 IN BATCH? N (Y/N) screen. Replace the N with a Y.

6 Select option 2: GEN PRIMARY INDEX REBUILDS.

A series of messages are displayed beginning with the following:

```
==> SPLIT AND BUILD "SCI NDX" INDEX DDL FILES:
DROP INDEXES      : DSN=DSNHLQ. SIEBEL. SCI NDX. GPDRPIX. DDL
EIM               : DSN=DSNHLQ. SIEBEL. SCI NDX. EIM. DDL
PRE-GEN_PRIMARY  : DSN=DSNHLQ. SIEBEL. SCI NDX. GPPRE. DDL
POST-GEN_PRIMARY : DSN=DSNHLQ. SIEBEL. SCI NDX. GPPST. DDL
```

Messages continued:

```
==> JCL TO RUN DROP/CREATE INDEX "DDL" WRITTEN TO:
DROP INDEXES      : DSN=DSNHLQ. SIEBEL. INSTALL. JCL(@GPDRPIX)
CREATE INDEXES    : DSN=DSNHLQ. SIEBEL. INSTALL. JCL(@EIMIX)
DSN=DSNHLQ. SIEBEL. INSTALL. JCL(@GPPREIX)
```

The Split and Build runs next.

The following message appears:

```
NOTE: FOLLOWING THE MESSAGES BELOW IN APPROX 15 SECONDS, AN "ENTER" PROMPT WILL
APPEAR REQUIRING YOUR ENTRY. . .
```

These additional messages appear:

```
DROP STATEMENTS WILL BE BUILT FOR OPTIONAL USE:
BUILDING - DROP DATABASE STATEMENTS
BUILDING - DROP TABLESPACE STATEMENTS
BUILDING - DROP TABLE STATEMENTS
BUILDING - DROP INDEX (UNIQUE ONLY) STATEMENTS
BUILDING - DROP INDEX (NON-UNIQUE ONLY) STATEMENTS
BUILDING - DROP INDEX (OBSOLETE ONLY) STATEMENTS
```

7 At the following prompt, enter the maximum number of indexes for a single job.

```
ENTER THE MAXIMUM NUMBER OF INDEXES TO BE INCLUDED IN A SINGLE REBUILD JOB. (PRESS
ENTER TO ACCEPT DEFAULT OF 3, MAXIMUM IS 10)
```

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.

The JCL generation for the Gen Primary Index Rebuilds runs approximately 40 minutes. The following messages appear as the generation runs:

```

*** => GENERATING INDEX REBUILD JOBS FOR INDEXES TO BE REBUILT "****BEFORE****"
GEN_PRIMARY DATA MIGRATION IS RUN

"NEW-SCHEMA" - SIEBEL 7.7 BUILDING "PRE"-GEN_PRIMARY INDEX REBUILDS (GPPREIX)

DSN=DSNHLO.SIEBEL.SCI NDX.GPPRE.JCL

DSN=DSNHLO.SIEBEL.SCI NDX.GPPRE.SQL
    
```

The number of generated index rebuilds appears.

The following message appears:

```

NUMBER OF INDEX REBUILD JOBS BUILT = 2226

PROCESSING COMPLETE. . .
    
```

NOTE: Index count varies according to your upgrade path and the number of indexes included in each job.

- 8 Press Enter when three asterisks (***) appear.
The Index Menu reappears. The panel ID is SBLIDXP.

Optimizing the Target Database Job Stream

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

CAUTION: Do not perform this task if your environment has multiple tables per tablespace.

To improve performance and reduce the number of jobs managed in a one-table-per-tablespace (including partitioned tablespaces) database configuration, you can use an optional feature called Job Stream Optimization (JSO) to remove gen primary index rebuild jobs for tables that contain no data.

NOTE: This process is irreversible unless you make a copy of your datasets before deleting jobs that act on tables containing no data.

The jobs are located in *DSNHLO.SIEBEL.INSTALL.JCL*.

To remove gen primary index rebuild jobs for tables that contain no data

- 1 After transferring the index schema to the zSeries host, verify the following JCL members:
 - JSOTARQ
 - JSOTARD

These members are new in Release 7.7, and substitutions were applied to them on the midtier for source and target DB2 subsystems and tableowners.

- 2 Run the JCL member JSOTARQ.
This member produces a target candidate list of tables that are empty.
- 3 Review the SDSF output for pre gen primary index rebuilds (non-unique and obsolete).
- 4 Run the JCL member JSOTARD.
This member deletes the PDS members and modifies the submission executable.

Executing the Upgrade on the zSeries Host (Phase 2 of 2)

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

To execute the upgrade process on the zSeries, complete the following procedures:

- ["Loading Target Siebel Log Tables" on page 228](#)
- ["Dropping Old Indexes and Creating New Indexes" on page 229](#)
- ["Rebuilding the Gen Primary 7.7 Indexes" on page 231](#)
- ["Migrating the Gen Primary 7.7 SQL" on page 232](#)
- ["Creating New EIM Indexes" on page 233](#)

Loading Target Siebel Log Tables

Perform the following procedure to load target Siebel log tables. This job loads logging records for the new schema indexes (GEN PRIMARY indexes).

To load target Siebel log tables

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:
TSO SBLUPG77
- 2 Select option 4: UPGRADE - RUN UPGRADE/POST-UPGRADE MAINTENANCE.
The Siebel Upgrade Job Submission Menu appears.
 - For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
 - For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.
- 3 Select the option for the application you are upgrading.
 - For Siebel eBusiness application upgrades, select option 9: LOAD TARGET LOG (2 of 2).

- For Siebel Industry application (SIA) upgrades, select option 12: LOAD TARGET LOG (2 of 2). This places you in edit mode for dataset `DSNHLO.SIEBEL.INSTALL.JCL(LTARLOG2)`.
 - 4 Run the job using the JCL in dataset `DSNHLO.SIEBEL.INSTALL.JCL(LTARLOG2)`.
After the job is run, verify that the job ran successfully. (RC=0 or RC=4 indicates that the job was successful.)
 - 5 After submitting the job, enter `cancel` on the command line or press PF3 to save changes.
 - 6 Load the target Siebel log table using logload file `DSNHLO.SIEBEL.JOBLOG.LOADFILE`.
 - 7 Verify that the job completed successfully (RC=0 or RC=4).
Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: `TARGET_TABLEOWNER.TMP_SBLLOG_TAR`.
- NOTE:** You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard”](#) on page 165.

Dropping Old Indexes and Creating New Indexes

Perform the following procedure to drop old schema indexes and create new schema indexes.

To drop old index schemas and create new index schemas

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:
`TSO SBLUPG77`
- 2 Select option 4: UPGRADE - RUN UPGRADE/POST-UPGRADE MAINTENANCE.
The Siebel Upgrade Job Submission Menu appears.
 - For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
 - For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.
- 3 Select the option for the application you are upgrading.
 - For Siebel eBusiness application upgrades, select option 10: OLD SCHEMA DROP INDEX.
 - For Siebel Industry application (SIA) upgrades, select option 13: OLD SCHEMA DROP INDEX.This places you in edit mode for dataset `DSNHLO.SIEBEL.INSTALL.JCL(@GPDRPIX)`.
- 4 Run the job using the JCL in dataset `DSNHLO.SIEBEL.INSTALL.JCL(@GPDRPIX)`.
NOTE: After running this job, verify that it was successful by reviewing the sysout in SDSF and making sure the RC=0.
- 5 After submitting the job, enter `cancel` on the command line or press PF3 to save changes.

- 6** Verify that the JCLTEST return code is FLUSH (RC=0 or RC=4).

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: `TARGET_TABLEOWNER.TMP_SBLLOG_TAR`.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

This job runs DDL to drop old-schema non-unique indexes.

- 7** After the `###DRPIX` job has *successfully completed*, run the PRE-GEN PRIMARY INDEX DDL job.

- 8** Make sure you are at the Siebel Upgrade Job Submission Menu.

- For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
- For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.

- 9** Select the option for the application you are upgrading.

- For Siebel eBusiness application upgrades, select option 11: PRE-GEN PRIMARY INDEX DDL.
- For Siebel Industry application (SIA) upgrades, select option 14: PRE-GEN PRIMARY INDEX DDL.

This places you in edit mode for dataset `DSNHLQ.SIEBEL.INSTALL.JCL(@GPPREIX)`.

- 10** Run the job using the JCL in dataset `DSNHLQ.SIEBEL.INSTALL.JCL(@GPPREIX)`.

After the job is run, verify that the job ran successfully. (RC=0 or RC=4 indicates that the job was successful.)

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: `TARGET_TABLEOWNER.TMP_SBLLOG_TAR`.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

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

- 12** Verify that the JCLTEST return code is FLUSH (RC=0 or RC=4).

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: `TARGET_TABLEOWNER.TMP_SBLLOG_TAR`.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

This job runs DDL to create new-schema non-unique indexes.

Rebuilding the Gen Primary 7.7 Indexes

Perform the following task to submit the rebuild jobs for the 7.7 Gen Primary indexes.

To rebuild a primary index

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:

TSO SBLUPG77

- 2 Select option 4: UPGRADE - RUN UPGRADE/POST-UPGRADE MAINTENANCE.

The Siebel Upgrade Job Submission Menu appears.

- For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
- For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.

- 3 Select the option for the application you are upgrading.

- For Siebel eBusiness application upgrades, select option 12: PRE-GEN PRIMARY INDEX REBUILDS.
- For Siebel Industry application (SIA) upgrades, select option 15: PRE-GEN PRIMARY INDEX REBUILDS.

This places you in edit mode for dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBGPRES)*.

- 4 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBGPRES)*.

After the job is run, verify that the job ran successfully. (RC=0 or RC=4 indicate that the job was successful.)

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: *TARGET_TABLEOWNER.TMP_SBLLOG_TAR*.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see ["Running the Siebel Upgrade Wizard" on page 165](#).

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

This submits 2000+ Index Rebuild jobs. The number of Index Rebuild jobs varies by upgrade type. Job names range from ###10001 to ###1nnnn.

- 6 Verify that the job ran successfully (RC=0 or RC=4).

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: *TARGET_TABLEOWNER.TMP_SBLLOG_TAR*.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see ["Running the Siebel Upgrade Wizard" on page 165](#).

Migrating the Gen Primary 7.7 SQL

Perform the following procedure to run the data migration scripts for Gen Primary.

To migrate the gen primary index SQL

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:

TSO SBLUPG77

- 2 Select option 4: UPGRADE - RUN UPGRADE/POST-UPGRADE MAINTENANCE.

The Siebel Upgrade Job Submission Menu appears.

- For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
- For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.

- 3 Select the option for the application you are upgrading.

- For Siebel eBusiness application upgrades, select option 13: GEN PRIMARY.
- For Siebel Industry application (SIA) upgrades, select option 16: GEN PRIMARY.

This places you in edit mode for dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBGENP)*.

- 4 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(SUBGENP)*.

After the job is run, verify that the job ran successfully.

RC=0, RC=4, or RC=8 indicates that the job was successful. RC=8 is associated with jobs with an X in position 3 of the member name.

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: *TARGET_TABLEOWNER.TMP_SBLLOG_TAR*.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

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

This submits 45 GenPrim job streams (the number of job streams varies by upgrade type). All 45 jobs can run in parallel—no dependencies exist.

RC=0, RC=4, or RC=8 indicates that the job was successful. RC=8 is associated with jobs with an “X” in position 3 of the member name.

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: *TARGET_TABLEOWNER.TMP_SBLLOG_TAR*.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

- 6 View the job status.

For information on how to view job status if you selected Siebel-scheduled mode, see [“Viewing the Siebel Job Log Status” on page 240](#).

- 7 After all GENPRIM jobs have completed successfully, run the POST-GEN PRIMARY INDEX DDL job.

Creating New EIM Indexes

Perform the following procedure to create the new EIM indexes.

NOTE: This topic applies to Siebel Industry applications upgrades.

To create a new EIM index

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:

```
TSO SBLUPG77
```

- 2 Select option 4: UPGRADE - RUN UPGRADE/POST-UPGRADE MAINTENANCE.

The Siebel Upgrade Job Submission Menu appears.

- For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
- For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.

- 3 Select the option for the application you are upgrading.

- For Siebel eBusiness application upgrades, select option 16: EIM INDEX DDL.
- For Siebel Industry application (SIA) upgrades, select option 19: EIM INDEX DDL.

This places you in edit mode for dataset *DSNHLO.SIEBEL.INSTALL.JCL(@EIMIX)*.

- 4 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(@EIMIX)*.

- 5 Verify that the job ran successfully.

RC=0 indicates that the job was successful.

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: *TARGET_TABLEOWNER.TMP_SBLLOG_TAR*.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

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

- 7 Verify that the JCLTEST return code is FLUSH (RC=0 or RC=4).

This job runs DDL to Create New-Schema (Siebel 7.7) EIM Indexes (Unique and Non-Unique).

Generating RUNSTATS

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

The following procedure generates RUNSTATS jobs for tablespaces that contain multiple-tables-per-tablespace or partitioned tablespaces and have *not* had statistics collected during any of the previous upgrade processes.

NOTE: This process excludes *all* interface tables (EIM and tables with an `_IF` suffix).

To generate RUNSTATS jobs

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:
TSO SBLUPG77
- 2 Select option 4: UPGRADE - RUN UPGRADE/POST-UPGRADE MAINTENANCE.
The Siebel Upgrade Job Submission Menu appears.
 - For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
 - For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.
- 3 Select the appropriate menu option for your upgrade path for intersection table maintenance:
 - For Siebel eBusiness application upgrades, select option 17: RUNSTATS (SCHEMA W/O STATS).
 - For Siebel Industry application upgrades, selection option 20: RUNSTATS (SCHEMA W/O STATS).
- 4 To start generating RUNSTATS jobs, press Enter.

Performing Intersection Table Maintenance

Upgrades: All upgrades.

Environments: Production environment only.

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

The intersection table maintenance jobs (ITM) identify and resolve potential duplicate row IDs from intersection tables.

To run the intersection table maintenance process

- 1 If you are not on the Siebel Upgrade Main Menu, enter the following command:
TSO SBLUPG77

- 2 Select option 4: UPGRADE - RUN UPGRADE/POST-UPGRADE MAINTENANCE.

The Siebel Upgrade Job Submission Menu appears.

- For Siebel eBusiness application upgrades, the panel ID is SBLRUNHP.
- For Siebel Industry application (SIA) upgrades, the panel ID is SBLRUNFP.

- 3 Select the appropriate menu option for your upgrade path for intersection table maintenance:

- For Siebel eBusiness application upgrades, select option 18: INTERSECTION TABLE MAINT.
- For Siebel Industry application upgrades, selection option 21: INTERSECTION TABLE MAINT.

- 4 On the ITM Maintenance Process Menu (Panel ID SBLITMP), select option 0: CHANGE JCL TEMPLATES - SET *DSNHLO* (REQUIRED), and then press Enter.

- 5 On the ITM Maintenance Process Menu (Panel ID SBLITMP), select option 1: BUILD/POPULATE PDS DATASETS (UNPACK03), and then press Enter.

This places you in edit mode for dataset *DSNHLO.SIEBEL.INSTALL.JCL(UNPACK03)*.

- 6 Run the job using the JCL in dataset *DSNHLO.SIEBEL.INSTALL.JCL(UNPACK03)*.

This job populates (unpacks) all ITM JCL and SQL PDS members into their corresponding PDS datasets.

```
DSNHLO.SIEBEL.ITM.SQL
```

```
DSNHLO.SIEBEL.ITM.JCL
```

- 7 Verify that the job ran successfully.

RC=0 indicates that the job was successful.

Verify that the JCLTEST return code is FLUSH.

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: *TARGET_TABLEOWNER.TMP_SBLLOG_TAR*.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

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

- 9 On the ITM Maintenance Process Menu (the panel ID is SBLITMP), select option 2: ADD JOBCARDS TO INTERSECTION TABLE MAINTENANCE (ITM) JOBS, and then press Enter.

- 10 At the prompt, enter a three-character job name prefix, and then press Enter.

NOTE: The job name prefix is used for the ITM JCL generated for the Update and Delete jobs.

A message indicates that the JCL generation is in process, then a series of messages indicates the number of jobs generated. The number of jobs varies by upgrade path.

- 11 When three asterisks (***) appear at the bottom of the display messages, press Enter to return to the ITM Maintenance Process Menu.

The jobcards are added to the JCL templates.

- 12** On the ITM Maintenance Process Menu (the panel ID is SBLITMP), select option 4: RUN UNLOAD JOBS, and then press Enter.

This places you in edit mode for dataset `DSNHLQ.SIEBEL.INSTALL.JCL(SUBITMUL)`.

TIP: If you want to run the ITM Unload jobs manually, select option 3 to view the PDS list of all unload jobs. Option 4 submits all ITM Unload jobs in a single job submission.

- 13** To run the job using the JCL in dataset `DSNHLQ.SIEBEL.INSTALL.JCL(SUBITMUL)`, enter submit on the command line and then press Enter.

This job automatically submits all the unload jobs references in datasets `DSNHLQ.SIEBEL.EXEC(@ITMUL)`.

- 14** Verify that this job (and *all* jobs submitted automatically) ran successfully.

RC=0 or RC=4 indicates that the job was successful.

Verify that the JCLTEST return code is FLUSH.

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: `TARGET_TABLEOWNER.TMP_SBLLOG_TAR`.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

TIP: To review a list of all jobs submitted automatically, view datasets `DSNHLQ.SIEBEL.EXEC(@ITMUL)`. This corresponds to all PDS members listed in option 3.

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

- 16** On the ITM Maintenance Process Menu (the panel ID is SBLITMP), select option 5: RUN DELETE/UPDATE JOBS USING PDS LIST, and then press Enter.

This places you in edit mode for dataset `DSNHLQ.SIEBEL.ITMJOBS.JCL`. This is a PDS, so all PDS members are listed.

NOTE: If a message indicates that no maintenance jobs were generated, either option 4 was not run or no tables require maintenance.

- 17** To run each PDS member job using the JCL in dataset `DSNHLQ.SIEBEL.ITMJOBS.JCL`, select a PDS member, enter submit on the command line and press Enter. Repeat for each PDS member.

All jobs can run in parallel (that is, at the same time).

18 Verify that this job (and *all* jobs submitted automatically) ran successfully.

RC=0 or RC=4 indicates that the job was successful.

Verify that the JCLTEST return code is FLUSH.

Jobs with an ABEND (end abnormally) U99 or U099 message are jobs that failed.

If you selected Siebel-Scheduled Mode, this job is logged (as *Completed Successfully* or *Failed*) to the target Siebel log table: `TARGET_TABLEOWNER.TMP_SBLLOG_TAR`.

NOTE: You must fix any failed jobs before proceeding with the upgrade. For information on restarting failed jobs, see [“Running the Siebel Upgrade Wizard” on page 165](#).

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

14 Reviewing the Database Upgrade Log Files

This chapter contains the following topics:

- “About the Database Upgrade Logs” on page 239
- “Viewing the Siebel Job Log Status” on page 240
- “Reviewing Upgrade Log Files for Errors” on page 241
- “Manually Archiving Upgrade Log Files” on page 243
- “Reviewing Log Files for Errors in Jobs Run from the Midtier” on page 243
- “Running SQL in Siebel Logs” on page 243

About the Database Upgrade Logs

Upgrades: All upgrades.

Environments: All environments.

The Siebel Upgrade Wizard produces all log files for a given process in one directory.

Windows: `SIEBEL_ROOT\log\PROCESS`

UNIX: `$SIEBEL_ROOT/log/PROCESS`

where `PROCESS` defaults to `UPGRADEOPTIMON_ENVIRONMENT_VERSION`.

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

Windows: `SIEBEL_ROOT\log\upgprep_dev_704_mf`

UNIX: `$SIEBEL_ROOT/log/upgprep_dev_704_mf`

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

Windows: `SIEBEL_ROOT\log\upgprep_prod_704_mf`

UNIX: `$SIEBEL_ROOT/log/upgprep_prod_704_mf`

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

The log file directory contains the following files and subdirectories:

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

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

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

- **Summary.** Directory that contains the source files for summary.html.
- **Summary.html.** Browser-based file that summarizes the log files in the output directory.
- **Summary.txt.** Text file that summarizes the log files in the output directory.

NOTE: For UNIX, if a browser is not installed on the machine you are using, review `summary.txt` instead of `summary.html`.

- **Summary.xml.** File produced during production upgrades and used by the Siebel Upgrade Tuner to parallelize table creation and index creation and inactivate SQLs that affect no rows. (Development upgrades do not produce a `summary.xml` file.)
- **Upgtuner_ftp_get.txt (UNIX Only).** File produced during production upgrades. You can use this file to transfer the upgrade scripts from your UNIX machine to a Windows temporary directory to perform upgrade tuning. (Development upgrades do not produce `upgtuner_ftp_get.txt`.)
- **Upgtuner_ftp_put.txt (UNIX Only).** File produced during production upgrades. After you tune your production upgrade scripts, you can use this file to transfer the upgrade scripts from the temporary directory on your Windows machine back to your UNIX environment. (Development upgrades do not produce `upgtuner_ftp_put.txt`.)

Viewing the Siebel Job Log Status

Upgrades: All upgrades.

Environments: Production environment only.

If you are using Siebel-scheduling to run your upgrade jobs, you can query the Siebel job log 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 Upgrade Main Menu, enter the following command:
TSO SBLUPG77
- 2 Select option 6: JOB LOG - DISPLAY JOB STATUS (SIEBEL SCHEDULED MODE)
The SIEBEL JOB LOG QUERY panel displays. The panel ID is SBLLQGP.
A list of successful and failed jobs appears.

- a Next to the label LIST BY JOB TYPE/STATUS:, enter 2 or 9 for FAILED jobs. Enter option 1 to list the UNLOADS jobs.

The list displays 250 lines only. The Unload, Load and Index Rebuild jobs have more than 250, so you must query using another option or by specific or partial job name.

- 3 Press PF3 when you are finished viewing the log.

The Siebel Upgrade Main Menu for your upgrade path appears.

Reviewing 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 the Upgrade.”](#)

Do *not* proceed with the upgrade until unacceptable errors have been corrected. If you cannot correct the error, contact Siebel Technical Support or Professional Services to report the error in detail. Certain errors, such as a failure to create indexes, can result in performance problems or anomalous behavior in Siebel eBusiness applications.

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

If you encountered an error during the upgrade, review log files named UpgWi z. l og (Windows) or srvrupgwi z1. l og (UNIX). The log file is located in the following directory:

Windows: `SI EBEL_ROOT\l og\PROCESS\output`

UNIX: `$(SI EBEL_ROOT)/l og/PROCESS/output`

The name of the log file increments for subsequent log files that are created when the Siebel Upgrade Wizard encounters a problem and the user runs 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.

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

- Windows: `sw_cfg_xxx. l og` and `si ebel . l og`
- UNIX: `srvrupgwi z_*. l og` and `si ebel _*. l og`. For example, `srvrupgwi z_001. l og`
- Any other log file that existed before the start of the upgrade

CAUTION: UNIX Only: The log file `srvrupgwiz_001. l og` is a different file than `srvrupgwiz1. l og`. Do not ignore log files named `srvrupgwiz1. l og`, `srvrupgwiz1_01. l og` and so on.

Related Topics

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

Troubleshooting Steps 28: Errors.rtf -- Siebel version 7.5.3, located on SupportWeb

To manually review the log files for unacceptable errors

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

Windows: *SIEBEL_ROOT\log\PROCESS\state*

UNIX: *\$SIEBEL_ROOT/log/PROCESS/state*

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

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

UNIX: *DBSRVR_ROOT/DATABASE_PLATFORM/errors.txt*

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

Windows: *SIEBEL_ROOT\log\PROCESS\output*

UNIX: *\$SIEBEL_ROOT/log/PROCESS/output*

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

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

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

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

The log files generated by the repository Siebel Upgrade Wizard (for example *srvrupgwi z1.log*) appear in the errors file as *upgwiz1.log*, *upgwiz2.log*, incrementing for additional log files.

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

On SupportWeb, navigate to Troubleshooting Steps > Product Areas > Upgrade, and choose Troubleshooting Steps 21. Common errors and how to resolve them are listed in this document. If the error is not listed or you cannot resolve it, contact Siebel Technical Support. *Do not proceed with the upgrade.*

- 6 Repeat Step 5 for each log file.

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 the Upgrade.”](#)

After a successful installation and upgrade, you must manually save and archive the log files located in the *SIEBEL_ROOT/1og/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 *si ebel _l og_ archi ve* environment variable to 20—for example, to retain twenty (20) log files.

Reviewing Log Files for Errors in Jobs Run from the Midtier

Upgrades: All upgrades.

Environments: Development environment only.

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

To find errors in the log files for jobs run from the midtier during an upgrade, search on the word `SQLSTATE`.

Running SQL in Siebel Logs

Upgrades: All upgrades.

Environments: Development (mainframe-centric) environment and Production environment.

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

By using SPUFI or the command line, you can construct SQL queries to run against the source or the target log tables.

The following statements reports the status of the load jobs for the source or target:

- `SELECT JOB_DESC, JOB_NAME, JOB_STATUS FROM CQ10A901.TMP_SBLLOG_SRC 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 statement to report the status of any jobs by changing the `LIKE JOB_NAME` 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_SRC WHERE  
JOB_STATUS != 'COMPLETED SUCCESSFUL' AND JOB_NAME LIKE 'LKB%';
```

15 Performing the Siebel Tools Repository Merge

This chapter contains the following topics:

- “About Inheriting Upgrade Behavior” on page 245
- “About the Postmerge Utilities” on page 248
- “About the Incorporate Custom Layout (ICL) Upgrade Option” on page 248
- “Configuring Objects to Inherit Upgrade Behavior” on page 253
- “Renaming the Tools Repository” on page 254
- “Migrating Repository Objects to the Standard UI” on page 255
- “Running the Repository Preparation Wizard on Release 6.x Repository” on page 259
- “Performing a Repository Merge” on page 262
- “Determining if a Repository Merge was Successful” on page 271
- “Reviewing Attribute Conflicts in the Repository Merge” on page 273
- “Reviewing Deleted Objects in the Repository Merge” on page 275
- “Reviewing Obsolete Objects in the Repository Merge” on page 275
- “Reviewing Customized Business Components” on page 276
- “Generating EIM Temporary Columns After a Repository Merge” on page 277
- “Regenerating the Repository Definition Files” on page 278
- “Deleting Unneeded Repository Files” on page 280
- “Safeguarding the New Custom Repository Export File” on page 280
- “Moving the Repository Files” on page 281
- “Creating a New SRF File” on page 282

About Inheriting Upgrade Behavior

Upgrades: All upgrades.

Environments: Development environment only.

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

Use standard objects as upgrade ancestors. A *standard object* is an uncustomized repository object provided by Siebel Systems. Objects in the Prior Siebel Repository and the New Siebel Repository are standard objects.

Typically, you create customized objects by making a copy of a standard object and then modifying the copy, called the *descendent*.

You can specify upgrade ancestors for the following object types:

- Applets
- Business components
- Integration objects
- Reports

You specify the upgrade ancestor in the descendent's Upgrade Ancestor field in Siebel Tools. During the repository merge, the descendent is upgraded in the same way as the upgrade ancestor.

For example, you copy the standard object Applet A and name the descendent Applet B. In the New Siebel Repository, Applet A has been modified to include several new list columns.

Here is what happens to Applet B (the descendent) during the repository merge:

- If you specified Applet A as the upgrade ancestor, then Applet B is upgraded to include the two new list columns.
- If you did not specify Applet A as the upgrade ancestor, then Applet B does not receive the two new list columns.

Limitations on the Upgrade Ancestor Property

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

- If an upgrade ancestor is not present in the New Siebel Repository, it is obsolete, and its upgrade behavior is not propagated to descendents. If an ancestor object is obsolete in the New Siebel Repository, descendents are not also obsolete. They are copied to the New Customer Repository.
- If an upgrade ancestor is not present in the New Siebel Repository, error messages display during the repository merge and are written to the merge log file. These errors are acceptable and do not mean the merge has failed.
- The Upgrade Ancestor property is not considered during repository imports. However, imported objects can specify an upgrade ancestor. When the next application upgrade is done, the Upgrade Ancestor property is taken into account.
- The setting of the Upgrade Ancestor property is not considered when applying application patches. If the upgrade ancestor is modified by the patch, descendents are not modified.
- The Upgrade Ancestor Property is not used by the postmerge utilities. The postmerge utilities revise layout objects without regard to how ancestor objects were revised.
- Specifying an upgrade ancestor for objects slows the repository merge.

Upgrade Ancestor Picklist

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

Applets

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

Reports

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

Business Components

- Bus Comp is the same as the current business component
- Class is the same as the current business component
- Upgrade Ancestor is null
- Business component is a standard object

Integration Objects

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

Propagating Changes to Objects After the Merge

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

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

Related Topics

[“About the Repository Merge” on page 38](#)

[“About the Postmerge Utilities” on page 248](#)

About the Postmerge Utilities

Upgrades: All upgrades.

Environments: Development environment only.

The postmerge utilities start automatically after the merge is complete. The postmerge utilities make revisions to layout objects from the Prior Customer Repository that have been added to the New Customer Repository:

- **User interface navigation.** Release 7.7 introduced a revised method for UI navigation. The postmerge utilities make revisions to new or customized layout objects from the Prior Customer Repository to integrate them into the new UI navigation scheme.
- **Multi-Value group applets.** The postmerge utilities enable the shuttle feature for multi-value group (MVG) applets. MVG applets must meet the criteria described in *Configuring Siebel eBusiness Applications* for MVG shuttle applets.
- **Flow-Based applets.** The postmerge utilities convert many flow-based form applets to grid-based applets. If you have customized a form applet by adding fields and controls, the postmerge utilities place these at the bottom of the applet. You must reposition them after the merge.

How the Postmerge Utilities Work with Objects with Upgrade Ancestors

The postmerge utilities do not use the Upgrade Behavior property. This includes the following changes:

- Enabling the shuttle feature for MVG applets
- Converting applets to grid-based layout.

The postmerge utilities make these changes to applets based on their characteristics rather than upgrade behavior. For a full description of the characteristics applets must have before these changes can be applied, see *Configuring Siebel eBusiness Applications*.

Related Topics

[“About the Repository Merge” on page 38](#)

[“About Inheriting Upgrade Behavior” on page 245](#)

About the Incorporate Custom Layout (ICL) Upgrade Option

Upgrades: Release 7.x using ICL.

Environments: Development environment only.

If you are upgrading from Siebel 7.x, you can choose to preserve the layout of applets and views. This reduces the workload required to reconfigure customized UI objects after upgrading to a new release that has significant UI changes.

For example, if you are upgrading from Release 7.0.x, you can preserve Release 7.0.x layouts when you upgrade. If you are upgrading from Release 7.5.x, you can preserve Release 7.5.x layouts when you upgrade.

An ICL upgrade preserves the layouts of both standard and customized UI objects.

When to Use ICL

To determine when to use ICL, assess both your current implementation and your future implementation plans.

Use the following factors to assess your current implementation:

- The higher the number of customizations of standard UI objects, the greater the benefit from choosing ICL.
- If you have customized only a few standard UI objects, choosing ICL may not be cost effective.
- If you have customized many standard UI objects but these objects are in applications that do not support ICL, performing an ICL upgrade may not be cost effective.

It is important to assess your future development plans because you can select ICL for only every other upgrade. This means you should choose ICL for upgrades where your customizations are heaviest.

For example, your current release has a moderate number of customized standard UI objects. At the next release, you plan to make heavy customizations of standard UI objects. Consider not choosing ICL when upgrading your current release. This enables you to choose ICL at the next release, when your customizations will be heavier.

You Can Preserve Layouts for Only One Upgrade

You can preserve a release's layouts for only one upgrade.

If you select ICL for an upgrade, then at the following upgrade, the repository merge process requires that you do the following:

- Before the repository merge, you must upgrade the UI objects preserved in the prior ICL upgrade to the standard objects for the release from which you are upgrading. This brings your UI forward to the installed release.
- Perform the repository merge for the new release. Selecting ICL for this merge is not recommended. If you select ICL, you must review all UI objects to verify that they correctly preserve the layout of the release from which you are upgrading. You must also reconfigure customized applets. If you do not select ICL, you only need to reconfigure customized applets.
- After the upgrade to the new release, you must reconfigure customized layouts for applets, views, and screens.

How Layouts Are Preserved

When you select ICL during a repository merge:

- The layout of applets in the Prior Customer Repository is preserved.

- The location of controls in applets is preserved.
- The association of applets to views is preserved.
- The layout of views is preserved.
- The association views to screens is preserved.
- View locations and groupings within screens are similar but not identical.
- Changes to navigation in the release you are upgrading to are implemented.

For example, in Release 7.7, a new navigation scheme was introduced. If you are upgrading from a release prior to 7.7, this navigation scheme is implemented during upgrade regardless of whether you choose ICL. If controls in applets conflict with the new navigation scheme, they are handled as described later in this topic.

How the Repository Manages an ICL Upgrade

Repository objects are affected as follows during an ICL merge:

- View and applet Web templates, including some child objects, in the New Customer Repository are inactivated and made read-only. Their names are appended with the string -UPG. You cannot delete these objects from the repository or change their modification time.
- The inactivated Web templates and child objects are replaced with corresponding records from the Prior Customer Repository. This preserves the layout of the release from which you are upgrading.

What Happens at the Next Release

At the next release, before performing the repository merge, you must reverse the ICL changes by running a Siebel Tools utility that does the following:

- Removes the active Web templates and child objects in the repository. (These are the objects that were preserved from the Prior Customer Repository.)
- Activates the Web templates and child objects with the -UPG suffix and removes the suffix. This reverses the effect of ICL and upgrades your UI to the standard layouts.

If you do not select ICL, the applet and view Web templates in the New Customer repository replace uncustomized applet and view Web templates in the Prior Customer Repository. This upgrades the UI directly to the new release.

If you select ICL again, the newly activated objects from the Prior Customer Repository would replace their counterparts in the New Customer Repository during the merge. Since you have not previously reviewed the newly activated objects, you would have to review the entire UI after the merge, which would require additional reconfiguration. This is why selecting ICL again is not beneficial.

The Tools utility provides a method for defining a filter to identify Web templates and child objects that have been customized since the prior upgrade. If an object has been customized, it is not replaced by the standard object. You must manually reconfigure these objects after the merge.

The ICL Option Does Not Affect All Objects

In Release 7.7, Siebel Tools objects have a new, read-only property called Upgrade Behavior:

- If this property is set to NULL or Preserved and ICL is selected during repository merge, ICL preserves the layout of the previous release.
- If this property is set to ADMIN, then ICL does not attempt to preserve the layout of the previous release. Layouts in the current release are applied to the object.

The following types of objects have the Upgrade Behavior property set to ADMIN. The layout of these objects is not preserved even if ICL is selected during the repository merge:

- Siebel administrative views and applets where core functionality must be upgraded to include feature enhancements. In some cases, these enhancements have such a large impact on the user interface that preserving layouts from a prior release is not practical.
- Siebel Employee Relationship Management (ERM) and Siebel Marketing applets and views, since they have been significantly enhanced for this release.

How ICL Affects Specific UI Objects

When you select the ICL option, you use a new set of Web templates and certain child objects that preserve the layout of the release you are upgrading from. This affects UI objects in the ways listed below.

General UI

The following characteristics apply to uncustomized and customized UI objects when you choose an ICL upgrade:

- Layouts are preserved.
- Attributes defined on Web templates are preserved. This includes relative positioning of fields and layouts.
- Properties of controls or list columns are not preserved. For example, height, width, caption, and pop-up icons are not preserved.
- ICL preserves existing controls. No controls are removed. Location of controls is also preserved.
- System pages such as Help, About View, About Record, and Technical Support are not preserved.
- Font style, color, text size, and object alignment are not preserved.
- Stylesheets are not preserved. Web template files are approximations of their 7.x counterparts.

Applets

The following characteristics apply to uncustomized and customized applets when you choose an ICL upgrade:

- Associations to views are preserved.
- The upgrade converts certain form-based applets to grid layout. If ICL is selected, the postmerge utilities converts them back to flow-based.

- Applet drilldown child objects are preserved.
- Applet-level and application-level menus are not preserved.
- For flow-based applets, field labels are placed on top as in 7.0x and 7.5x. However, you can choose to put the labels on the left as part of setting up the merge.
- Release 7.7 added two buttons in attachment applets—New File and New URL. If you select ICL, these buttons will not be added to applets.
- As of Release 7.7, MVG dialog boxes display only an OK button to close them. If you select ICL, MVG dialog applets will have both OK and Cancel buttons.
- Release 7.7 added a UI feature, Query Assistant. If you select ICL, it will not be added to applets.
- Release 7.7 applets have three default buttons—New, Delete, and Query. If you select ICL, applets will keep their existing default buttons.
- In Release 7.7, the Reset button was removed from applets. If you select ICL, applets will keep the Reset button.
- In Release 7.7, Pick and Association applets support in-line queries. This new feature is implemented regardless of whether or not you choose ICL.

Views

The following characteristics apply to uncustomized and customized applets when you choose an ICL upgrade:

- Applets appear in the views they reference.
- Their location in the view is preserved.
- View Web template child objects are preserved.
- On parent list views, detail tabs display in a row near the middle of the view as they do in 7.0x and 7.5x.

Screens

The following characteristics apply to uncustomized and customized screens when you choose an ICL upgrade:

- Screens contain the views they reference.
- The view locations and groupings will be similar but not identical.

Navigation

- A new navigation method, introduced in Siebel 7.7, is applied to the UI regardless of whether or not you select ICL during the repository merge.

How ICL Works with the Upgrade Ancestor Property

The postmerge utilities implement the ICL changes to objects in the New Customer Repository. These utilities do not consider the setting of the Upgrade Ancestor property when making changes to Web templates or child objects. If the postmerge utilities change to Object A, they do not propagate the changes to objects that specify Object A as the upgrade ancestor.

Repository Merges that Do Not Use ICL

For upgrades from Release 7.x that do not use ICL, new Siebel objects, usability enhancements, and layout changes are applied. This modifies the layout of applets, views, and screens and attempts to merge rather than preserve customizations.

After the upgrade, you will need to identify those areas of the upgraded interface where you want to reapply customizations.

Related Topics

[“About Inheriting Upgrade Behavior” on page 245](#)

[“About the Postmerge Utilities” on page 248](#)

Configuring Objects to Inherit Upgrade Behavior

Upgrades: All upgrades.

Environments: Development environment only.

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

As of Release 7.7, you can link objects together so that one object inherits the upgrade behavior of another. You do this by specifying an upgrade ancestor for an object.

For more information, see [“About the Siebel Database Server” on page 41.](#)

You can specify an upgrade ancestor for the following object types:

- Applet
- Business component
- Integration object
- Report

Before doing the repository merge, review new objects you have created and determine whether you want to specify an upgrade ancestor.

To specify an upgrade ancestor

- 1 Navigate to the object in Siebel Tools.

- 2 Click in the Upgrade Ancestor field.
A dialog box appears. It lists available upgrade ancestors.
- 3 Select the desired upgrade ancestor and click Pick.

To view the descendents or copies of an object

- 1 Right-click on the object.
A dialog box appears and lists the descendents.
- 2 Select View Descendents from the picklist.

Renaming the Tools Repository

Upgrades: All upgrades.

Environments: Development environment only.

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

Your upgrade will encounter errors if you have more than one existing repository for a development upgrade. Export and archive any redundant repositories from your environment before you upgrade your development environment.

To prevent a naming conflict, before you run the upgrade, you must rename your existing development repository (*Siebel Repository*) to *Prior Customer Repository*. After the upgrade, your new development repository is given the name *Siebel Repository*.

When you rename the repository, you may prefix the name with additional characters (except for a leading space). Do not append the repository name with additional characters, because this results in an error.

To rename the repository

- 1 Start Siebel Tools and connect to the Siebel Database.
Use the version of Siebel Tools for the Siebel release from which you are upgrading.
- 2 If you archived repository objects as .sif files, and you want to have them available in your application, import these archive files back into the repository.
If you do not check these objects back into the repository, they will not be upgraded. You need only to check in those archived objects that you need in the future and want to have available in your upgraded application.
- 3 Choose View > Options.
- 4 Click the Object Explorer tab.
The Object Explorer hierarchy displays.

- 5 Locate Repository in the list, put a check mark in the adjacent box, and then click OK.
This exposes the repositories.
- 6 In the Object Explorer, click the Types tab, and then Click Repository.
- 7 In the Repositories list view, verify that your existing repositories do not use the names reserved for the upgrade process:
 - New Customer Repository
 - New Siebel Repository
 - Prior v6.x Siebel Repository
- 8 Locate your current Siebel Repository in the list applet.
- 9 Click on the name and change it to Prior Customer Repository.
When you rename the repository, you may prefix the name with additional characters (except for a leading space, which causes the validation to fail).
Do not append the repository name with additional characters, because this results in an error.
For more information about renaming repositories, see *Configuring Siebel eBusiness Applications*.
TIP: Prefix the repository name with a date; for example, 20031202 Prior v6.x Customer Repository.
- 10 Step off the list to commit the record to the database.
If the validation check fails, verify that the repository name is not prefixed with a leading space.
If no repository is named Prior Customer Repository, the upgrade process renames the Siebel Repository to Prior Customer Repository.

Migrating Repository Objects to the Standard UI

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

Environments: Development environment only.

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

If you selected the Incorporate Custom Layout (ICL) feature for your previous upgrade, you must migrate UI repository objects to the standard UI before doing the repository merge for your next upgrade. The migration does not affect customized objects in the Prior Customer Repository. Customized objects are preserved and are included in the repository merge.

Migrating to the standard UI uses defined logic to select objects and then modifies these objects as follows:

- **ICL UI objects:** Inactivates and deletes the Web templates and certain child objects. These are the objects that preserved the layouts of the previous release.

- **Standard UI objects.** Activates the UI Web templates and child objects that provide the standard UI look and feel for the release you are upgrading from. These objects have -UPG appended to the name.

For a full description of how ICL works during an upgrade, see [“About the Incorporate Custom Layout \(ICL\) Upgrade Option” on page 248.](#)

How Customizations Are Handled

ICL UI objects you have customized after the previous upgrade are not affected by the migration to the standard UI. These changes are preserved and are included in the repository merge for the new release.

Siebel Tools identifies these customizations by comparing the modification time of ICL UI objects with their corresponding standard UI objects. If the modification times differ by more than a specified lag time, Siebel Tools does not change the ICL UI object, and it is treated as a customized object in the upcoming repository merge.

For example, your previous ICL repository merge required about three days to complete. This means the modification time of an ICL UI Web template and its corresponding standard UI Web template did not differ by more than three days when the repository merge completed. You later modified the ICL UI Web template. Its modification time now differs by more than three days from the corresponding standard UI Web Template.

You then use Siebel Tools to migrate the repository to the standard UI and use a lag time of three days. Since the modification time comparison for the ICL UI Web template is greater than the lag time, the ICL UI Web template is not replaced by the corresponding standard UI Web template. The ICL UI Web template is treated as a customized object in the upcoming repository merge.

The default lag time is 72 hours. You can specify a lag time between 24 and 120 hours. Observe the following guidelines:

- Avoid setting a lag time that is shorter than the time required to complete the previous repository merge. This can cause objects that were not customized after the merge to be treated as customized.
- Do not set a lag time that is significantly larger than the length of the previous repository merge. This increases the risk that customizations were made before the lag time expired. These customizations will be lost during the upcoming repository merge.

How Repository Objects Are Changed

The migration process locates both standard UI objects and ICL UI objects in the repository. The process then determines whether to modify the standard UI object or the corresponding ICL UI object:

- Deleting the ICL UI object and activating the corresponding standard UI object migrates the UI to standard.
- Deleting a standard UI object and retaining the corresponding ICL UI object preserves the customized ICL UI object.

Table 21 on page 257 shows the logic used to modify these objects. Interpret the table columns as follows:

- **ICL UI Object Found?** Yes means Siebel Tools has located a standard UI object, and it has a corresponding ICL UI object. No means that a corresponding ICL UI object is not in the repository. In most cases, this is because the object was inactive in the Prior Customer Repository during the previous merge. When this occurs, the object is appended with -UPG. No ICL UI object that preserves look and feel is created.
- **Within Lag Time?** Yes means that the modification time comparison is within the specified lag time. No means the comparison is not within the specified lag time. No indicates that you have modified the ICL UI object after the repository merge.
- **Status in Prior Standard Repository.** Siebel Tools checks the Active/Inactive status in the Prior Standard Repository to prevent activating an object that is inactive in the Prior Standard Repository.

In the table, Prior Customer Repository refers to your current Siebel Repository, which you renamed in preparation for the upcoming repository merge.

Table 21. Logic Used to Modify Repository Objects

ICL UI Object Found?	Within Lag Time?	Status in Prior Standard Repository	How Migration Modifies Objects in the Prior Customer Repository
Yes	Yes	Active	<ul style="list-style-type: none"> ■ Deletes ICL UI object. ■ Removes -UPG suffix from name of corresponding standard UI object. ■ Changes object status from Inactive to Active in the Prior Customer Repository. ■ Removes read-only restriction. Object can be modified or deleted.
Yes	Yes	Inactive	<ul style="list-style-type: none"> ■ Deletes ICL UI object. ■ Removes -UPG suffix from name from corresponding standard UI object. ■ Status is not changed from Inactive to Active in the Prior Customer Repository. ■ Removes read-only restriction. Object can be modified or deleted.

Table 21. Logic Used to Modify Repository Objects

ICL UI Object Found?	Within Lag Time?	Status in Prior Standard Repository	How Migration Modifies Objects in the Prior Customer Repository
Yes	No	N/A	<ul style="list-style-type: none"> ■ Makes no changes to ICL UI object. (Object has been customized after the repository merge.) ■ Deletes corresponding standard UI object and children of this object.
No	N/A	N/A	<ul style="list-style-type: none"> ■ Status is not changed from Inactive to Active. ■ Removes read-only restriction. Object can be modified or deleted.

How Logging Is Done

The migration process logs the changes to the repository in the following log file:

SIEBEL_ROOT\log\icl migration.log

The beginning of the log lists top-level objects that were affected by the ICL feature. For each object, the log then iteratively lists the operations performed on all child objects.

Specifying a Lag Time

The default lag time is 72 hours. You can revise this by editing the Siebel Tools .cfg file. The minimum is 24 hours, and the maximum is 120 hours.

Prerequisite: Your Siebel Tools installation must be release 7.7 or later.

To specify a lag time

- 1 In the Siebel Tools installation directory, navigate to the \bin\lang directory, where lang is the installed language, for example ENU.
- 2 Using a text editor, open tool.cfg, and locate the [Siebel] section.
- 3 Add the following variable at the end of the section:

```
PriorICLMergeTimeLag = time
```

where time is an integer greater than or equal to 24 and less than or equal to 120.
- 4 Save the file.
- 5 Restart Siebel Tools.

Migrating to the Standard UI

A menu option in Siebel Tools migrates the repository Web templates and child objects to the standard UI. Upgrading the UI requires a minimum of three hours to complete.

Prerequisites:

- Your Siebel Tools installation must be release 7.7
- Set the lag time as desired. The default is 72 hours
- Verify that you have backed up the upgraded database

To migrate to the standard UI

- 1 Start Siebel Tools.
- 2 In the Tools menu, choose Upgrade. Then select Migrate ICL Objects to Standard.
- 3 In the ICL Migration dialog, make the following selections and click Continue:
 - **Prior Customer Repository:** Select Prior Customer Repository. This is the Siebel Repository in your current release, which you have renamed in preparation for the upgrade.
 - **Prior Standard Repository:** Select Prior V7.X Siebel Repository. This is the Prior Standard Repository for your currently installed release. This repository was loaded when you performed the database upprep for the new release.
- 4 In the ICL Migration Warning dialog, click Yes to confirm you have backed up the Prior Customer Repository.

The ICL Migration Status dialog box appears. It displays log entries during the migration.

During the migration, you cannot perform other operations in Siebel Tools. You also cannot close Siebel Tools.

A pop-up message displays when the migration is complete.

- 5 To cancel the migration after it has begun, do the following:
 - a In the ICL Migration Status dialog box, click Cancel.
 - b In the dialog box that asks you to confirm you want to cancel, click Yes.

If you cancel the migration after it has begun, you must restore the saved Prior Customer Repository from backup and begin the migration again.

Running the Repository Preparation Wizard on Release 6.x Repository

6.x upgrades only. This topic applies to upgrades from Release 6.x.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform the 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 7.x 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 7.x, 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 7.x 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 7.x application, do not use the default input file to merge labels and fields. See the Caution note in Step 3 on page 260.

- **Merge Web templates.** Release 7.x 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 SupportWeb 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 `appls.txt`. Modify the input file only if you have additional applets that should not be merged.

CAUTION: If you are upgrading to a Siebel Financial Services 7.x application, you should not use the default input file, `appls.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, you should browse to `SIEBEL_ROOT\Tools\Bin` directory and select the input file named `fi ns_appls.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 260](#), then click Next.

NOTE: The utility operates on every form applet except those specified in the input file. The default input file is `appls.txt`. Modify the input file only if you have additional applets that should not be 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 `templ atemerge.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 following table.

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>templ atemerge.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 `templ atemerge. 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 Repository Merge

Upgrades: All upgrades.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform the 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 [“Reviewing Attribute Conflicts in the Repository Merge” on page 273](#) for examples of conflict resolution.

About the Repository Merge

The configuration utility that you ran while upgrading your development environment loaded two version 7.7 standard repositories. You must now use Siebel Tools to merge your existing custom configuration into one of these new repositories, creating a version 7.7 custom configuration that includes all of your previous configuration changes.

The four repositories that currently exist in your development database are listed in [Table 22](#).

Table 22. Development Database Repositories

Repository Name	Description
Prior V6.x Siebel Repository or Prior V7.x Siebel Repository	Standard version 6.x or 7.x repository, depending on the version from which you are upgrading.
Prior Customer Repository	Customized version 6.x or 7.x repository, depending on the version from which you are upgrading.
New Siebel Repository	Newly loaded version 7.7 standard repository.
New Customer Repository	Newly loaded version 7.7 repository into which your custom configuration is merged.

Follow the guidelines provided in [“Optimizing Performance of the Repository Merge” on page 264](#) 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.

- Use a workstation with a minimum of 512 megabytes (MB) of RAM and at least 2 GB of virtual memory or a 2-GB page file. Otherwise, the repository merge does not complete successfully.
- Close all other applications before performing a repository merge.
NOTE: Some repositories may require additional memory, based on level of customizations.
- If necessary, increase the swap space, using the Control Panel System applet, and then restart the development workstation before proceeding.
- Make sure you have a high-performing network connection.
NOTE: A slow network connection significantly increases the time required for the repository merge.

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 265](#).
- For EBCDIC databases, perform the procedure, [“Merging Repositories for an EBCDIC Database” on page 269](#).

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.
 - 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.
 - Close all other applications.
 - Close all 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.
 - Make sure that temporary tablespace 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.
- 3 Run the repository merge on a different machine than the Siebel Database Server. When you run the repository merge on a different machine, you can increase the foreground performance of the repository merge. To do this, increase the application performance setting on the machine and verify that the Siebel Tools application is the foreground application.

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 ensure 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 269](#).

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.

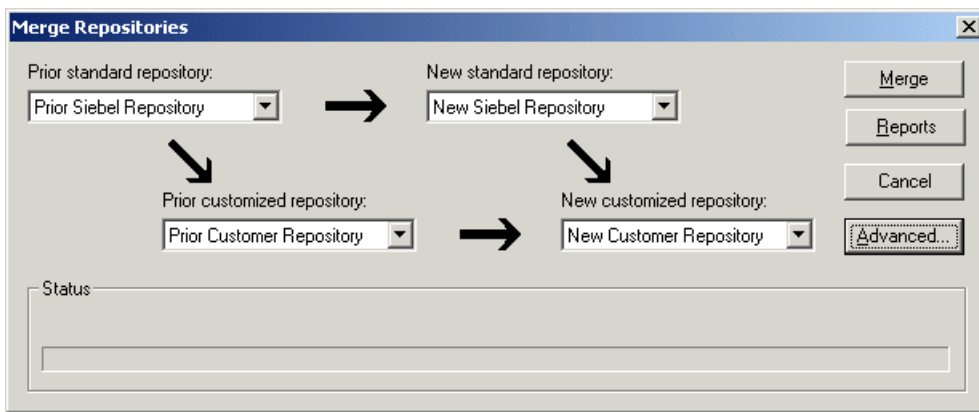
This is the user interface language for SRF files compiled from the New Customer Repository. It will also be the language used by the postmerge utilities.

- 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 opened 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.x only).** This button manually launches the postmerge reports for user interface configuration described in *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.x 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 5 on page 267](#).

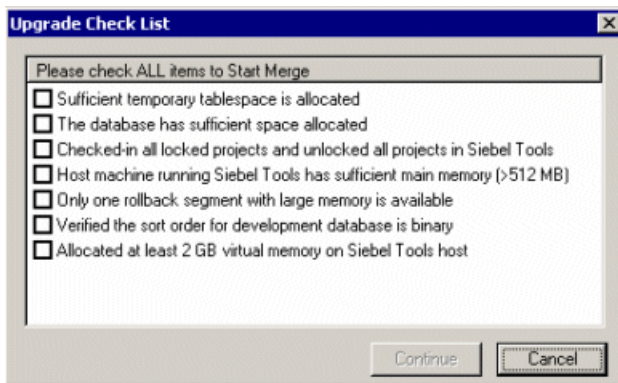
- In the Merge Repositories dialog box, choose the appropriate repository name from each picklist, using the repository names listed in the following table.

Drop-Down List Item	Value to Choose
Prior Standard Repository	Prior 6.x 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

- Review the settings in the Merge Repositories dialog box, and then click Advanced. The Merge Options dialog box appears.
- 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.

- To continue, click OK. The Upgrade Check List dialog box appears.
- 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.

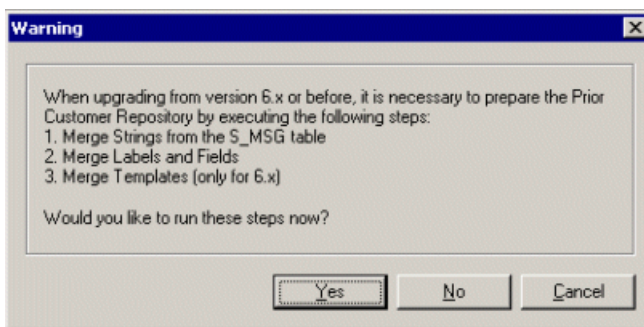


10 To continue, click Continue.

A warning screen prompts you to prepare the Prior Customer Repository for the merge if you are upgrading from Release 6.x.

11 In the Warning screen, choose the option for your upgrade path:

- For upgrades from Release 6.x, 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 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. Additional time should be planned 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.

12 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. See *Upgrade Guide* for more information about the postmerge utilities.

13 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 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 After preparing to run the Database Server configuration utility, use the repimexp utility to export the following repositories from your prior EBCDIC database:

- Prior Standard Repository
- Prior Customized Repository
- New Standard Repository
- New Customer Repository

To export a repository, type the following command in the command line:

```
repimexp /A E ARGUMENT_LIST
```

The following parameter settings are passed as export arguments to repimexp.

Parameter	Required	Meaning
<i>/U USER_NAME</i>	Yes	Siebel administrator user name.
<i>/P PASSWORD</i>	Yes	Siebel password.
<i>/C ODBC_DATASOURCE</i>	Yes	ODBC data source. The default is the value in the <i>SIEBEL_DATA_SOURCE</i> environment variable.
<i>/D TABLE_OWNER</i>	Yes	Siebel Database tableowner. The default is the value in the <i>SIEBEL_TABLE_OWNER</i> environment variable.
<i>/W LANGUAGE_CODE</i>	Yes	Language mode, such as ENU or JPN. The default is <i>SIEBEL_LANGUAGE</i> . If this is not set, the default is ENU.
<i>/R REPOSITORY</i>	Yes	Repository name. The default is Siebel Repository.
<i>/1 EXPORT_REPOSITORY_USERNAME</i>	Yes	Export repository user name. The default is the same as for <i>/U</i> .
<i>/2 EXPORT_REPOSITORY_PASSWORD</i>	Yes	Export repository password. The default is the same as for <i>/P</i> .
<i>/3 EXPORT_REPOSITORY_ODBC_DATASOURCE</i>	Yes	Export repository ODBC data source. The default is the same as for <i>/C</i> .

Parameter	Required	Meaning
<i>/4 EXPORT_REPOSITORY_TABLE</i>	Yes	Export repository tableowner. The default is the same as for /D.
<i>/5 EXPORT_REPOSITORY</i>	Yes	Export repository name. The default is Siebel Repository.
<i>/B APPLICATION_SERVER_ROOT</i>	Yes	Siebel Server installation directory to override <i>SIEBEL_HOME</i> environment variable.
<i>/F DATA_FILE</i>	Yes	Data file, including path, to which to export.
<i>/T <Y N></i>	No	Test only, do not export into database. <ul style="list-style-type: none"> ■ Y=Yes ■ N=No
<i>/V <Y N></i>	No	Verify data. The default is N.
<i>/N <0 1 2></i>	No	Change creation and update information: <ul style="list-style-type: none"> ■ 0= no change ■ 1= change <i>CREATED_BY</i>, <i>UPDATED_BY</i> ■ 2=change <i>CREATED_BY</i>, <i>UPDATED_BY</i>, dates columns The default is 1.
<i>/M <Y N></i>	No	Commit changes even if verification fails. The default is N.
<i>/L LOG_FILE</i>	No	Log file for output messages.

2 Prepare a new Siebel 7.7 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_DBSCRIPT
- S_BI TMAP_DATA
- S_SERVICE_SCRIPT

3 Use the repimexp utility to import the repositories exported from the EBCDIC database in [Step 1 on page 269](#) into the ASCII database prepared in [Step 2](#).

- To import a repository using repimexp, type the following command in the command line:

```
repimexp /A I /G LANGUAGE_CODES
```

where *LANGUAGE_CODES* is a list of all languages used, for example, ENU, FRA, JPN.

NOTE: If you want to import your repository with locale objects, you must specify at least one language code. Otherwise no locale objects are imported. Therefore, when you compile the imported repository, it will not have any text in the user interface.

- 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 265.](#)
- 6 Perform the task [“Generating EIM Temporary Columns After a Repository Merge” on page 277.](#)
- 7 Perform the task [“Determining if a Repository Merge was Successful” on page 271](#) to review the repository merge results. Verify that the repository merge was successful, and that all reported validation messages are either acceptable or fixed.
- 8 Use the repimexp utility 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 repimexp utility 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, PM7.7; UpgEimCol ,
```

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, PM7.7; UpgEimCol '
```

```
Where name='New Customer Repository'
```

Determining if a Repository Merge was Successful

Upgrades: All upgrades.

Environments: Development environment only.

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

The repository merge is successful if it completes without unacceptable errors:

- **Acceptable errors.** If an ancestor object is specified in an object definition, and the ancestor object is not present in the New Siebel Repository, this causes a merge error. This is an acceptable error and can be ignored.

Here is an example of an acceptable error in the postmerge utilities log file, merge0.txt:

```
!!ERROR: : CANNOT upgrade objects which have Bridging Tracking Profile Appl et - Product marked as 'Upgrade Anc'
```

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

Merge errors are displayed in the Upgrade Applications Objects List view in Siebel Tools. Additional details on merge errors are located in the repository merge log:

Tools_install_dir\bin\merge0.txt.

Each time you run the merge process, the name of the merge0.txt file is incremented, for example merge1.txt.

To determine if the repository merge was successful

- 1 In Siebel Tools, navigate to Screens > Application Upgrader > Application Upgrade Object List.
- 2 In the Application Upgrades list, select the record of the merge.
- 3 Check the entry in the Status column.

- **Completed.** This means the merge completed without errors.
- **Completed with Errors.** This means the merge contains errors.

If the Status column indicates Completed, no further action is required. The merge was successful.

If the Status column indicates *Completed with Errors*, you must review the errors to determine if the merge was successful. To review the errors, complete the remaining steps in this task.

- 4 In the Object Differences list, click Query.
- 5 In the Status field, enter ERROR: : * .
- 6 Press Enter to run the query.

This displays all the objects where the merge process encountered errors.

- 7 Open the merge log file, merge0.txt. It is located in the following directory:
Tools_install_dir\bin.

If there are multiple files, open the one with the highest number in the file name, for example merge1.txt.

- 8 To locate merge errors in the file, search for ! (exclamation point).
- 9 Use the objects displayed in the Object Differences list and the errors displayed in the log file to analyze the errors:
 - If all the errors are acceptable, the merge is successful.

- If the log contains unacceptable errors, the merge has failed.

10 If the merge contains unacceptable errors, go to SupportWeb and search for *Troubleshooting Steps 19**.

This document explains the meaning of many of the error messages that can appear in the log file. Use this document to correct the errors. If you cannot resolve all the errors, use the directions in *Troubleshooting Steps 19* to contact Siebel Technical Support.

Reviewing Attribute Conflicts in the Repository Merge

Upgrades: All upgrades.

Environments: Development environment only.

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

You can change how attribute conflicts were resolved during the repository merge.

How Attribute Conflicts Occur

Attributes, such as color and height, are defined on repository objects. An attribute conflict occurs in the repository merge when all of the following are true:

- The object and attribute exist in the Prior Siebel Repository, the Prior Customer Repository, and the New Siebel Repository.
- You have changed an attribute value in the Prior Customer Repository. (The attribute value in the Prior Customer Repository and the Prior Siebel Repository are not the same.)
- The attribute has been changed in the New Siebel Repository. (The attribute in the New Siebel Repository and the Prior Siebel Repository are not the same.)

If the values in the Prior Siebel Repository and New Siebel Repository are the same but the value in the Prior Customer Repository is different, there is no conflict. The value in the Prior Customer Repository is preserved in the merged repository.

The merge process resolves attribute conflicts by referring to the setting of the Standard Win property. For about 90% of repository objects, the merge process resolves attribute conflicts by using the attribute values in the New Siebel Repository. Do not change the setting of this property.

Application Upgrade Attribute List View

You can review and change how attribute conflicts were resolved using the Application Upgrade Attribute List view in Siebel Tools. The Attribute Differences List in the view includes the following columns:

- **Object Name.** The name of the object on which the attribute is defined.
- **Attribute.** The attribute name.

- **Conflict.** The merge process puts a check mark in this field if there was an attribute conflict during the merge.
- **Resolution.** Displays which attribute value the merge process used to resolve the conflict:
 - **Standard Value.** The attribute value in the New Siebel Repository was used. This value is displayed in the New Standard column.
 - **Custom Value.** The attribute value in the Prior Customer Repository was used. This value is displayed in the Prior Customized column.
- **Override.** Put a check mark in this column to change how the attribute conflict is resolved. Overriding the resolution changes the attribute value in the merged repository. If the resolution was the Standard Value it is switched to the Custom Value and vice versa.

Putting a check mark in the Override column does not change the value displayed in the Resolution column. It indicates that the displayed value was manually overridden in the merged repository.
- **Prior Standard.** Displays the value of the attribute in the Prior Siebel Repository.
- **Prior Customized.** Displays the value of the attribute in the Prior Customer Repository. In the Resolution column, this value is called the Custom Value.
- **New Standard.** Displays the value of the attribute in the New Siebel Repository. In the Resolution column, this value is called the Standard Value.

Prerequisite: The repository merge must have been successful. See [“Determining if a Repository Merge was Successful” on page 271](#).

To review attribute conflicts

- 1 In Siebel Tools, navigate to Screens > Application Upgrader > Application Upgrade Attribute List.
- 2 In the Application Upgrades list, select the record of the successful merge.
- 3 In the Attribute Differences list, click Query.
- 4 In the Attribute Differences list, click in the Conflict field so that a check mark appears.
- 5 Press Enter to run the query.

The query displays a list of all object attributes for which there is a conflict.

- 6 For each record, review the entry in the Resolution field.
- 7 To change the resolution, click in the Override field.

A check mark appears. This changes the value of the attribute in the merged repository.

Avoid overriding conflicts for the following attributes. Visually review these attribute changes in the upgraded application before changing them:

- Left
- Right
- Top
- Height

- Width

Reviewing Deleted Objects in the Repository Merge

Upgrades: All upgrades.

Environments: Development environment only.

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

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

Prerequisite: The repository merge must have been successful. See [“Determining if a Repository Merge was Successful” on page 271.](#)

To generate a list of deleted objects

- 1 In Siebel Tools, navigate to Screens > Application Upgrader > Application Upgrade Object List.
- 2 In the Application Upgrades list, select the record of the successful merge.
- 3 Click Query.
- 4 Enter your query criteria in the Object Differences list:
 - Click in the In Prior Standard field so that a check mark appears.
 - Click in the Added to New Customized field so that a check mark appears.
 - Click in the In Prior Customized field so that a check mark appears. Then click in it again so that no check mark appears.
- 5 Press Enter to run the query.

Deleted objects appear in the Object Differences list. You can filter the objects displayed by using the Top Parent Type and Object Type fields.
- 6 Review the list carefully to determine that deleted objects that have been restored to the merged repository will not have an adverse effect on upgraded applications.

Reviewing Obsolete Objects in the Repository Merge

Upgrades: All upgrades.

Environments: Development environment only.

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

Objects that were available in the Prior Siebel Repository but are not available in the New Siebel Repository are obsolete. After performing the repository merge, you can generate a list of obsolete objects using Siebel Tools. Objects that were available in the Prior Siebel Repository are compared with the objects that are available in the New Siebel Repository.

Prerequisite: The repository merge must have been successful. See [“Determining if a Repository Merge was Successful”](#) on page 271.

To generate a list of obsolete objects

- 1** In Siebel Tools, navigate to Screens > Application Upgrader > Application Upgrade Object List.
- 2** In the Application Upgrades list, select the record of the successful merge.
- 3** Click Query.
- 4** Enter your query criteria in the Object Differences list:
 - a** Click in the In Prior Standard field so that a check mark appears.
 - b** Click in the In New Standard field so that a check mark appears. Then click in the field again so that no check mark appears.
 - c** Click in the In Prior Customized field so that a check mark appears.
 - d** Click in the Attribute field so that a check mark appears. Then click in the field again so that no check mark appears.
- 5** Press Enter to run the query.

All obsolete objects appear in the Object Differences list. You can filter the objects displayed by using the Top Parent Type and Object Type fields.
- 6** Review the list carefully to determine that obsolete objects that have been deleted will not have an adverse effect on upgraded applications.

Reviewing Customized Business Components

Upgrades: All upgrades.

Environments: Development environment only.

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

After upgrading the Siebel Database, thoroughly review the postupgrade configuration to make sure that the object level definitions are preserved as expected.

The access control buscomp migration utility, upgcust, is run on the Prior Customer Repository by the Siebel 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 buscomps are reconfigured by the access control buscomp migration utility. For example, customer buscomps 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 buscomp 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

Reimplement the extension columns on the replacement tables. After you have done this, review the business component definition to verify that it is operating properly.

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. 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, you should 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*.

Generating EIM Temporary Columns After a Repository Merge

Upgrades: All upgrades.

Environments: Development environment only.

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

The repository merge process does not preserve EIM processing columns for custom mappings. After the completion of a successful merge, you must generate the missing custom EIM processing columns in the New Customer Repository.

Prerequisite: The repository merge must have been successful. See [“Determining if a Repository Merge was Successful” on page 271.](#)

To generate EIM temporary columns

- 1 In Siebel Tools, select File > Open Repository, and choose New Customer Repository.
- 2 Navigate to Tools > Upgrade > Generate EIM Processing Columns.
A dialog box displays.
- 3 In the dialog box, click OK to generate EIM processing columns for custom mappings.
- 4 In the Object Explorer window, choose New Customer Repository and verify that the Comment field shows UpgEi mCol .

This indicates that the EIM temporary columns were created successfully.

Regenerating the Repository Definition Files

Upgrades: All upgrades.

Environments: Development environment only.

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

If you have done any development after the development environment upgrade (upgphys) and before upgrading the production environment, use this topic to regenerate the schema.ddl and custrep.dat files.

The development environment upgphys upgrade step creates two repository definition files:

- **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\sch_dict.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 Server installation directory.
 - *DatabasePlatform* is the Siebel Database Server directory name for the database, for example IBM DB2 UDB. The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
 - *SiebelLogdir* is the path to the directory where you want the output log placed (log output directory). The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
- 3 After the command completes, review the output logs for errors. If the log indicates there are errors, contact Siebel Technical Support.

Regenerating the *custrep.dat* File

Use this procedure to regenerate the *custrep.dat* file.

To regenerate the *custrep.dat* file

- 1 On the Siebel Server where the Siebel Database Server files are installed, navigate to the following location:

Windows: *SIEBEL_ROOT*\bin

UNIX: \$SIEBEL_ROOT/bin

- 2 Run the following command:

```
repimexp /a e /u DatabaseOwner /p Password /c "ODBCDataSource" /d TableOwner  
/r "New Customer Repository" /f DBSRVR_ROOT\DatabasePlatform\custrep.dat  
/l SiebelLogDir\exprep.log
```

where:

- *DatabaseOwner* is the Siebel Database administrator account name.
 - *Password* is the Siebel Database administrator account password.
 - *ODBCDataSource* is the ODBC name for connecting to the database. Enclose the name in quotes.
 - *TableOwner* is the Siebel table owner name.
 - *DBSRVR_ROOT* is the absolute path to the Siebel Database Server installation directory. The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
 - *DatabasePlatform* is the Siebel Database Server directory name for the database, for example IBM DB2 UDB.
 - *SiebelLogdir* is the path to the directory where you want the output log placed (log output directory). The example shows Windows path syntax. On UNIX systems, use UNIX path syntax.
- 3 After the command completes, review the output logs for errors. If the log indicates there are errors, contact Siebel Technical Support.

Deleting Unneeded Repository Files

Upgrades: All upgrades.

Environments: Development environment only.

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

When you are confident that the repository has been upgraded successfully, export the New Customer Repository and Prior Customer Repository for safekeeping. You can also delete the following repositories:

- Prior Standard Repository
- New Standard Repository

Safeguarding the New Custom Repository Export File

Upgrades: All upgrades.

Environments: Development environment only.

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

The Siebel Software Configuration utility exports two customized repository files called `custrep.dat` and `schema.ddl`, located in the `DBSRVR_PLTFRM_ROOT` directory on the Siebel Server from which you ran the upgrade. Locate these files and make a backup copy of them.

If you make *any* changes to your custom repository after running the development upgrade Siebel Software Configuration utility, including reapplying custom extensions or modifying EIM mappings in the two previous steps, you must export a new copy of the repository before performing the production upgrade.

Use the script `master_exprep.ucf`, located in `DBSRVR_ROOT`, to export your current repository to the file that you designated as the output file in the Siebel Software Configuration utility. You can use the Siebel Software Configuration utility to invoke the `master_exprep.ucf` script that exports your current repository. Before you export the repository, you should make sure that all Siebel Tools projects are unlocked to promote a stable environment.

The `master_exprep.ucf` script calls the `repimexp` utility, which connects to two separate repositories—the repository whose structure is to be extracted, and the repository whose content is to be extracted. Apart from unusual circumstances, you always extract *both* the structure and content of the same repository.

For information on exporting repositories, see *Configuring Siebel eBusiness Applications*.

Moving the Repository Files

Upgrades: All upgrades.

Environments: Production environment only.

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

Before doing the production upgrade, you must copy the upgraded repository definition files (`schema.ddl` and `custrep.dat`) from the development environment to the production environment.

Prerequisite: If you modified repository objects or schema definitions after completing the development upgrade (`upgphys`), you must regenerate the `schema.ddl` and `custrep.dat` files. See [“Regenerating the Repository Definition Files” on page 278](#).

To move the repository files

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

Windows: `DBSRVR_ROOT\Platform`

UNIX: `DBSRVR_ROOT/Platform`

Where:

`Platform` is the database platform, for example `DBSRVR_ROOT\DB2UDB`.

- 2 Copy the following files:

`custrep.dat`

`schema.ddl`

- 3 In the production environment, put these files in the following location:

Windows: *DBSRVR_ROOT\PI at form*

UNIX: *DBSRVR_ROOT/PI at form*

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 the 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 *Global Deployment Guide*.

Prerequisite: The repository merge must have been successful. See [“Determining if a Repository Merge was Successful” on page 271.](#)

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 283.](#)

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: *SIEBEL_ROOT\OBJECTS\language*

UNIX: *\$SIEBEL_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 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 upgphys 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: *SIEBEL_ROOT\OBJECTS\language*

UNIX: *\$SIEBEL_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 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.

16 Migrating Siebel 6.x Customizations

This chapter contains the following topics:

- “About Migrating to the Siebel Web Client” on page 285
- “About Migrating 6.x Scripts” on page 291
- “About Migrating 6.x Client-Side Interfaces” on page 292
- “Identifying What Will Be Migrated” on page 293
- “Exposing Hidden Properties of Applets and Views” on page 294
- “Running the Siebel Web Client Migration Wizard” on page 295
- “Reviewing Applet and View Migration” on page 298
- “Running the Web Layout Wizard” on page 299
- “Migrating Scripts Attached to Controls” on page 300
- “Migrating Scripts Attached to Applets” on page 302
- “Migrating Business Component, Business Service, and Application Scripts” on page 305
- “Migrating Outbound COM Interfaces” on page 306
- “Migrating Inbound COM Interfaces” on page 307
- “Regenerating the Postmerge Utilities Log” on page 309
- “Reviewing Customized Business Components” on page 310

About Migrating to the Siebel Web Client

Upgrades: Release 6.x only.

Environments: Development environment only.

In Release 7.x, the Siebel Web Client is used as the deployment framework for all Siebel eBusiness 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 eBusiness applications architecture resides on the user computer.

The Siebel 7 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.

Release 7.7 enhancements enable the Siebel Web Client Migration Wizard to perform 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.x customer applications

The Release 7.7 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 paradigms, similar to the Release 6.x 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 7.x 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 eBusiness Applications*.

Example of the Siebel Web Client UI

Figure 5 shows a standard Siebel 6.x Windows Client. Figure 6 on page 288 shows a view that has been migrated from the Windows Client to the Siebel Web Client. This view is running in a browser.

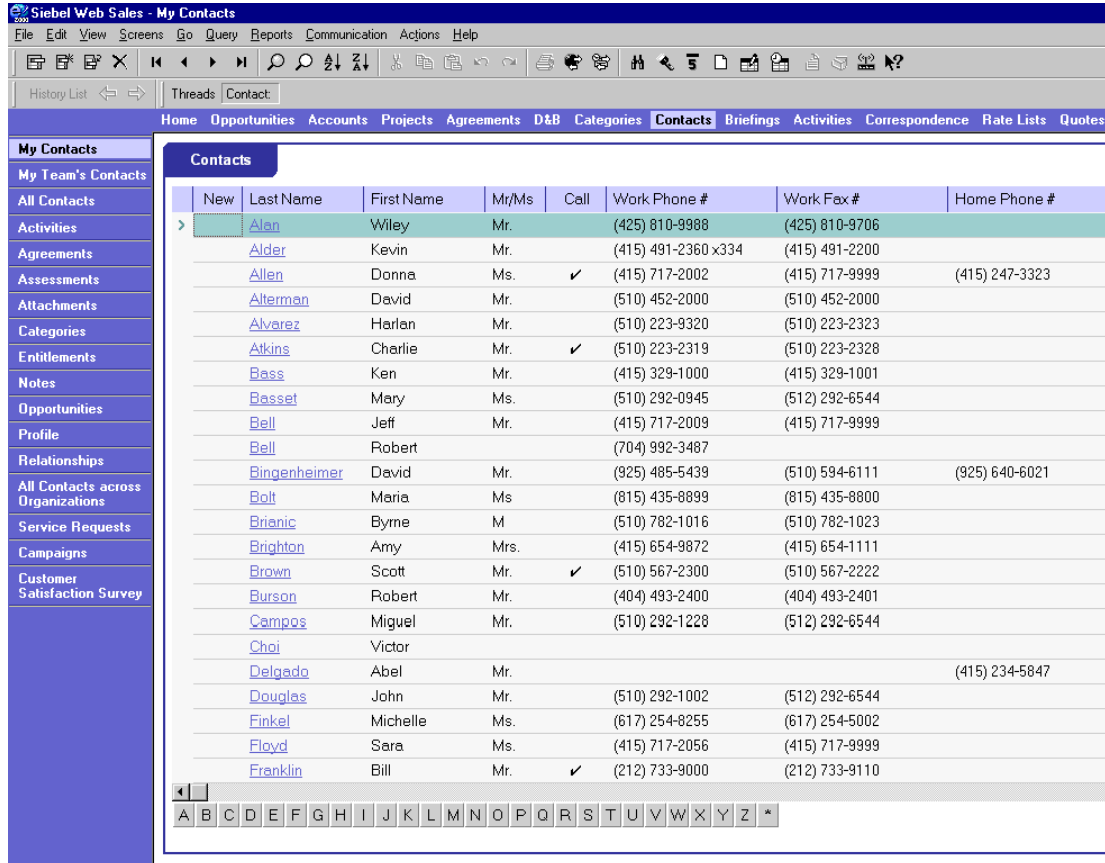


Figure 5. Contact List View—Windows Client

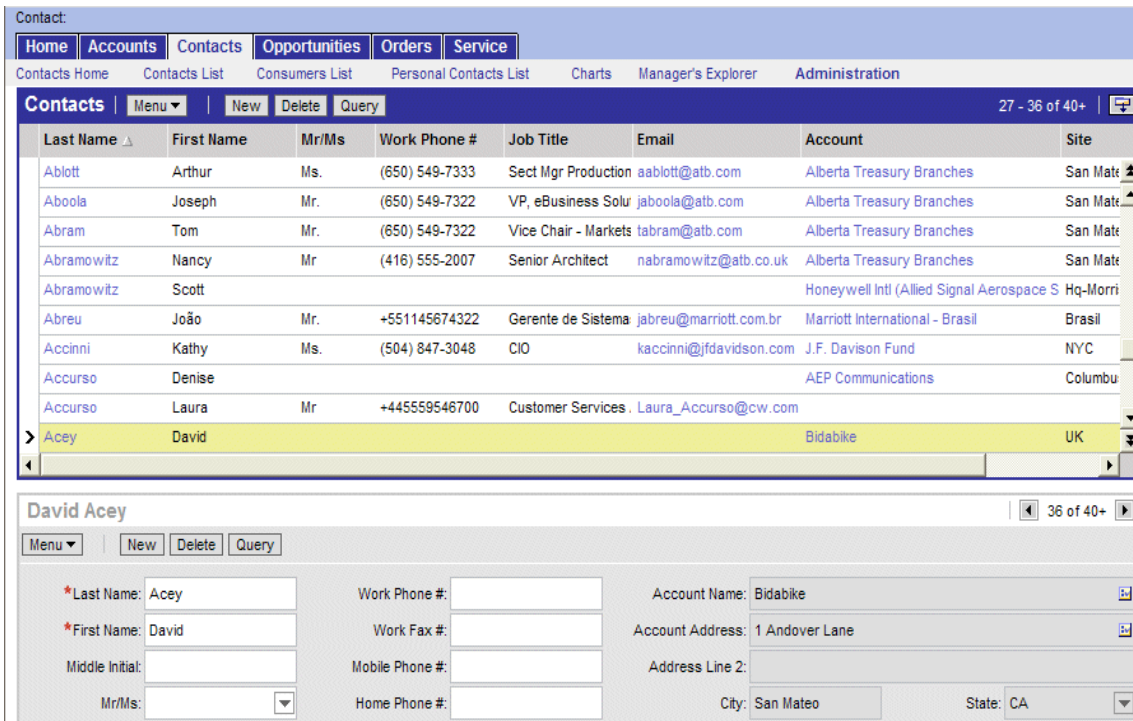


Figure 6. Contact List View—Siebel Web Client

About the Siebel Web Client Migration Wizard

When you upgrade to Release 7.7, 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.
- Remigrate 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 migrate your applets and views is time-consuming and is not recommended.

Model Applets and Views

The Release 7.7 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.

The following list of preconfigured model applets and views is available in this release:

- Model chart applet
- Model form applet
- Model list applet
- Model association list applet
- Model MVG applet
- 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, see *Configuring Siebel eBusiness 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 eBusiness 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.x 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. See [Table 23](#).

Table 23. 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.x Scripts

Upgrades: Release 6.x only.

Environments: Development environment only.

You must migrate scripts on UI objects to server scripts or browser scripts when upgrading to Release 7.7.

Server Scripts

Server scripts execute in the Application Object Manager. They existed in prior releases and continue to be supported in Release 7.7.x and later. 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 Technical Note 386, available on Siebel SupportWeb. For more information about events that are exposed through the DOM, see *Siebel Tools Online Help*.

You must migrate Release 6.x 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.x applet scripts such as Control_Click event handlers to business components
- Rescripting Release 6.x ActiveX controls in browser script
- Rewriting other Release 6.x applet scripts in browser script

To identify the number of 6.x applet scripts to be reviewed or migrated, look at the Object List Editor in Flat mode in Siebel Tools.

You must review Release 6.x 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.x 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, see *Siebel Tools Online Help*.

About Migrating 6.x Client-Side Interfaces

Upgrades: Release 6.x 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.

- **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.x, desktop applications communicated with the Siebel Client through an Automation interface. In Release 7.7.x and later, 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 Dedicated 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.x only.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform the 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 7.7.

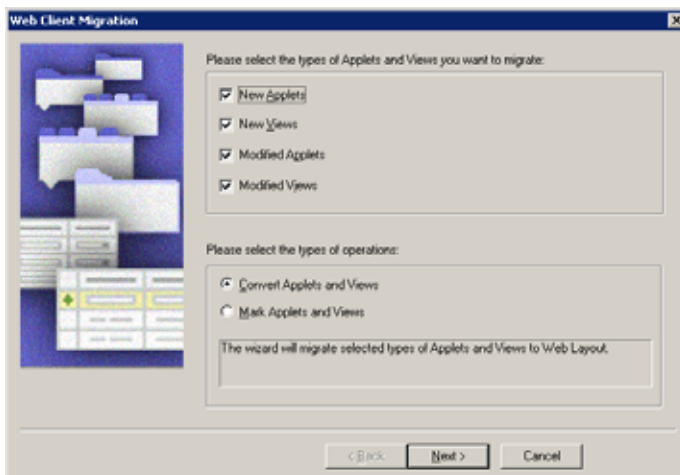
For more information on migrating created or modified applets and views to the Siebel Web Client, see [“About Migrating to the Siebel Web Client” on page 285.](#)

Prerequisite: The repository merge must have been completed successfully. See [“Determining if a Repository Merge was Successful” on page 271.](#)

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.x only.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform the 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 7.7 Web Client, expose these properties to make the migration of your customizations to Release 7.7 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 285.](#)

To expose hidden properties

- 1 Using a text editor, open the tool s.cfg file located in the following directory:
`Siebel Tool sDi r\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.x only.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform the 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 7.7.

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 285.](#)

Prerequisites:

- The repository merge must have been completed successfully. See [“Determining if a Repository Merge was Successful” on page 271.](#)
- Identify model applets and model views. See [“Identifying What Will Be Migrated” on page 293.](#) This prerequisite is optional.
- Expose hidden properties on objects. See [“Exposing Hidden Properties of Applets and Views” on page 294.](#)

Related Topic

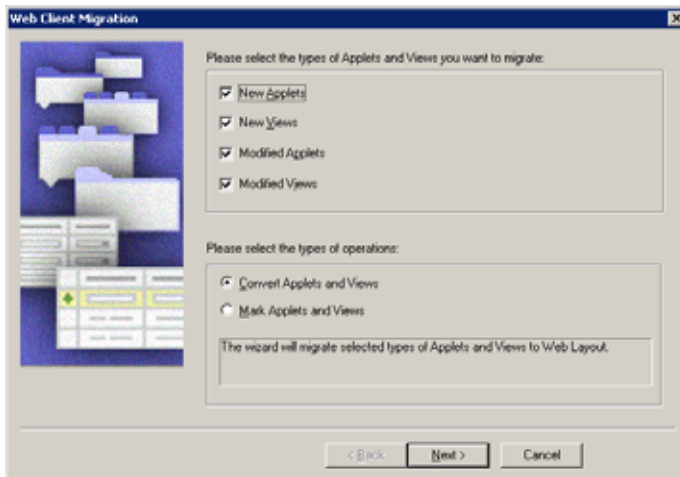
[“Running the Web Layout Wizard” on page 299](#)

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.

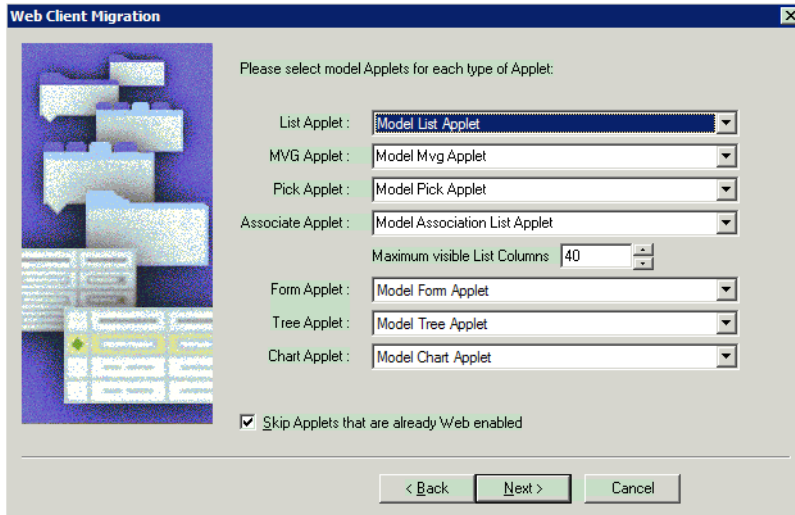
This starts the Siebel Web Client Migration Wizard. The model applet selection screen appears.



- 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.

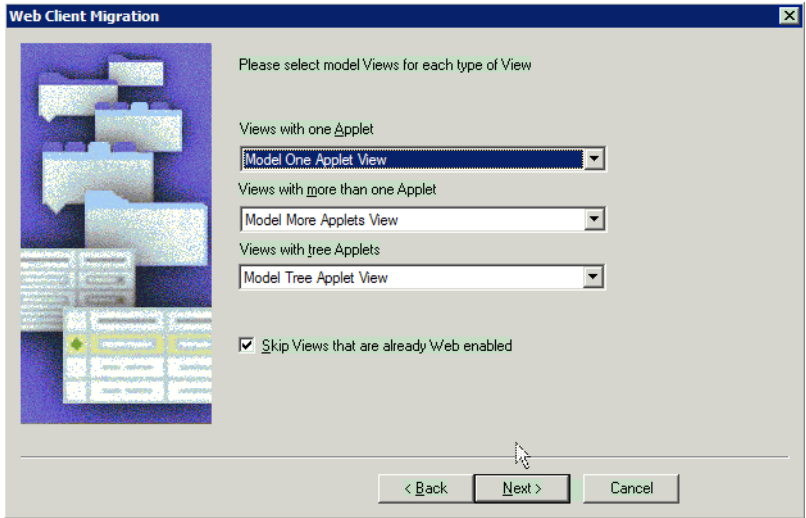
To continue, click Next.

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.x only.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform the 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 285](#).

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.x only.

Environments: Development environment only.

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

If some views or applets did not migrate properly when you ran the Siebel Web Client Migration Wizard to upgrade to Release 7.7, you can remigrate them using the Web Layout Wizard. With the Web Layout Wizard you can migrate applets or views individually or in groups.

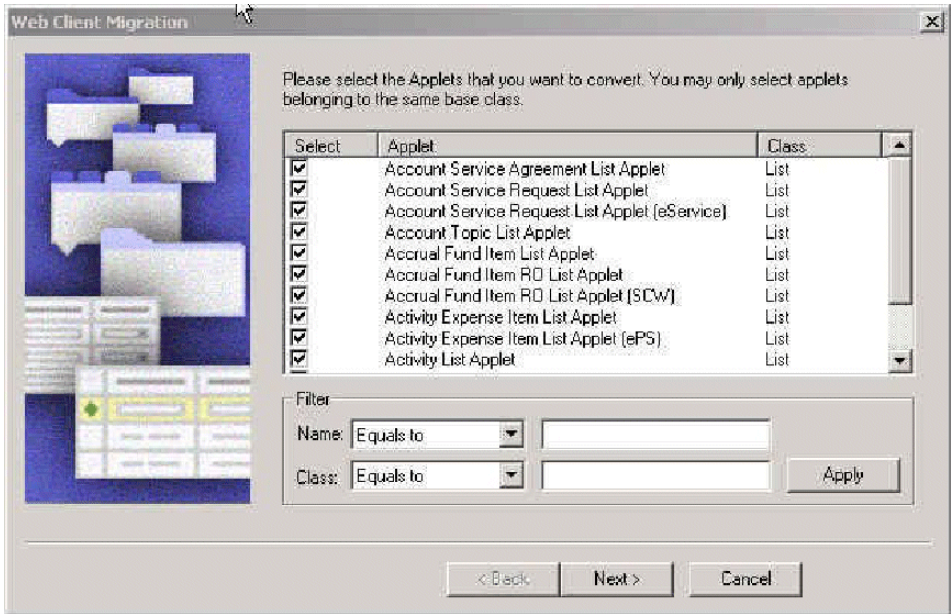
When you select applets for migration, they must be in the same base class. When you select views, they must be of the same type.

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 285](#).

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.x only.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform the 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 7.x, all control methods are available through browser scripting and execute in the high interactivity applications only. Table 24 provides a list of the control methods that are supported by browser script. For more information, see *Siebel Tools Online Help*.

Table 24. 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 7 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.

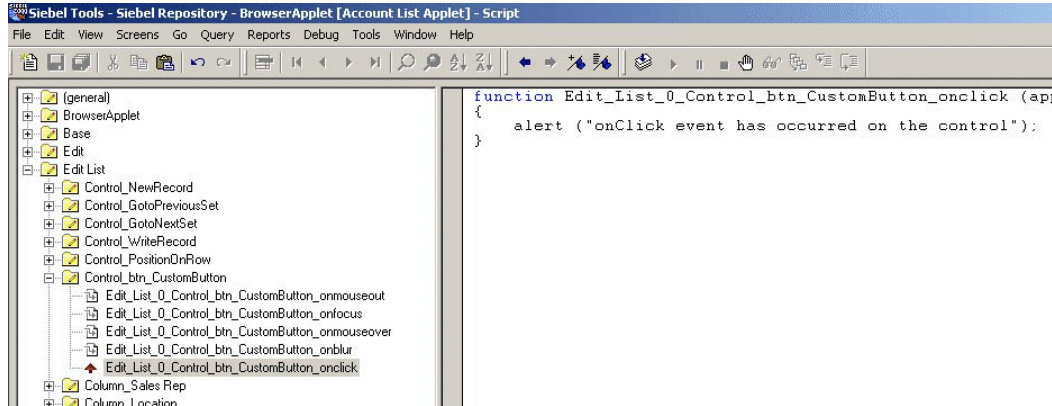


Figure 7. DOM Events When MiniButton Control Added to Applet

For more information on migrating scripts, see [“About Migrating 6.x Scripts”](#) on page 291.

Related Topics

- “Migrating Scripts Attached to Applets” on page 302
- “Migrating Business Component, Business Service, and Application Scripts” on page 305

Migrating Scripts Attached to Applets

Upgrades: Release 6.x only.

Environments: Development environment only.

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

When upgrading to Release 7.7, you must do one of the following:

- Migrate Release 6.x 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.x applet scripts such as Control_Click event handlers to business components
 - Rescripting Release 6.x ActiveX controls in browser script
 - Rewriting other Release 6.x applet scripts in browser script
- Rewrite Release 6.x scripts as browser scripts.

To identify the number of 6.x applet scripts to be reviewed or migrated, look at the Object List Editor in Flat mode in Siebel Tools.

As shown in [Figure 8 on page 303](#), applet scripts implemented in Siebel 6.x 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

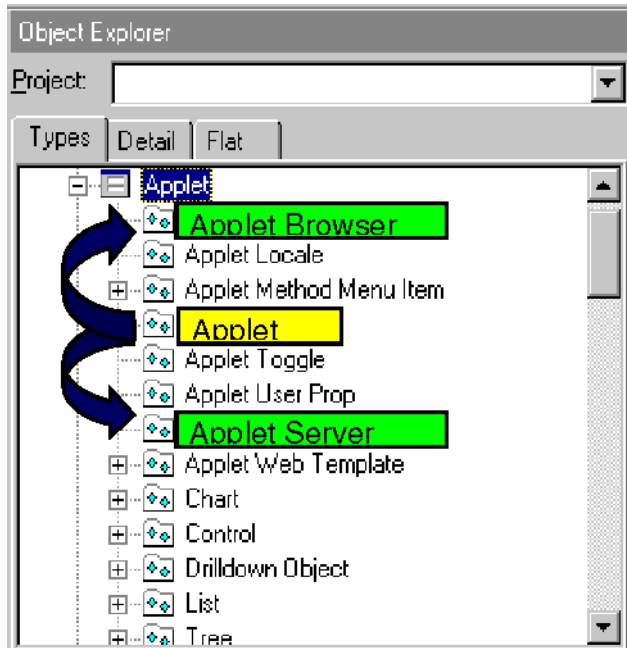


Figure 8. Movement of Applet Scripts

In Release 7.x, 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.x applet events. Corresponding browser or server script event handlers might be available, but the scripts are not automatically migrated. Table 25 depicts Siebel 6.x events and corresponding Release 7.x events available in either browser or server script.

Table 25. Siebel 6.x Applet Events and Release 7.x Equivalents

Siebel 6.x 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.x events and Release 7.7 events (for example Applet_LostFocus event), you must evaluate and reimplement the functionality of the Siebel 6.x 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 Application.Navigate or PreNavigate event.

The following list of applet-level methods are no longer available (obsolete) in this release:

- ActiveControl
- FindControl
- FindActiveXControl
- GotoControl
- PostChanges
- Drilldown

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.

For more information on migrating scripts, see [“About Migrating 6.x Scripts” on page 291](#).

Related Topics

[“Migrating Scripts Attached to Controls” on page 300](#)

[“Migrating Business Component, Business Service, and Application Scripts” on page 305](#)

Migrating Business Component, Business Service, and Application Scripts

Upgrades: Release 6.x only.

Environments: Development environment only.

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

The majority of application, business components, and business services scripts implemented in Siebel 6.x can remain unchanged and should 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 FAQ 1562 on Siebel SupportWeb.

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 7.x, see *Siebel Object Interfaces Reference* and Technical Note 386 on Siebel SupportWeb.

For more information on migrating scripts, see [“About Migrating 6.x Scripts” on page 291](#).

Related Topics

[“Migrating Scripts Attached to Controls” on page 300](#)

[“Migrating Scripts Attached to Applets” on page 302](#)

Migrating Outbound COM Interfaces

Upgrades: Release 6.x only.

Environments: Development environment only.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform the 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;

// Pl ace some text in the fir st cell

Sheet. Acti veSheet. Cel l s(1, 1). Val ue = "Thi s i s row 1 col umn A";

// Save the sheet.

Sheet. SaveAs("C: \\TEST. XLS");

// Cl ose Excel and qui t.

Sheet. Appl i cati on. Qui t();
```

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 on migrating client-side interfaces, see [“About Migrating 6.x Client-Side Interfaces” on page 292](#).

Related Topic

[“Migrating Inbound COM Interfaces” on page 307](#)

Migrating Inbound COM Interfaces

Upgrades: Release 6.x only.

Environments: Development environment only.

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

Release 7.x 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.x, but it does not expose applet or control objects.

The Release 7.x 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 ("SiebelMessage")
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 on migrating client-side interfaces, see ["About Migrating 6.x Client-Side Interfaces" on page 292](#).

Related Topic

["Migrating Outbound COM Interfaces" on page 306](#)

Regenerating the Postmerge Utilities Log

Upgrades: Release 6.x only.

Environments: Development environment only.

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

When upgrading from Release 6.x to Release 7.7, 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
```

Prerequisite: 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.x only.

Environments: Development environment only.

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

After upgrading the Siebel Database to Release 7.7, thoroughly review the postupgrade configuration to make sure that the object level definitions are preserved as expected.

The access control buscomp 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 buscomps are reconfigured by the access control buscomp migration utility. For example, customer buscomps 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 buscomp 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, you should 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*.

17 Running the Postmerge Utilities

This chapter contains the following topics:

- “Running the Postmerge Utilities” on page 313
- “Troubleshooting Postmerge UI Problems” on page 314
- “Verifying Business Address Applet Configuration” on page 315
- “Reviewing Grid-Based Applets After the Merge” on page 316
- “Reviewing UI Navigation After the Merge” on page 317
- “Reviewing Multi-Value Group (MVG) Shuttle Applets After the Merge” on page 318
- “Revising Rich Text Controls After the Merge” on page 320
- “Reviewing New Aggregate Categories After the Merge” on page 321
- “Revising Visibility Filters to Display Correctly After the Merge” on page 322
- “Assigning a Category and Type to Chart Views After the Merge” on page 323
- “Assigning a Category and Type to Explorer Views After the Merge” on page 324
- “Setting Up Navigation to Inaccessible Detail Views After the Merge” on page 324
- “Eliminating Obsolete UI Fields After the Merge” on page 325
- “Reviewing UI Objects Affected by ICL After the Merge” on page 326
- “Reviewing Required Fields in the UI After Merge” on page 326
- “Setting Label Alignment for Text Fields” on page 327
- “Assigning an Item Identifier to Web Template Items” on page 327

Running the Postmerge Utilities

Upgrades: All upgrades.

Environments: Development environment only.

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

After the merge is completed successfully, run the postmerge utilities. These utilities make the following changes to user interface objects in the merged repository:

- Convert flow-based form applets to grid-based.
- Verify that new or customized screens, views, and applets from the Prior Customer Repository are configured correctly and can be accessed in the UI.
- Revise multi-value group (MVG) applets so that they are shuttle-enabled.

- If you selected Incorporate Custom Layout (ICL) during the repository merge, the utilities list the UI objects that were affected.
- Lists UI objects that have missing required fields.

You can rerun the postmerge utilities as needed. If the postmerge utilities encounter a problem, do the following:

- Try to fix the problem in the merged repository, and then rerun the utilities.
- If you cannot fix the problem, restore the backup of the postmerge Siebel Database or repository. Then, resolve the problem and run the utilities again.

The postmerge utilities take a significant amount of time to run.

Prerequisite. Before running the postmerge utilities, verify you have a backup of the Siebel Database or the Siebel Repository that was done after the repository merge.

To run the postmerge utilities

- 1 If you are rerunning the postmerge utilities and want to save the existing log, rename the log. The path to the log is as follows:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

If you do not rename the log, it will be overwritten.

- 2 In Siebel Tools, navigate to Screens > Application Upgrader > Application Upgrade Object List.
- 3 In the Application Upgrades list, select the record of the merge.
- 4 In the Application Upgrades list, right-click and select Launch Post Merge utility from the pop-up menu.

A dialog box appears and displays the postmerge utilities log. When the utilities have finished, a message showing completion displays in the log.

Troubleshooting Postmerge UI Problems

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

Environments: Development environment only.

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

Basic Troubleshooting Guidelines

If views or screens do not display, do the following:

- Verify that the user has been assigned the correct responsibilities.
- In Siebel Tools, verify that the screen view and its parents have the Display In Site Map property, and the Display In Page property set to TRUE.

- Verify that the screen view and its parents have the Viewbar Text and Menu Text properties filled in.
- If an applet does not display all the fields or controls after upgrade, check in Siebel Tools for a Web template of the same name, but appended with -Expanded and specify this Web template for the applet. These templates provide additional placeholders for mapping fields and controls.
- In Siebel 7.5, the Business Address business component was replaced with the CUT Address business component. If you are upgrading from a release prior to 7.5, you must modify all links to Business Address applet so they point to CUT Address.

About the Postmerge Utilities Log

After the repository merge, you must run the postmerge utilities. These utilities do the following:

- Validate UI components to verify they were migrated correctly to the new repository.
- Modify UI objects to implement new UI features. For example, they modify form applets and multi-value group applets to conform to the UI standard introduced at Release 7.7.
- Verify that customized UI objects are configured correctly.

The postmerge utilities log lists the actions performed by the postmerge utilities. The log contains the following types of messages:

- **STATUS.** These messages provide information on the specific things that the postmerge utilities did. No action is required, so you can ignore these messages.
- **INFO.** These messages provide information on the specific things that the postmerge utilities did. No action is required, so you can ignore these messages.
- **WARNING.** These messages provide information on UI objects that may be incorrectly configured, so you should review them.
- **ERROR.** These messages indicate that a problem has been found that must be corrected.

The postmerge utilities log is located here:

`SIEBEL_TOOLS_INSTALL_DIR\repatch\log\reputility.log`

Verifying Business Address Applet Configuration

Upgrades: Release 7.0.x only. This topic applies only to Siebel Industry application (SIA) upgrades. It does not apply to Siebel Business Application (HOR) upgrades.

At release 7.5, the Business Address business component was replaced with the CUT Address business component. You must modify all links to the Business Address applet so they point to CUT Address.

Reviewing Grid-Based Applets After the Merge

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Development environment only.

Reputility log section: Start Invalid Web Template Item Mapping Clean-up.

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

The postmerge utilities convert many form applets to grid-based layout in Release 7.7.

If a Prior Customer Repository applet that will be converted to grid-based layout has been customized by adding new fields or controls, the utilities put these fields and controls at the bottom of the applet in the New Customer Repository. After the repository merge, you must reposition these fields and controls.

ICL upgrades only: If Incorporate Custom Layouts (ICL) was selected for the repository merge, many form applets are converted to grid-based layout. The postmerge utilities then deactivate the grid-based layout and activate the Prior Customer Repository, flow-based form of the applet. For more information on Incorporate Custom Layouts (ICL), see [“About the Incorporate Custom Layout \(ICL\) Upgrade Option” on page 248.](#)

The reputility.log, lists applets that were converted by the postmerge utilities. Review the applets listed in the section of the log referenced above and revise layouts as needed.

The log is located here:

`SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log`

[Table 26](#) lists common issues and corrective actions for grid-based applets.

Table 26. Common Grid-Based Applet Issues

Log Examples	What To Do
STATUS:: Succeed Update AWTI List Mgmt Lists Entry Applet Base Status	The utility mapped a control to a different location. Action: Review the modified Applet. Verify that the new location for the control works.
STATUS:: Succeed Delete AWTI List Mgmt Lists Entry Applet Base Status ESN	To avoid possible overlapping, the utilities deleted a locale record for an applet Web template item. Action: Review the modified Applet Web Template and re-create the locale record.
WARNING:: Grid -> Grid merge/upgrade, Skip	Only flow-based form applets are converted to the grid-based Web template. Action: None required.

Table 26. Common Grid-Based Applet Issues

Log Examples	What To Do
WARNING:: Upgrade Ancestor "Account Form Applet" Not Found in New Siebel Repository, Skip	The utilities did not find the upgrade ancestor's applet Web template in the New Siebel Repository. Action: Review the applet for invalid Web template items.
WARNING:: Applet Web Template Not Found in New Siebel Repository, Skip	The utility is trying to find the Applet Web Template from New Siebel Repository, but it is not found. Action: Review the applet for invalid Web template items.
WARNING:: No List Column Found, Applet "Program Expenditure List Applet", AWT "Edit", AWTI "Type", Control "Type", Skip	No List Column found for the Applet Web Template Item. Action: Delete the applet Web template item.
WARNING:: No Control Found, Applet "Expense Item Entry Applet", AWT "Edit", AWTI "Site", Control "Site", Skip	No Control is found for the Applet Web Template Item. Action: Delete the applet Web Template.
WARNING:: Button Control, Applet "Expense Item Entry Applet", AWT "Edit", AWTI "NewQuery", Control "NewQuery", Invoke Method "NewQuery", Skip	If a control has an Invoke Method action defined on it, the utility treats it as a button and will not remap it. Action: Review the applet and verify that the button control displays correctly.

Reviewing UI Navigation After the Merge

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

Environments: Development environment only.

Reputility log section: User Interface Navigation Upgrade.

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

The postmerge utilities analyze the repository and verify that objects referenced in screens, views and applets are defined correctly. The reputility.log lists objects that need to be modified. Review the section of the log referenced above and make the needed revisions.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

Table 27 shows examples of common issues and corrective actions.

Table 27. Common UI Navigation Issues

Log Examples	What To Do
WARNING:: Project <i>projname</i> is not found in this Repository	The postmerge utilities exclude certain Tools projects from UI Navigation cleanup. One of the excluded projects was not found in the repository. Action: None required.
WARNING:: Ignoring Screen View Record. View Definition Not Found [View: Account Briefing View]	The utility is unable to read the view definition to determine where it should display. Action: Remove the invalid screen view record that references the invalid view.
WARNING:: Error Writing Category Record, Ignoring Changes [Name: <i>catname</i>]	The utility could not update or insert a record in the Siebel Database. Possible causes are that the record already exists or there is a database access problem. Action: Verify that a duplicate <i>catname</i> record does not already exist, and then check database access.
WARNING:: Error Writing Screen View Record, Ignoring Changes [View: <i>viewname</i>]	The utility could not create a screen view record. Possible cause is that the category does not exist. This error is often a consequence of an error updating or inserting a category record. Action: Verify that the screen view category exists, and then check database access.

Reviewing Multi-Value Group (MVG) Shuttle Applets After the Merge

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Development environment only.

Reputility log section: Multivalued Group Shuttle Applet Upgrade.

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

As of Release 7.7, MVG applets are shuttle-enabled by default. The postmerge utilities shuttle-enable MVG applets in the New Customer Repository. This includes MVG applets from the Prior Customer Repository that you created or customized.

MVG applets must have a specific configuration in order to be enabled as MVG shuttle applets. For information on creating and managing MVG shuttle applets, see *Configuring Siebel eBusiness Applications*.

In the reputility.log, the utilities list MVG applets that were converted. Review the section of the log referenced above and resolve any problems encountered during conversion.

The log is located here:

`SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log`

Table 28 lists common issues and corrective actions for MVG applets.

Table 28. Common MVG Applet Issues

Log Entry Example	What To Do
<p>WARNING:: [APPLET: Account Address Mvg Applet (NB)] MVG Applet is Inactive. Ignoring Applet.</p>	<p>This applet is inactive. Action: No action required.</p>
<p>WARNING:: [APPLET: Primary Employee Mvg Applet] [Applet Web Template: Base] Applet Web Template is Inactive. Ignoring Applet Web Template.</p>	<p>The applet Web template is inactive. Action: No action required.</p>
<p>WARNING:: [APPLET: FINS Application Contact Mvg Applet - ACAPS] [CONTROL METHOD INVOKED: ExecuteQuery] Has a Non Standard Control Type.</p>	<p>The utility is trying to map an existing Go (ExecuteQuery) button to the Edit List mode to enable Popup Inline Query. However, the Go button does not have the correct attributes. Action: Revise the control definition and map it to the Edit List mode.</p>
<p>WARNING:: [APPLET: State Model - State Mvg Applet] [CONTROL METHOD INVOKED: UndoQuery] Has a Non Standard Control Type.</p>	<p>The utility is trying to map an existing Cancel (UndoQuery) button to the Edit List mode to enable Popup Inline Query. However, the Cancel button defined does not have the correct attributes. Action: Revise the control's definition and map it to the Edit List mode.</p>
<p>WARNING:: [APPLET: LOY Account Address Assoc Applet] [CONTROL: CancelQuery] Control is Inactive. Please inspect and Reactivate.</p>	<p>The utility is trying to map an existing Cancel (UndoQuery) button to the Edit List mode to enable Popup Inline Query. However, the Cancel button defined is inactive. Action: Redefine a Cancel button and map it to the Edit List mode.</p>
<p>WARNING:: [APPLET: LOY Account Address Assoc Applet] [CONTROL: ExecuteQuery] Control is Inactive. Please inspect and Reactivate.</p>	<p>The utility is trying to map an existing Go (ExecuteQuery) button to the Edit List mode to enable Popup Inline Query. However, the Go button defined is inactive. Action: Redefine a Go button and map it to the Edit List mode.</p>

Table 28. Common MVG Applet Issues

Log Entry Example	What To Do
<p>WARNING:: [APPLET: Account Address Mvg Applet] [Applet Web Template: Edit List]</p> <p>[Applet Web Template Item: ExecuteQuery] Applet Web Template Item is Inactive. Please inspect and Reactivate.</p>	<p>The utility is trying to map an existing Go/Cancel button to the Edit List mode to enable Popup Inline Query. However, the mapping already exists, but is marked inactive.</p> <p>Action: Activate the mapping (Applet Web Template Item) in the Edit List mode and test.</p>
<p>WARNING:: [APPLET: Account Address Mvg Applet]</p> <p>[CONTROL: UndoQuery] Applet Web Template Item occupying Item Id 108. Cannot Map Control UndoQuery</p>	<p>The utilities tried to map a Cancel button to the default location 108 in Edit List mode. However, another control is already mapped to this location.</p> <p>Action: Move the control at location 108 to another location, and then map the Go button to location 108.</p>
<p>WARNING:: [APPLET: Assoc Data Type Applet] Association List</p> <p>Applet contains both Base and Edit List. Manual review needed.</p>	<p>An MVG applet definition specifies both a base and edit-list Web template. At Release 7.7, the UI standard is that when MVGs first display, they are editable.</p> <p>Action: For MVGs that users can edit, verify that the Web template specified for base mode and edit-list mode are the same. If not, change the base mode Web template so that it is the same as the edit-list template.</p>
<p>WARNING:: [APPLET: Activity Order Mvg Applet] [CONTROL: ExecuteQuery]</p> <p>Control is at an unexpected location. Expected Location is 107</p>	<p>The utilities tried to map a Go button to the default location 107 in Edit List mode. This is part of enabling Popup Inline Query. However, another control is already mapped to this location.</p> <p>Action: Move the control at location 107 to another location, and then map the Go button to location 107.</p>

Revising Rich Text Controls After the Merge

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Development environment only.

Reputility log section: Issue 1: Rich Text Controls (RTC) That Need to Have Properties Reconfigured.

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

As of Release 7.7, the configuration of rich text controls (RTCs) changed. The user properties called RTC Graphic Field and RTC Link Field were moved from the Applet User Properties to the Control User Properties of the Body field.

The postmerge utilities review the repository and verify that rich text controls are defined correctly. The reputility log lists the controls that need to be modified. Review the section of the log referenced above and make the needed revisions.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

To revise RTC definitions

- 1 Start Siebel Tools. Set the Object Explorer to Flat.
- 2 Navigate to Views > Options > Object Explorer. Under applet, verify that Applet User Prop and Control User Prop are check-marked.
- 3 Refer to the reputility.log and query for one of the listed applets.
- 4 In the Object Explorer, select Applet User Prop.
If the user RTC Graphic Field and RTC Link Field user properties are marked Inactive, no further action is required.
- 5 For the following user properties, write down the value—active RTC Graphic Field and RTC Link Field user property:
 - RTC Graphic Field. The value for RTC Graphic Field typically is Body Field Graphic.
 - RTC Link Field. The value for RTC Link Field typically is Body Field Link.
 - RTC Body Field. The value is the control name.
- 6 For the applet, select Control > Control User Prop.
- 7 In the Controls list, query for the value you wrote down for RTC Body Field. This is the control name.
- 8 For the control, select Control User Prop.
- 9 Define the following control user properties on the control. Assign the values you wrote down for the applet user properties:
 - RTC Graphic Field
 - RTC Link Field

Reviewing New Aggregate Categories After the Merge

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Development environment only.

Reputility log section: Issue 2: New Aggregate Category Records That Should Be Renamed.

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

The following new properties were added to the Screen View object in Siebel Tools to support new UI navigation introduced in Release 7.7:

- Type (Aggregate Category, Aggregate View, Detail Category, Detail View)
- Parent Category
- Category Name
- Category Default View
- Display in Page
- Display in Site Map

For a description of these properties, see *Configuring Siebel eBusiness Applications*.

If you have created new views or have modified existing views, the postmerge utilities create new Aggregate Category records to support the new properties. The utilities name the new aggregate category records *busobj_name* List. For example, a new aggregate category record for the eEvents screen would be named eEvents List.

The reputility log lists the category records that were created by the postmerge utilities. Review the section of the log referenced above and make the following revisions as needed to the listed objects:

- Revise the Viewbar Text and Menu Text properties in all installed languages as required.
- Verify that the navigation hierarchy, including sequence numbers, is correct.

The log is located here:

`SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log`

Revising Visibility Filters to Display Correctly After the Merge

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Development environment only.

Reputility log section: Issue 3: Views that need an applet in View Web Template Item ID 1.

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

In Release 7.7, the Show menu was replaced by a visibility filter. This drop-down menu lists alternative views for displaying records.

The postmerge utilities review the repository and verify that filters are defined correctly. The reputility log lists screens that have incorrectly defined filters. The most common problem is that none of the views has an Item Identifier of 1, which prevents the filter from displaying. Review the section of the log referenced above and make needed revisions.

The log is located here:

`SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log`

To revise visibility filters to display correctly

- 1 Query for one of the views listed in the reputility.log.
- 2 For the view, select Base > View Web Template Item.
- 3 Set the Item Identifier for one of the applets to 1.
- 4 Refine the query to display the Parent Category listed in the log.

Which applet has the Item Identifier set to 1 is not important. Verify that only one applet has the Item Identifier set to 1.

Assigning a Category and Type to Chart Views After the Merge

Upgrades: Release 7.x not using ICL.

Environments: Development environment only.

Reputility log section: Issue 4: Chart Views Needing Migration to Aggregate Type.

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

The Chart menu item is relocated as of Release 7.7 and displays in the applet header. To ensure that the Chart menu item is located correctly, all the relevant charts for a view must be assigned to the same Aggregate Category. Also, each chart view must be of type Aggregate View. If the chart view is not of type Aggregate View, the chart menu item displays in a nonstandard position in the applet header.

The postmerge utilities review the repository and verify that Chart views have been defined correctly. The reputility log lists the screen views that require revision. Review the section of the log referenced above and make the needed revisions.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

To assign a category and type to chart views

- 1 Set Siebel Tools Object Explorer to Types.
- 2 Query for the screen.
- 3 In the Object Explorer, select Screen View.
- 4 Query for the screen views listed in the log.
- 5 Verify that all the views are assigned to the same category (Category Name).
The Category Name can be null.
- 6 Verify that all the views are of type Aggregate View.

Assigning a Category and Type to Explorer Views After the Merge

Upgrades: Release 7.x not using ICL.

Environments: Development environment only.

Reputility log section: Issue 5: Explorer Views Needing Migration to Aggregate Type.

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

As of Release 7.7, explorer views are defined as Aggregate Views. In addition, explorer views use a new Web template, Tree 2.

The postmerge utilities review the repository and verify that explorer views are defined correctly. The `reputility.log`, lists the explorer views that require revision. Review the section of the log referenced above and make the needed revisions.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

To assign a new Web template to a view

- 1 In Siebel Tools, query for a view name listed in the log.
- 2 In Object Explorer, select View Web Template.
- 3 In View Web Templates, change the Web Template to Tree 2.
- 4 Repeat for all the views listed in the log section.

To assign a new type to a screen view

- 1 In Siebel Tools, query for the screen.
- 2 In Object Explorer, select Screen View.
- 3 For the screen views listed in the log, change the Type to Aggregate View.
- 4 Repeat for each screen listed in the log section.

Setting Up Navigation to Inaccessible Detail Views After the Merge

Upgrades: Release 7.x not using ICL.

Environments: Development environment only.

Reputility log section: Issue 6: Categories Where Parent Applets Are Missing Drilldowns to a Detail View.

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

The postmerge utilities verify that you can navigate from a screen to all the screen detail views listed in the parent category for the screen. If any of the screen's detail views are not accessible, the utilities list the screen name, parent category, and the Aggregate View in the log.

In many cases, the problem is caused by a missing or incorrectly defined drilldown in a list applet in the view shown in the log. The missing drilldown prevents the user from navigating to a view containing third-level view tabs that provide access to all the detail views.

Review the section of the log referenced above and make needed revisions.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log
```

To set up navigation to inaccessible detail views

- 1 In Siebel Tools, navigate to the screen listed in the log. In Object Explorer, select Screen View.
- 2 In the Screen Views List, query for the following:
 - Type = Detail View
 - Parent Category = the Parent Category listed in the log
- 3 Start the application and navigate to the screen.
- 4 Try to navigate to the detail views listed in the query.

TIP: Use the Web Layout Editor in Tools to identify a detail view containing third-level view tabs that provide navigation to all the detail views. Verify that you can navigate to this view from a drilldown in the screen.
- 5 When you have identified the inaccessible detail view containing third-level view tabs, review the drilldown definitions in Siebel Tools for the list applet in the screen. Define a drilldown to the detail view if one does not exist.

Eliminating Obsolete UI Fields After the Merge

Upgrades: Release 7.x using ICL.

Environments: Development environment only.

Reputility log section: Issue 7: Fields deprecated from business components.

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

Some of the business component fields in your installed release may be obsolete in the new release. If you have customized existing applets or created new ones, and you selected ICL during the merge, the UI may contain obsolete fields or controls. If a business component field is not available for an applet field or control after the merge, the field or control does not display.

The reputility log lists the applets that contain obsolete business component fields. Review the section of the log referenced above and revise applet definitions and layouts as required.

The log is located here:

`SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log`

Reviewing UI Objects Affected by ICL After the Merge

Upgrades: Release 7.x using ICL.

Environments: Development environment only.

Reputility log section: Issue 8: List of the items affected by PCL.

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

If you selected Incorporate Custom Layout (ICL) for the merge, the postmerge utilities list all the UI objects that were affected by the ICL feature in the reputility log. Use this list to identify the screens, views, and applets you want to review when testing the UI.

The list has two parts:

- **UI elements changed between prior Siebel Release and current Siebel Release.** This part lists the screens, views, and applets that have UI changes in the new release. If you selected ICL during the merge, these changes are not reflected in the UI. Instead, the UI for the release you are upgrading from has been preserved.
- **UI elements changed between prior Siebel Release and prior customer implementation.** This part lists your customizations. The list includes screens, views, and applets you modified as well as those you created. These customizations are included in the merged repository. If you selected ICL for the merge, the UI for the customizations is the same as the release from which you are upgrading.

The log is located here:

`SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reputility.log`

Reviewing Required Fields in the UI After Merge

Upgrades: Release 7.x using ICL.

Environments: Development environment only.

Reputility.log section: Issue 9: List of Required Fields Missing from the UI.

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

If you select Incorporate Custom Layout (ICL) during the repository merge, the Web templates for the UI you are upgrading from are used to display the UI after the merge. The Web templates for the new release are not used.

The Web templates used may not contain all the fields required by the new release. For example, a required field may be missing from the Web template used to display the Contacts applet. If you try to create a new record, the database will reject it because the record does not contain all the required fields.

This section of the reutility log lists all the applets that have missing required fields. Use the Web Layout Editor in Siebel Tools to add the required fields.

The log is located here:

```
SIEBEL_TOOLS_INSTALL_DIR\reppatch\log\reutility.log
```

Setting Label Alignment for Text Fields

Upgrades: Release 7.x using ICL.

Environments: Development environment only.

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

If you selected Incorporate Custom Layout (ICL) for the repository merge and chose Label On Top, you must edit the style sheet (main.css) to set label alignment.

To set label alignment for text fields

- 1 Using a text editor, open the main.css file in the Siebel Tools installation directory:

```
Windows: \publi c\<lang>\files\main.css
```

```
UNIX: /publi c//lang/files/main.css
```

where

lang is the installed language, for example enu.

- 2 Search for mceLabel2.

This is a section containing several entries. Verify that the section begins .mceLabel2, not .mceLabel.

- 3 In the .mceLabel2 section, set the following values as shown:

- vertical-align : top
- text-align: left
- font-weight : bold

- 4 Save the file and exit.

Assigning an Item Identifier to Web Template Items

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

Environments: Development environment only.

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

All View Web Template items must have an item identifier. In Siebel Tools, query View Web Templates for a null Item Identifier. Replace any null Item Identifiers with a number. If an item is part of a group of items that have the same grandparent (GParent), the number must be unique in the group.

18 Postupgrade Tasks for a Development Environment

This chapter contains the following topics:

- “Reapplying Custom Extensions and FK Columns on Obsolete Tables” on page 329
- “Resolving Business Component and Join Conflicts” on page 331
- “Deleting Duplicate EIM Mappings” on page 332
- “Exporting the New Custom Repository Export File” on page 333
- “Regenerating the Database Template File” on page 334
- “Extracting Developers or Siebel Tools Clients” on page 334
- “Resetting Database Server Configuration Parameters” on page 335
- “Upgrading to the Symbolic String Model” on page 335

Reapplying Custom Extensions and FK Columns on Obsolete Tables

Upgrades: All upgrades.

Environments: Development environment only.

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

In the current release, several tables are obsolete or have been replaced by new tables. If you added extension columns or foreign key columns to tables that are obsolete in the current release, you must reapply these changes to the new tables.

The upgrade process generates a report that you can review for information about obsolete tables. This report, `xtndobstbl . txt`, lists custom columns in obsolete tables.

Several tables used in prior versions of Siebel eBusiness applications are no longer used, or have been replaced by new tables in Release 7. The new configuration uses the new tables. You need to reapply the extensions that you added to previous versions of the tables to the new tables.

The Database Server Configuration Wizard generated a report that you must review for information about tables that are either obsolete or no longer in use for Release 7. This report, `xtndobstbl . txt`, lists extension columns that reside on obsolete tables, and therefore need to be moved to alternate tables. Each table is listed with one of three status codes:

- **Not Used.** These tables are not used in Release 7, but you can continue to use them; for example, `S_ORG_PROD` or `S_PERIOD_BU`. These tables are supported as is (for instance, with docking or EIM).
- **EOL (end of life).** These tables are not used in Release 7, and the recommendation is that you delete them since they will not be supported in future releases; for example, old `S*_IF` tables.

- **Inactive.** These tables have been discontinued, and are not supported in Release 7. Extension columns that reside on inactive tables must be moved to alternate tables; for example, S_EMPLOYEE should be moved to S_CONTACT, S_USER, or S_EMP_PER.

If no tables are listed in xtndobstbl . txt, no action is required. If this file lists any tables, their custom extensions must be reapplied to Siebel eBusiness applications Release 7 tables using Siebel Tools. See *Using Siebel Tools*.

Table 29 lists examples of previously used tables which are inactive in Release 7 (you can no longer use these tables) and lists the suggested new tables to which custom extensions need to be reapplied. The new tables are recommendations only; the tables that you need to apply the extensions to might vary depending on their type and use. Contact Siebel Technical Services to validate the reapplication of extension columns and to review the steps necessary to migrate any extension column data to the new tables.

This data must be migrated during both the development and production environment upgrades.

Table 29. Examples of 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

If you have created many custom extension columns on the tables S_EMPLOYEE or S_ORG_INT, both of which are no longer used in Release 7, 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 need to 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. You should 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 buscomp migration utility, you will find a list of fields that require your attention.

Table 30 lists examples of previously used tables that are no longer used in Release 7, but that you may want to continue to use.

Table 30. Examples of Tables That Are Not Used in Release 7

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 31 lists examples of tables which were unused in previous releases of Siebel eBusiness applications, but are now used in Release 7.

Table 31. Examples of Previously Unused Tables That Are Used in Release 7

Now Used Table	Used to be...
S_ACT_EMP	S_EVT_ACT
S_ACT_CON	S_EVT_ACT

Resolving Business Component and Join Conflicts

Upgrades: All upgrades.

Environments: Development environment only.

After the Upgrade Siebel Database Schema phase, you may need to perform manual steps to business component fields and joins, depending upon the complexity of your business component configuration. It is strongly recommended that you thoroughly review the post upgrade configuration to make sure that the object level definitions are preserved as expected. During the upgrade, a list of business component joins and fields that need to be manually rectified are generated in a log file titled, upgcust. log. This particular log file, along with others generated by the upgrade process, can be found under *SIEBEL_ROOT*/log.

The log file contains two distinct sections:

■ 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

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, you should manually configure the business component field to reference that database column. Any manual reconfiguration must be done in the New Customer Repository after the upgrade has been run.

■ Part 2:

Because the data from S_EMPLOYEE and S_CONTACT has moved into more than one table, there is potential for conflicts between custom configuration from a previous release and standard configuration in 7. You need to resolve these conflicts in order for the application to function as designed.

After the repository merge has been run, there may be inconsistencies in the join names and joins set at the field level. If you do not resolve these discrepancies, it is likely that the application configuration will result in errors or will 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 following properties:

- Business component name
- Join name

With this list, you will need to 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, any manual configuration must be done in the New Customer Repository after the upgrade has been run.

Deleting Duplicate EIM Mappings

Upgrades: Release 6.x only.

Environments: Development environment only.

Any custom Enterprise Integration Manager (EIM) mappings that were added to previous versions of Siebel eBusiness applications will be automatically migrated to the new custom repository. If custom mappings are duplicated by new mappings added as part of Siebel version 7, errors will result when you use EIM.

NOTE: Custom EIM mappings are only added to old _IF tables, not to new EIM tables. If you need to use the new EIM tables, you must manually add the customization to them.

The Database Server Configuration Wizard that you ran in “Upgrading the Physical Schema” on page 185 generated a file called mapclash.txt. This file lists any custom EIM mappings that conflict with Siebel Systems-provided mappings. The file contains the columns illustrated and defined in Table 32.

Table 32. Mapclash.txt File Column Names and Definitions

Column Name	Definition
IT_Name	Interface table name (example: S_ACCOUNT_IF)
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_IF, 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.

Exporting the New Custom Repository Export File

Upgrades: All upgrades.

Environments: Development environment only.

The Database Server Configuration Wizard exports the new customized repository to a file called `custrep.dat`, located in the `DBSRVR_PLTFRM_ROOT` directory on the Siebel Server from which you ran the upgrade. Locate this file and make a backup copy of it. You will need it for the upgrade of the production database.

If you make *any* changes to your custom repository after running the development upgrade Database Server Configuration Wizard, including reapplying custom extensions or modifying EIM mappings in the two previous steps, you must export a new copy of the repository before performing the production upgrade. Use the script `DBSRVR_PLTFRM_ROOT/master_exprep.ucf` to export your current repository to the file that you designated as the output file in the Database Server Configuration Wizard.

The `master_exprep.ucf` calls the `repi mexp` utility, which connects to two separate repositories—the repository whose structure is to be extracted, and the repository whose content is to be extracted. Apart from unusual circumstances, you will always extract *both* the structure and content of the same repository.

For information on exporting repositories, see the *Using Siebel Tools*.

NOTE: Before you export the repository, make sure that *all* Siebel Tools projects are unlocked to promote a stable environment.

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 Server. 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*.

Extracting Developers or Siebel Tools Clients

Upgrades: All upgrades.

Environments: Development environment only.

Extract all Siebel Tools developers and clients using the Database Extract component from a new Siebel Server.

After you have extracted and initialized the mobile databases of all Siebel Tools developers, they must check out a read-only copy of all projects into the local database. For further procedures on setting up Siebel Tools developers, see *Using Siebel Tools*.

Testing the Upgraded Configuration

You must test the upgraded configuration thoroughly to make sure that all functionality works as expected. You need to execute your full development or acceptance test plan, including at least one synchronization session from a mobile client, before proceeding to upgrade your production environment.

If you make further changes to your customized configuration, remember to re-export the repository to the `custrep.dat` file before running the production environment upgrade.

After you have thoroughly and successfully tested your upgraded configuration, your development environment upgrade is complete. Proceed with [Chapter 10, “Preparing a Production Environment for Upgrade.”](#)

NOTE: Save the log files generated during the repository upgrade and schema upgrade. Locate the `custrep.dat` file in the `DBSRVR_ROOT\COMMON` subdirectory. Save the `custrep.dat` file for your production upgrade.

Resetting Database Server Configuration Parameters

Upgrades: All upgrades.

Environments: Development environment only.

After you complete your upgrade, you need to reset your Siebel Database Server configuration to installation settings. See the *Siebel Installation Guide* for the operating system you are using.

Upgrading to the Symbolic String Model

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Development environment only.

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

Release 7.7 introduced a symbolic string model. It replaces the locale-based string model. In the locale-based string model, each UI text string in the Tools repository is part of a UI object definition, such as an applet. The translations of the text string are maintained as child objects in child locale records for each UI object. For commonly used text strings, this means there is a lot of redundancy between UI objects.

The new symbolic string model is object-oriented. A single symbolic string replaces the text string translations. For each language, a text string is defined and assigned to the symbolic string as an attribute. This simplifies multilingual management of text strings throughout the UI.

Some strings will not be converted to the symbolic string model during upgrade. Seed data, error messages, lists of values (LOVs), and non-translatable attributes (such as the text alignment property on a control) will continue to use locale-based strings.

You must execute a conversion utility (consoleapp) to convert and consolidate your custom locale-based strings to the new model. If you plan to install a language pack, it is recommended that you do so before you run the string conversion or consolidation process.

Procedures for converting or consolidating to the symbolic strings model are found in *Using Siebel Tools*.

19 Postupgrade Tasks for a Production Environment

This chapter contains the following topics:

- “Upgrading Siebel Mobile and Dedicated Web Clients” on page 337
- “Upgrading Regional Servers” on page 337

Upgrading Siebel Mobile and Dedicated Web Clients

Upgrades: All upgrades.

Environments: Production environment only.

After a successful upgrade of Siebel eBusiness applications, you are ready to upgrade your Siebel Mobile and Dedicated Web Clients.

To upgrade Siebel client software on Siebel Dedicated Web Clients and Siebel Mobile Client workstations

- 1 Uninstall the previous version of the Siebel client application.
- 2 Install the Release 7.7 version of the Siebel client application. You can use either of the following methods for the installation:
 - Use the Siebel Image Creator utility to uncompress JAR files and create a Siebel network image, or staging area, for your Siebel applications.
 - Use a custom installer created using the Siebel Packager utility. For more information about the Packager utility, see *Siebel Installation Guide* for the operating system you are using.

NOTE: See *Siebel Installation Guide* for the operating system you are using for detailed procedures about how to uninstall or install Siebel clients.

Upgrading Regional Servers

Upgrades: All upgrades.

Environments: Production environment only.

After the production upgrade has executed successfully and the system integrity has been adequately verified, database extracts must be run for all Regional Servers in the deployment. Before initialization of the Regional Nodes, you must drop all Siebel tables, views, and indexes using the appropriate documentation from the RDBMS vendor. See *Siebel Remote and Replication Manager Administration Guide* for more information on database extracts, and regional server initialization.

20 Postupgrade Tasks for Applications

This chapter contains the following topics:

- “Generating Reporting Relationships After Upgrade” on page 340
- “Setting Up Your Environment to Support Global Time Zone” on page 342
- “Updating Enterprise Application Integration (EAI) After Upgrade” on page 343
- “Setting Visibility Modes for Access Control” on page 344
- “Removing Call Center Duplicate Logins After Upgrade” on page 347

Siebel ERM

- “Upgrading ERM Customized Microsite and Group News Pages” on page 347
- “Migrating Course Duration Information for Siebel Training” on page 349
- “Upgrading the Launch Field in Siebel Training LOV” on page 349
- “Upgrading the Test Status in Siebel Training After Upgrade” on page 350
- “Verifying Class and Session Times in Siebel Training” on page 350
- “Upgrading ERM Approval Business Process Workflows” on page 352

Siebel Financial Services

- “Migrating Data to the Bankruptcy Status Field” on page 353
- “Upgrading File System Attachments for Siebel Financial Services Call Reports” on page 353
- “Upgrading File System for Household Notes and Attachments” on page 354

Siebel Marketing

- “Upgrading Marketing Responsibilities” on page 356
- “Reviewing Marketing Campaign Data” on page 356
- “Reviewing Renamed Fields in Siebel Marketing” on page 357
- “Displaying Marketing Regions” on page 357
- “Revising Marketing Program Flowchart Icons” on page 358
- “Setting Default Campaign Execution Options” on page 358
- “Upgrading Activity Plans for Programs and Campaigns” on page 358
- “Upgrading the Newsletter Offer Type” on page 359
- “Configuring Universal Inbox” on page 359

- “Reviewing Data Warehouse Schema Changes” on page 360

Order Management

- “Upgrading Siebel Purchase Orders” on page 360
- “Configuring Asset-Based Ordering” on page 361
- “Reviewing Address Records After Upgrade” on page 362
- “Configuring Products and Quotes” on page 363

Siebel Pricer

- “Upgrading Attribute Pricing” on page 366
- “Verifying the Upgrade to Aggregate Discounts in Pricer” on page 367

Financial Services

- “Migrating Data to the Bankruptcy Status Field” on page 368

Workflows

- “Upgrading Seeded Workflows” on page 368
- “Upgrading Inbound Workflows” on page 368

Product Configurator

- “Upgrading Siebel Product Configurator” on page 369

Generating Reporting Relationships After Upgrade

Upgrades: All upgrades.

Environments: All environments.

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

The Generate Reporting Relationships process needs to be executed after the upgrade to Release 7.x and whenever the denormalized hierarchy structure (S_PARTY_RPT_REL) becomes out of sync with the data in the normalized tables (S_PARTY). Tables can become out of sync in the following cases:

- After upgrading to Release 7.x, the organizational hierarchy (even if there is only one organization) must be established to maintain appropriate visibility in the views cited above.
- When EIM is used to import or update any of the hierarchies (positions, organizations, or access groups).

In Siebel 7.x, there are three visibility hierarchies—position, organization, and access groups. These hierarchies are denormalized and maintained in the table S_PARTY_RPT_REL. These denormalized hierarchies are necessary for executing visibility modes that go up or down a hierarchy. For example:

- **Manager view mode.** My Team's Accounts View displays all accounts on which managers and their subordinates are working.
- **Suborganizations view mode.** All Contacts across My Organizations View displays all contacts that are associated to either my organization or any of my organization's suborganizations.

The Generate Reporting Relationships process rebuilds the denormalized relationships in the S_PARTY_RPT_REL table so that the hierarchical view modes display the correct information. The basic operation of the function is to empty the S_PARTY_RPT_REL table and then walk through each S_PARTY record to re-create the denormalized hierarchical structures in the table. This process generates a large number of transactions for Siebel Remote users and regional nodes.

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

This operation is time and CPU/memory-intensive. The process may take several minutes, depending on the size and complexity of your organizational structures. Do not perform this when you are running other memory-intensive processes.

To expose the Generate Reporting Relationships button

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

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

To generate reporting relationships

- 1 If you have an active Siebel Remote environment, confer with a Siebel administrator. The administrator needs to arrange for the Transaction Processor to be paused before performing this procedure.
- 2 Choose Group Administration under Site Map and navigate to the Positions view in the Siebel Web client application. Click the Generate Reporting Relationships button in the Position List Applet NB. Note that generating the reporting relationship may cause a large number of Siebel Remote transactions to be generated.
- 3 When this has completed, restart the Transaction Processor.

Setting Up Your Environment to Support Global Time Zone

Upgrades: All upgrades.

Environments: All environments.

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

If you previously upgraded to Release 7.x and you enabled your environment for global time zone, you do not need to repeat the procedure. Instead, you need to upgrade data from columns that were *not* UTC-enabled in Release 7.x that are UTC-enabled in the current release.

Global deployments typically span multiple time zones. The global time zone feature converts and stores date and time data using the Universal Time Coordinated (UTC) standard. This feature enables you to track dates and times in a common format across multiple time zones.

Although enabling your environment for global time zone is optional in Release 7.x, it is strongly recommended that you operate your production environment with global time zone enabled.

See [“Upgrading UTC Delta Columns” on page 343](#) for information about upgrading 7.x data for columns that were *not* UTC-enabled in Release 7.x that changed to UTC-enabled in the current release.

Enabling Global Time Zone

To enable global time zone support, after your upgrade is complete, you need to set the global time zone parameter (Universal Time Coordinated system preference) to TRUE through Siebel Tools. The UTC system preference is not enabled after an upgrade to Release 7.x. To enable the global time zone feature, you must run the UTC conversion utility. The high-level steps you need to perform are outlined in the following procedure (for more information, see *Global Deployment Guide*).

To enable global time zone support after an upgrade

- 1 Stop Siebel Servers.
- 2 Set UTC system preference to FALSE before you convert your historical data.
- 3 Prepare your data for conversion to global time zone, as instructed in *Global Deployment Guide*.

- 4 Convert your historical data, using the UTC conversion utility, to make all existing date/time values consistent with global time zone logic. Perform this step as instructed in *Global Deployment Guide*.
- 5 After you have confirmed that your UTC conversion was successful, turn on global time zone by setting the UTC system preference to TRUE. To reset this parameter through Server Manager, navigate to Application Administration > System Preferences.
- 6 Bring up Siebel Servers and Web Server.

Upgrading UTC Delta Columns

If you enabled UTC time zone support in a previous upgrade, perform the following steps to upgrade the delta columns (those that are newly UTC-enabled in the current release).

For more information about UTC, see *Global Deployment Guide*.

To upgrade delta columns to UTC

- 1 Verify that you have a current backup of your database.
- 2 Open the master_utc.ucf. It is located in the following directory in the Siebel Database Server installation:
 - Windows: `DBSRVR_ROOT\DATABASE_PLATFORM\master_utc.ucf`
 - UNIX: `DBSRVR_ROOT/DATABASE_PLATFORM/master_utc.ucf`
- 3 Edit the File Name parameter by replacing `driver_utc.ucf` with `driver_utc_delta.ucf`.
- 4 Launch the appropriate Database Server Configuration utility:
 - UNIX: `dbsrvr_config.ksh`
 - Windows: Start > Programs > Siebel Enterprise Servers > Configure DB Server

You can also start the wizard under Windows by typing `ssinconfw -l language_code -v Y` at a Command prompt. The *language_code* is the three-letter language code, in all capitals, for the language in which you want the GUI to display—for example ENU for English.

If you are prompted to select an .scm file, select `dbsrvr.scm`.

- 5 When you reach the Database Server Options prompt, select Run Database Utilities.
- 6 At the Database Utility Selection prompt, select Universal Time Code Conversion.
- 7 Continue until you are prompted to run the Siebel Upgrade Wizard, then click OK.

Updating Enterprise Application Integration (EAI) After Upgrade

Upgrades: All upgrades.

Environments: Development environment only.

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

If you use Enterprise Application Integration (EAI), perform the following procedure to update the definitions of the business objects to account for changes in data type, length, edit format or other properties during upgrading to a new version of Siebel eBusiness applications.

To upgrade integration objects

- 1 Determine whether you need to synchronize the integration objects, and synchronize if necessary.

To determine whether you need to synchronize integration objects, review the synchronization considerations in *Integration Platform Technologies: Siebel Enterprise Application Integration*.

- 2 Validate the integration objects.
- 3 If you receive validation errors, inactivate the user keys or fields that cause the error.
- 4 If you receive the error *List Of* in the XML Parent Element, manually remove the value *List Of* from the XML Parent Element.

Setting Visibility Modes for Access Control

Upgrades: Release 6.x only.

Environments: Development environment only.

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

Certain areas of the Release 7.x 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.

For analysis and recommendations for choosing and implementing access control strategies from an upgrade perspective, refer to Siebel SupportWeb for the *Access Control Upgrade and Migration Guide* technical note.

Removing Call Center Duplicate Logins After Upgrade

Upgrades: Release 6.x only.

Environments: Production environment only.

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

During the upgrade to Release 7.7.x, 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 ERM Customized Microsite and Group News Pages

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

As of Release 7.7, virtual business components are provided for microsite and Group News pages. These new business components include important rendering optimizations. If you have created customized page sections, you must change the underlying business component.

These old business components are no longer supported:

- ePortal MM Page Item 1
- ePortal MM Page Item 2
- ePortal MM Page Item 3
- ePortal MM Page Item 7
- ePortal MM Page Item 8

They are replaced by the following virtual business components:

- ERM Microsite Section Body VBC
- ERM Microsite Section Navigation Bar VBC
- ERM Section Page Footer VBC
- ERM Section Page Title VBC
- ERM Microsite Section Quick Picks VBC

New Virtual Business Components

In addition, two virtual business components are provided for creating new page sections:

- For microsite pages: ERM Microsite Section VBC (Copy to create a new Section)
- For Group News pages: ERM Group NewsVBC (Copy to create a new Section)

To upgrade customized microsite or Group News pages

- 1** In Siebel Tools, locate the applet for the microsite or Group News page that you want to upgrade. Write down the section code from the applet's search specification.
- 2** Locate the appropriate virtual business component:
 - For microsite pages: ERM Microsite Section VBC (Copy to create a new Section)
 - For Group News pages: ERM Group NewsVBC (Copy to create a new Section)
- 3** Copy this virtual business component and give the copy a new name.
- 4** Activate the new virtual business component.
- 5** In the new virtual business component's user properties, activate MicrositeSection. Set it equal to the section code from the applet.
- 6** Find the applet again and change its business component to the new virtual business component.
- 7** Delete the applet's search specification.
- 8** Change the applet's class to CSSSWEFrameListERMPageRender.

- 9 Add the new virtual business component to the business object of the views to which the applet belongs.
- 10 Recompile and deploy.

Additional information on upgrading microsite pages and Group News pages can be found in Technical Notes on Siebel SupportWeb.

Migrating Course Duration Information for Siebel Training

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

When upgrading from a release prior to Release 7.7, course duration information for Siebel Training is not migrated. To display course duration information you must move course duration to an extension column.

To migrate course duration information

- 1 In Siebel Tools, add an extension column to S_PROD_INT_CRSE. Assign the column the same length and data type as S_SRC_EVT.DURATION_DESC.
- 2 Use an SQL command to transfer existing course duration data to the extension column.
The SQL command should have the following form:

```
update S_PROD_INT_CRSE a
set a.<name of new extension column> = (select b.DURATION_DESC from S_SRC_EVT b)
where a.par_row = b.row_ID)
```
- 3 Expose the extension column in the user interface.
- 4 (Optional.) Create corresponding information in the S_PROD_INT_CRSE.CRSE_HOUR_NUM field.

Upgrading the Launch Field in Siebel Training LOV

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

In Siebel Training, the Distribution Method field is renamed Launch. This field is located in Administration - Training > Training Library.

The list of values for this field was revised in Release 7.7. The new values include *Download*, *Launch New Browser*, and *Launch In-Line*.

In previous releases, this LOV contained additional values. If existing records contain these values, you must add these values to the LOV, or the records will not display.

Upgrading the Test Status in Siebel Training After Upgrade

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

As of Release 7.7, completed tests can have one of two statuses:

- *Completed-Passed*
- *Completed-Failed*

In previous releases, completed tests had only one status: *Completed*.

If you want to revise existing data to reflect the new completion statuses, you can write a script to revise the records. The script should use the following logic:

- If at least one test has reached the maximum attempts and not been passed, mark the status as *Completed-Failed*.
- If all tests have been passed at least once, mark the status as *Completed-Passed*. Not all test attempts have to be cleared. If there were three attempts allowed and the student failed on the first two but passed on the third attempt, this is acceptable.
- If at least one test has not been attempted, then there should be no change in the status of the registration. So, if a course had a Course Survey associated to it, and if the user had not attempted it yet, its status cannot go to *Completed-Passed*.
- Else *CompletedFailed* takes precedence over *Completed-Passed*.

Verifying Class and Session Times in Siebel Training

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

Environments: Production environment only.

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

At Release 7.7, the way class and session start times and end times are displayed is changed. For existing classes and sessions you must do the following:

- For classes, verify that the times displayed in the Start Date and End Date fields are correct.

- For sessions, verify that the times displayed in the Start and End fields are correct.

To verify class times are correct

- 1 Navigate to Administration-Training > Class Details.
- 2 In the Classes applet, expose the following hidden fields:
 - Start Time
 - End TimeAlternative: These fields are displayed in the form applet.
- 3 For each class record, compare the time in the Start Time field with the time in the Start Date field.
- 4 Right-click in the Start Date field to adjust the time to be the same as that in the Start Time field as required.
- 5 Compare the time in the End Time field with the time in the End Date field.
- 6 Right-click in the End Date field to adjust the time to be the same as that in the End Time field as required.
- 7 Repeat this procedure for all the displayed classes.
- 8 Hide the Start Time and End Time fields.

To verify session times

- 1 In the sessions applet, expose the following hidden fields:
 - Start Time
 - End Time
- 2 Select a class in the Classes applet.
- 3 For each session in the Sessions applet, compare the time in the Start Time field with the time in the Start field.
- 4 Right-click in the Start field to adjust the time to be the same as that in the Start Time field as required.
- 5 Compare the time in the End Time field with the time in the End field.
- 6 Right-click in the End field to adjust the time to be the same as that in the End Time field as required.
- 7 Select the next class in the Classes applet and repeat this procedure.
- 8 When you have examined all the sessions for all the classes, hide the Start Time and End Time fields.

Upgrading ERM Approval Business Process Workflows

Upgrades: Release 7.5.x only.

Environments: Production environment only.

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

In Release 7.5.3, a number of approval business processes were included in the Sample Database. These were intended for use with Employee Self-Service. If you have activated these workflows, you must manually upgrade the approval steps in them.

Also, if you have designed approval business processes for use with Employee Self-Service, you must manually upgrade the approval steps to convert them to business service operations.

The manual upgrade converts the approval steps from a standard workflow operation to a business process operation. This enables the workflow to use new Universal Inbox business service functionality.

Both Siebel Marketing and Siebel ERM use Universal Inbox. Application administrators can see all approval tasks across all applications. To verify conformance to data access policies at your site, see [“Configuring Universal Inbox” on page 359.](#)

To manually upgrade approval business process workflows

- 1 In Siebel Tools Business Process Designer, locate the desired workflow.
- 2 Change the Business Object of the workflow to UInbox Item Task.
- 3 Retrieve the Priority from the workflow's Extract Events Fields step and save the priority in the process property.

The priority is used to set the Inbox Owner Priority.

- 4 Pass the Priority to the RouteInboxItem function:
 - Input Argument: Task.OwnerInfoTaskPriority
 - Type: Process Property
 - Property Name: Item Priority
- 5 In the workflow diagram, locate all steps that check the approver's status.
- 6 Change the step from a Siebel Operation to a business service:
 - Type: Business Service
 - Business Service Name: Universal Inbox
 - Business Service Method: GetInboxOwnerInfoEx
 - Leave the Business Component field blank
- 7 Locate the workflow's Deactivate Inbox Owner step.
- 8 Change the step from a Siebel Operation to a business service:

- Type: Business Service
 - Business Service Method: Universal Inbox
 - Business Service Method: DeactivateInboxOwner
 - Pass in `InboxItemId`, `InboxTypeName`, and `OwnerPartyId` to the method
- 9 Deploy and activate the revised workflow.

Migrating Data to the Bankruptcy Status Field

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

Environments: Production environment only.

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

At Release 7.7, a `BK_STATUS_CD` column is provided in the `S_BANKRUPTCY` table. This column stores Bankruptcy status information that is used by the Siebel Financial Services application.

If you have implemented a bankruptcy status field in the release you are upgrading from, use an SQL command to migrate the data from the custom extension column to `BK_STATUS_CD`.

Upgrading File System Attachments for Siebel Financial Services Call Reports

Upgrades: Release 6.x only.

Environments: Production environment only.

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

When you upgrade to Release 7.7.x, 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.x only.

Environments: Production environment only.

This topic is part of an upgrade process. See [Chapter 4, "How to Perform the 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 Enter the following command:

```
chg_file_sys.ksh -s SOURCE_TABLE -t TARGET_TABLE
-f "FILE_SYSTEM"
```

where:

SOURCE_TABLE = Name of the original table.

TARGET_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).

For example:

```
chg_file_sys.ksh -s S_ORGGRP_ATT -t S_CONTACT_ATT
-f "C:\siebfile"
```

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, xtndobtbl .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 [“Checking for Inactivated EIM Table Columns” on page 371](#).

Upgrading Marketing Responsibilities

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

Release 7.7 added many new views. In addition, many views from earlier releases are obsolete. You will need to replace any responsibilities you developed for previous releases. A new set of responsibilities are provided in seed data for all active views in Siebel Marketing.

To implement the new responsibilities, use one of the following approaches:

- Add any custom views you create to the seed data responsibilities for Siebel Marketing.
- Create a set of separate responsibilities for any custom views you create. Provide users with the seed data responsibilities as well as your custom responsibilities.

Reviewing Marketing Campaign Data

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

Some Siebel Marketing data is not automatically upgraded:

- Field values from the obsolete Campaign Occurrence header records (from S_SRC) are not merged to the surviving parent campaign. If you have important historical data that is stored on each campaign occurrence, export the list of obsolete campaign occurrence records from the database and determine which field values (if any) you need to apply to the parent campaigns.
- Exported Lists for campaign occurrences are not re-parented to campaign plans. They remain in the obsolete table S_DD_LST_DISTR.
- Campaign expense records (S_SRC_COST) for campaign occurrences are not migrated. This is to prevent campaign expenses from being double-counted after the campaign plan and any campaigns (occurrences) are merged.

During the upgrade, campaign plans and campaigns are merged. This may cause some data to appear double-counted due to re-parenting of similar objects to the same campaign:

- **Program and Campaign Activities.** If Activities are associated to the campaign plan as well as the campaign (occurrence), similar activities may appear twice in the upgraded campaign.

- **Campaign Contacts.** A contact (or prospect) may appear in the same campaign more than once if the campaign member was targeted in multiple occurrences of the same campaign plan. After upgrade, this campaign member will appear in more than one campaign load for the same campaign.

Reviewing Renamed Fields in Siebel Marketing

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

Some applet fields have been renamed from previous releases. If you have written any customized user documentation, you need to update references to these fields. Renamed fields are shown in [Table 33 on page 357](#).

Table 33. Renamed Fields in Siebel Marketing

Business Component Field Name	Business Component	Original Caption	New Caption
Budget	Program Container	Budget	Assigned Budget
Period	Campaign	Period	Execution Period
Response Type	Campaign	Response Type	Enabled Follow Up Action

Displaying Marketing Regions

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

After the upgrade, the Region field for Marketing Plans, Programs, and Campaigns does not display. This is because the Region field is associated with the Region hierarchy rather than the Region LOV field.

To display marketing regions

- 1 Create new Marketing Regions under Administration – Location to correlate to the previous LOV values for the Region field.

For more information, see the *Siebel Marketing Installation and Administration Guide*.

- 2 Rename the Region LOV field and add it to the applets in the user interface.

This provides backward compatibility.

Revising Marketing Program Flowchart Icons

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

After the upgrade, the icons in the Marketing Program Flowchart view display larger than normal.

To revise the size of Program Flowchart icons

- 1 Right-click in the Program Flowchart applet and deselect Snap to Grid.
- 2 Select the desired icon and reduce its size.

For more information on setting up Program Flowcharts, see the *Siebel Marketing Installation and Administration Guide*.

Setting Default Campaign Execution Options

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

This postupgrade task is not mandatory but may be required by local business processes.

Campaign Execution options are important for controlling the load behavior, launch behavior, assignment behavior, and collaboration options for campaigns. Each Campaign Execution option has an assigned default value. Review your marketing business process to confirm that each Campaign Execution Option is set correctly.

For a discussion of how to set default campaign execution options, see the *Siebel Marketing Installation and Administration Guide*.

Upgrading Activity Plans for Programs and Campaigns

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

This postupgrade task is not mandatory but may be required by local business processes.

As of Release 7.7, the Activity Template type DBM Campaign is obsolete and is not used. Change the template type to Campaign.

To upgrade Activity Plans

- 1 Navigate to Administration--Data > Activity Templates.
- 2 Query for templates of type DBM Campaign.
- 3 For each template, update the type to Campaign.

Upgrading the Newsletter Offer Type

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

This postupgrade task is not mandatory but may be required by local business processes.

As of Release 7.7, the offer type eNewsletter is obsolete. If you have existing offers that are of type eNewsletter, revise the type to Email.

To upgrade the offer type

- 1 Navigate to Offers > All Offers.
- 2 Query for Channel = eNewsletter.
- 3 For each offer, change the offer type to Email.

Configuring Universal Inbox

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

As of Release 7.7, Universal Inbox is implemented as a business service. It provides management of approval queues in both Siebel Employee Relationship Management (ERM) and in Siebel Marketing. Application administrators can see all the queued tasks in Universal Inbox across all applications.

This means that the Siebel ERM administrator can see tasks queued in Siebel Marketing and vice versa. If this cross-application visibility violates data visibility policies at your site, create a copy of the Universal Inbox. Then assign the copy to the workflows in one of the applications. Use the preconfigured version of Universal Inbox in the other application. This creates physically separate approval queues.

Universal Inbox in Siebel Marketing

As of Release 7.7, Siebel Marketing uses a new business service, Universal Inbox. It provides centralized management of approval queues and is used by both Siebel Marketing and Siebel Employee Relationship Management (ERM). Application administrators can see all approval tasks across all applications.

Reviewing Data Warehouse Schema Changes

Upgrades: Release 6.x only.

Environments: Production environment only.

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

Customers who use Siebel Marketing and the Marketing Server must manually upgrade their metadata for those cases in which they have mapped to tables or fields in the Siebel main database or the Business Data Warehouse that have changed in Release 7.x. Due to the extensive changes to the Data Warehouse schema in Release 7.x, releases, entirely new mappings may be required. See *Siebel Marketing Installation and Administration Guide* for information on table and field metadata mappings.

Upgrading Siebel Purchase Orders

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

If upgrading from a release prior to Release 7.7, you must run a utility to update the Transaction Amount field in Payment Lines that have Purchase Order as the payment method. The utility requires the .srf that you compiled after upgrading your development environment.

The utility performs the following steps:

- 1 Creates a new Order Entry business object.
- 2 Creates an Order Entry business component and a Payments business component.
- 3 Checks all order records and looks at the Payment Method of corresponding Payment Lines.
- 4 If the Payment Method of a Payment Line is Purchase Order, it updates the Transaction Amount field to Order Total.

The utility requires the following parameters:

- *USERNAME* = Siebel user login name.
- *PASSWORD* = Siebel login password.
- *LANG* = Language used.

- `CFG_FILE` = The configuration file used to launch the application.
- `DATA_SOURCE` = Date source used from the .cfg file.

To update the Transaction Amount field in Payment Lines

- Enter the following command:

```
Pmntupgd /u USERNAME /p PASSWORD /l LANG /c CFG_FILE /d DATA_SOURCE
```

Configuring Asset-Based Ordering

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Production environment only.

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

Release 7.7 introduced several architectural changes to the asset-based ordering workflows and views:

- **Asset-Based Ordering workflows.** The asset-based ordering workflows have been modified. If you are upgrading from Release 7.0.4 or Release 7.5, and you have modified these workflows, you must reapply your changes to these workflows after the upgrade.
- **Quote > Orders view.** The Sales Order button now uses a named method to invoke the SIS OM Quote To Order Workflow--PMT Version. You can now pass arguments to the workflow. In Release 7.5, this button was hard-coded. If you have written scripts or made other changes regarding the Sales Order button, consider modifying them to take advantage of the named method.
- **Order Header applet.** The Submit button now uses a named method to invoke the SIS OM Submit Order Process. You can now pass arguments to the workflow. In Release 7.5, this button was hard-coded. If you have written scripts or made other changes regarding the Submit button, consider modifying them to take advantage of the named method.
- **Disconnect and Modify workflows.** The named methods used to invoke the Disconnect and Modify workflows now pass the additional properties shown in [Table 34](#). If you have written scripts or made other changes regarding these buttons or named methods, you must manually add these after the upgrade.

Table 34. Additional Properties Passed by Named Methods

Property	Explanation
""Due Date"", "Today() + 1"	Sets the Due Date on all line items.
""BC Context"", ""Asset""	Passed as a property to the workflow.
""Compound Product Number"", "[Compound Product Number]"	The Compound Product Number is used for Network Order Entry.

- **Delta-In-Place feature.** Delta-In-Place updates the action code of a line item when a specific line item field value is updated. For example, if a user updates the Service ID value of a product in the Order Line Item applet, the action code changes from - (dash) to Update.

In Release 7.5, the Delta-In-Place feature was implemented through User Properties on the Quote Item and Order Entry – Line Items business components.

For example:

SIS OM On Field Update Set 6

```
"Service Id", "Action Code", "If([Action Code] = LookupValue(' DELTA_ACTION_CODE' , ' Existing' ), LookupValue(' DELTA_ACTION_CODE' , ' Modified' ), [Action Code])"
```

In Release 7.7, the Delta-In-Place feature was moved to the SIS OM PMT Service workflow. The SIS OM PMT Delta method that is used for changes made to products within the Configurator session is now used to generate the action code for changes to line-item level fields.

If you have added new fields to the Order Management process (quote, order, asset) and then used the SIS OM Field Update Set user property, you must create new user properties in the SIS OM PMT Service to replace these.

For example, if you created a new custom field called Point of Presence, you can delete the old SIS OM On Field Update Set user properties and create a new one as follows:

- **Name:** Delta Line Item Compare Field 26
- **Value:** [Point Of Presence]:[Point of Presence]
- **Siebel Configurator runtime: Done button.** In Releases 7.0.4 and 7.5, when a user clicks Done in a Siebel Configurator session, Siebel EAI transfers user selections into Quote and Order line items. This process used Quotes, Orders, and Assets-related business components.

As of Release 7.7, this process uses new, smaller business components. This improves performance. The new business components begin with MACD. If you have modified the original business components, you must apply these changes to the MACD business components.

Reviewing Address Records After Upgrade

Upgrades:

- From Release 6.x Siebel Industrial applications (except Communications, Media, and Energy) that you are upgrading directly to Release 7.7 Siebel Industrial applications
- From Release 6.x and 7.0.x Siebel Financial Services applications that you are upgrading to Release 7.7 Siebel Industrial applications
- From Release 6.x and 7.7 Siebel eBusiness applications that you are upgrading to Release 7.7 Siebel Industrial applications

- From Release 6.2.1 of Siebel Financial Services applications on IBM z/OS platforms that you are upgrading to Release 7.7.x of Siebel Industrial applications on IBM z/OS platforms

NOTE: This topic does not apply to Siebel eBusiness applications (HOR) that you are upgrading to a later release of Siebel eBusiness applications (HOR).

Environments: Production environment only.

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

The way address data is stored changed at Release 7.7. The upgrade migrated data from the table S_ADDR_ORG to S_ADDR_PER. The upgrade uses the following method for preserving address data:

- If any records have the same value for ADDR_NAME within or across S_ADDR_ORG and S_ADDR_PER, the upgrade process preserves ADDR_NAME for one of the records and appends the ROW_ID to ADDR_NAME in the others.
- If a record in S_ADDR_ORG has the same value for ADDR_NAME as a record in S_ADDR_PER, the upgrade process appends the ROW_ID to ADDR_NAME in one of the records.
- The upgrade process truncates the value of ADDR_NAME to fit the size of the column.
- The upgrade generates a report listing the records with duplicate ADDR_NAME within and across S_ADDR_ORG and S_ADDR_PER.

After the upgrade completes, review this report and edit or delete records from S_ADDR_PER as desired.

NOTE: This report is also generated for 7.0.x Siebel Industry Solutions upgrades to 7.7 Siebel Industry applications. Ignore this report for these upgrades.

To review address records after upgrade

- 1 Review the report generated by the upgrade:
Windows: `SIEBEL_ROOT\log\rpt_dup_addr_names.txt`
UNIX: `$SIEBEL_ROOT/1og/rpt_dup_addr_names.txt`
- 2 Use the report to identify records that are duplicates.
- 3 Use the application or Enterprise Integration Manager (EIM) to delete or revise records as needed.

Configuring Products and Quotes

Upgrades: Release 6.x only.

Environments: Production environment only.

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

You may need to adjust or configure some customizable products and quotes after upgrade.

Customizable Products

After upgrading to Release 7.7.x, 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_SQ_ITEM_ID = rtrim(PAR_SQ_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.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

Flag	Parameter	Description	Required
	SCRIPT_LOCATION	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_unix.ksh

Upgrading Attribute Pricing

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

Environments: Production environment only.

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

At Release 7.7, the attribute adjustments feature replaces attribute pricing in Siebel Pricer. The attribute adjustments feature is part of the dynamic matrix pricing feature.

You must manually upgrade attribute pricing data to attribute adjustments by running a business service method. The business service method does the following:

- Upgrades attribute pricing headers to attribute adjustment headers
- Upgrades attribute pricing attributes to dynamic matrix dimensions
- Upgrades attribute pricing values to dynamic matrix dimension domains
- Upgrades attribute pricing adjustment items to dynamic matrix rules

To upgrade attribute pricing to attribute adjustments

- 1 Launch Siebel Sales.
- 2 Verify that all attribute classes have been upgraded to product classes.
- 3 From the application-level menu, choose Site Map > Administration - Business Service > Simulator.
- 4 Create a new record:

Field	Value
Service Name	PSP Pricer Upgrade
Method Name	UpgradeDynamicMatrix

- 5 Click Run to start the business method.

Verifying the Upgrade to Aggregate Discounts in Pricer

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

Environments: Production environment only.

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

At Release 7.7, the aggregate discounts feature replaces bundle factors in Siebel Pricer. The bundle factor definitions are upgraded to aggregate discounts, and the sequencing of bundle factors are upgraded to aggregate discount sequences.

The name of the aggregate discount in Pricer will be set to *bundle factor name + row ID of the record*. This is because Pricer requires the aggregate discount name to be unique.

Sequencing of bundle factors within a pricing model is upgraded to aggregate discount sequencing. The name of the aggregate discount sequence is set to the pricing model name that contained the bundle factors.

The Price List and Price List Item will be stamped with the appropriate aggregate discount sequence name. In prior releases, the pricing model was specified at the Price List or, for customized products, at the Price List Line Item level. This ensured execution of the bundle factors at runtime.

At Release 7.7, the execution of aggregate discounts at runtime requires the association of the aggregate discount sequences at the Price List or Price List Line Item level.

The upgrade process makes the following assumptions about Pricer implementations prior to Release 7.7:

- Flowcharts were used to chain up bundle factors in the pricing model
- The bundle factor with the lowest sequence is connected to the Aggregate Start step
- Each Aggregate Start sequence contains only bundle factors and does not contain aggregate factors
- The next factor in the flowchart (when True or False) always has a larger sequence number.

If your implementation does not meet all the above criteria, the upgrade process moves the definitions to the appropriate Pricer entities (such as aggregate discounts), but the sequences will not be correct.

In such cases, you must manually verify that aggregate discount sequences chain up the aggregate discounts as intended. Use the sequence of execution that existed prior to the upgrade.

To verify upgrade to aggregate discounts

- 1** Launch Siebel Sales.
- 2** Navigate to Administration – Pricing > Aggregate Discount Sequences.
- 3** For each aggregate discount sequence, drill down to the detail view.

- 4 Locate the aggregate discount that corresponds to the first preupgrade bundle factor. Verify that it has the lowest sequence number. If not, revise the numbers in the Sequence, Next Discount If Used, and Next Discount If Not Used columns.
- 5 Verify that the numbers in the Next Discount If Used and Next Discount If Not Used columns are greater than the number in the Sequence column. Also verify that they point to the expected aggregate discounts. If not, revise the numbers in all three columns as required.

Migrating Data to the Bankruptcy Status Field

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

Environments: Production environment only.

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

At Release 7.7, a BK_STATUS_CD column is provided in the S_BANKRUPTCY table. This column stores Bankruptcy status information that is used by the Siebel Financial Services application.

If you have implemented a bankruptcy status field in the release you are upgrading from, use an SQL command to migrate the data from the custom extension column to BK_STATUS_CD.

Upgrading Seeded Workflows

Upgrades: All upgrades.

Environments: All environments.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform the 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: Release 7.0.x.

Environments: All environments.

This topic is part of an upgrade process. See [Chapter 4, “How to Perform the 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.x to 7.7, manually change the name of the *EAI MQSeries Transport* business service to *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 non-existent.

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. A special utility, REPIMEXP, is provided to migrate the Siebel Tools Repository to production. Refer to the *Siebel Business Process Designer Administration 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 7.7 begin S_WF_ (Note the underscore after WF.) After the upgrade to Release 7.7, 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.

Upgrading Siebel Product Configurator

Upgrades: Release 6.x only.

Environments: All environments.

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

When the upgrade process is finished, you must examine your upgraded models, in the form of customizable products, and release the products for end user use.

21 Postupgrade Tasks for Database and File System

This chapter contains the following topics:

- “Checking for Inactivated EIM Table Columns” on page 371
- “Validating Dock Objects and Rule Definitions” on page 371
- “Updating the File System Directory” on page 372
- “Upgrading to the Symbolic String Model” on page 373
- “Updating File System Attachments” on page 374

Checking for Inactivated EIM Table Columns

Upgrades: Releases 7.0.x, 7.5.x.

Environments: All environments.

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

In Release 7.7, some columns in EIM tables were inactivated (Active = FALSE). This was done to prevent the tables from exceeding the 32-KB table size limitation. In Siebel Tools, query for inactive columns in these tables and verify that this does not affect application function.

Validating Dock Objects and Rule Definitions

Upgrades: All upgrades.

Environments: Production environment only.

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

Changes to visibility rules and dock objects require the assistance of Siebel Technical Support or Siebel Expert Services.

If you deploy Siebel eBusiness applications to mobile users with local databases, you can run the DICTUTL utility to verify that all dock objects and rule definitions are correct. Dock objects allow mobile users to synchronize their local databases with the Siebel Server. Rules determine which data users synchronize. For more information about dock objects and rules, see *Siebel Tools Online Help* and *Siebel Remote and Replication Manager Administration Guide*.

Related Topic

[“Preserving Dock Objects and Visibility Rules” on page 103](#)

To verify that all dock object and rule definitions are correct

- 1 Navigate to the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Type the following command using the parameters specified in [Table 35](#):

```
dictutl /C ODBC_DATASOURCE /U USERNAME /P PASSWORD /D TABLEOWNER /N
"REPOSITORY_NAME" /A y 2> logfile.log
```

Table 35. Command Line Flags for DICTUTL

Flag	Parameter	Description	Required
/C	ODBC_DATASOURCE	ODBC datasource name	Yes
/U	USERNAME	User name to log in to database	Yes
/P	PASSWORD	User password to log in to database	Yes
/D	TABLEOWNER	User name of tableowner	Yes
/N	"REPOSITORY_NAME"	Name of repository for dictionary (the parameter must be bounded within double quotes)	Yes
/A	y or n	Enter the y parameter to ignore the dictionary cache. Enter n if you do not want to ignore the dictionary cache.	Yes

Review the `LOGFILE.log` file.

Updating the File System Directory

Upgrades: Release 6.x only.

Environments: All environments.

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

The Release 7.x file system contains subdirectories. During the upgrade of your production environment, Siebel Anywhere looks for certain files in the file system subdirectories, but these files only exist in the root file system directory.

Therefore, you need to create the appropriate subdirectories in your previous installation directory structure and then copy the required files from the root file system directory to the new subdirectories.

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 eBusiness applications.
- 3 Verify that files have copied correctly to the \siebfile\att directory.

After this has been verified, clean up the file system.

Upgrading to the Symbolic String Model

Upgrades: Releases 7.0.x, 7.5.x.

Environments: Development environment only.

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

Release 7.7 introduced a symbolic string model. It replaces the locale-based string model. In the locale-based string model, each UI text string in the Tools repository is part of a UI object definition, such as an applet. The translations of the text string are maintained as child objects in child locale records for each UI object. For commonly used text strings, this means there is a lot of redundancy between UI objects.

The new symbolic string model is object-oriented. A single symbolic string replaces the text string translations. For each language, a text string is defined and assigned to the symbolic string as an attribute. This simplifies multilingual management of text strings throughout the UI.

Some strings will not be converted to the symbolic string model during upgrade. Seed data, error messages, lists of values (LOVs), and non-translatable attributes (such as the text alignment property on a control) will continue to use locale-based strings.

You must execute a conversion utility (consoleapp) to convert and consolidate your custom locale-based strings to the new model. If you plan to install a language pack, it is recommended that you do so before you run the string conversion or consolidation process.

Procedures for converting or consolidating to the symbolic strings model are found in *Configuring Siebel eBusiness Applications*.

Updating File System Attachments

Upgrades: Release 7.0.x.

Environments: All environments.

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

Perform both of the following procedures to upgrade file system attachments.

Upgrading the Siebel File System

Perform the following procedure to upgrade the Siebel File System.

To upgrade the file system

- 1 Navigate to the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Enter the following command:

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

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

where:

- `SOURCE_TABLE` = name of the original, inactive, table.
- `TARGET_TABLE` = name of the new table to which the original data was migrated.
- `"FILE_SYSTEM"` = name of the directory where the file system resides (entered inside quotation marks).

Windows example:

```
chg_file_sys.bat S_EMPLOYEE_ATT S_CONTACT_ATT
"C: \siebfile"
```

UNIX example:

```
chg_file_sys.ksh -s S_EMPLOYEE_ATT -t S_CONTACT_ATT
-f "C: /siebfile"
```

Review the renamed files carefully to verify that they can be accessed by Siebel eBusiness applications.

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

Migrating Files

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

To migrate `S_LIT` files to `S_CB_ASSET_VER`

- 1 Navigate to the following directory:

Windows: `SIEBEL_ROOT\bin`

UNIX: `$SIEBEL_ROOT/bin`

- 2 Enter the following command:

Windows: `file_upg_mm.bat ODBC_SOURCE USER_NAME PASSWORD TABLE_OWNER
"FILE_SYSTEM_LOCATION" "SIEBEL_ROOT" "DBSRVR_ROOT"`

UNIX: `file_upg_mm.ksh ODBC_SOURCE USER_NAME PASSWORD TABLE_OWNER
FILE_SYSTEM_LOCATION $SIEBEL_ROOT DBSRVR_ROOT`

where:

- `ODBC_SOURCE` = the ODBC source of the database.
- `USER_NAME` = the database user name.
- `PASSWORD` = the password for the database user name.
- `TABLE_OWNER` = the database tableowner (if your implementation is on iSeries, you can use the value provided for `User_Name`).
- `FILE_SYSTEM_LOCATION` = the directory where the file system resides.
- `SIEBEL_ROOT` = the directory where the Siebel Server is installed.
- `DBSRVR_ROOT` = the directory where Siebel Database Server files are installed.

Windows example:

```
file_upg_mm.bat SEBL sadmi n sadmi npw SIEBEL "C:\siebfile" "C:\sea7xx\siebsrvr"  
"C:\sea7xx\dbsrvr"
```

UNIX example:

```
file_upg_mm.ksh SEBL sadmi n sadmi npw SIEBEL C:\siebfile $SIEBEL_ROOT  
C:/sea7xx/dbsrvr
```

Note that the UNIX syntax does not use quotes around `FILE_SYSTEM_LOCATION`, and `DBSRVR_ROOT`.

- 3 Review the renamed files carefully to verify that they can be accessed by Siebel eBusiness applications.

22 Tuning the Upgrade Scripts

This chapter contains the following topics:

- [“About Tuning the Production Upgrade Scripts” on page 377](#)
- [“Deactivating SQL that Affects No Data” on page 377](#)
- [“Transferring UNIX Files for Use by Upgrade Tuner” on page 378](#)
- [“Discarding Changes Made by the Upgrade Tuner” on page 380](#)

About Tuning the Production Upgrade Scripts

Upgrades: All upgrades.

Environments: Production environment only.

The Siebel Upgrade Tuner analyzes and eliminates SQL statements that do not affect any data. It also parallelizes table creation, table rebuilds, and index creation.

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 and to eliminate unneeded SQL statements.

The Siebel Upgrade Tuner is installed with the Siebel Server on Microsoft Windows and is used through a graphical user interface. The Siebel Upgrade Tuner does not run on the UNIX platform. To tune the production upgrade scripts for UNIX, you must first copy files to a Windows platform.

You can tune your production upgrade scripts at any time after upgrading the Siebel Database schema in your development environment.

You can run the Siebel Upgrade Tuner multiple times.

Deactivating SQL that Affects No Data

Upgrades: All upgrades.

Environments: Production environment only.

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

You must analyze your customizations and the nature of application data before you deactivate SQL statements that appear to affect zero rows. Do not deactivate zero-row SQLs on an empty or low-volume database. Use an exact copy of your production database, instead.

The upgrade scripts may contain SQL statements that run against all tables whether or not the tables contain any data. Such SQL executions may be redundant for your production upgrade. You can use the Siebel Upgrade Tuner to analyze, edit, and deactivate SQL statements that do not affect any data, thereby slowing the upgrade process.

The Siebel Upgrade Tuner contains four screens that you access by clicking on tabs:

- **Process Information.** This screen displays information retrieved by the Upgrade Tuner from the summary.xml file.
- **Parallelize Table Creation.** This screen allows you to parallelize table creation.
- **Parallelize Index Creation.** This screen allows you to parallelize index creation.
- **Deactivate 0-Row SQLs.** This screen displays SQL statements that run against tables without any data. You can selectively review these SQL statements, and if they are unnecessary, you may deactivate them for your production upgrade.

For more information on the Upgrade Tuner, see [“About Tuning the Production Upgrade Scripts” on page 377](#).

Prerequisites: For UNIX platforms, you must first transfer files to the Windows platform. See [“Transferring UNIX Files for Use by Upgrade Tuner” on page 378](#).

To identify and deactivate SQL statements that affect zero rows

- 1 Launch the Upgrade Tuner.
- 2 Click on the Deactivate 0-Row SQLs tab in the Siebel Upgrade Tuner.
The Deactivate 0-Row SQLs screen appears.
The Deactivate 0-Row SQLs screen only provides information for native SQL (not for odbcsql).
- 3 Inactivate an unnecessary zero-row SQL statement by clicking the check box for that statement in the Inactive column.
 - A check mark in the Inactive column indicates that the associated SQL statement is inactive and therefore not upgraded. The word *changed* appears beside the check box to indicate that the status changed.

Before deactivating a particular SQL statement, review it by scrolling through the comment field at the bottom of the screen, and then evaluate the safety of deactivating this statement. Make sure you are not deactivating SQL that is necessary for your upgrade.
 - To return all deactivated SQL statements to an active upgrade state (such as statements deactivated while tuning a previous upgrade), click Activate All.

Transferring UNIX Files for Use by Upgrade Tuner

Upgrades: All upgrades.

Environments: Production environment only.

Platforms: UNIX only.

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

To tune a production upgrade under UNIX, you must use file transfer protocol (FTP) to transfer your upgrade files to a Windows machine. Tune your production upgrade scripts in the Windows environment, then transfer your tuned upgrade files back to your UNIX machine.

Scripts generated during production upgrades simplify the file transfer process:

- **upgtuner_ftp_get.txt.** This script moves the upgrade files from a UNIX machine to a Windows temporary directory. (Development upgrades do not produce upgtuner_ftp_get.txt.)
- **upgtuner_ftp_put.txt.** This script moves the upgrade files from the Windows temporary directory back to the UNIX machine. (Development upgrades do not produce upgtuner_ftp_put.txt.)

For more information on the Upgrade Tuner, see [“About Tuning the Production Upgrade Scripts” on page 377.](#)

To transfer UNIX files for use by Upgrade Tuner

- 1 In the FTP scripts, replace placeholder parameters with actual values:

Placeholder	Value
&HostIP	IP address of UNIX machine
&Username	Username used to open an FTP session with the UNIX machine (for example sadmin)
&WindowsTempDir	Location of the temporary directory on the Windows machine (for example c:\temp\tuner)

- 2 On a Windows machine with the Siebel Upgrade Tuner installed, create a temporary directory for your tuning process (for example c:\temp\tuner).
- 3 Use file transfer protocol (FTP) or upgtuner_ftp_get.txt to move the following files from your UNIX machine to a temporary directory on the Windows machine:

File	Source Location on the UNIX machine
master_PROCESS.ucf	\$SIEBEL_ROOT/bin/
summary.xml	\$SIEBEL_ROOT/log/PROCESS/
driver_PROCESS.ucf	DBSRVR_ROOT/DBPLATFORM/upgrade/VERSION/
schema*.ddl	DBSRVR_ROOT/DBPLATFORM/
*.sql	DBSRVR_ROOT/DBPLATFORM/upgrade/VERSION/

- 4 After you have tuned the upgrade files, use file transfer protocol (FTP) or `upgtuner_ftp_put.txt` to return the files to the UNIX machine. Verify that the files are placed in the source locations specified in the table above.
- 5 Review the log file to verify that your file transfer was successful.

Discarding Changes Made by the Upgrade Tuner

Upgrades: All upgrades.

Environments: Production environment only.

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

If you make an error using the Upgrade Tuner, you can discard the changes to the upgrade scripts.

When you run the Upgrade Tuner, it automatically creates a backup copy of your original master, driver, DDL and SQL files, and appends the name of the original backup files with `.orig`.

This preserves the original backup files in case you need to roll back to the standard (pretuned) versions. If you run the Siebel Upgrade Tuner multiple times, the names of your most recent backup files are appended with `.old`.

For more information on the Upgrade Tuner, see [“About Tuning the Production Upgrade Scripts” on page 377.](#)

To discard changes made by Upgrade Tuner

- 1 Launch the Upgrade Tuner.
- 2 Locate and make a list of the files used as input by Upgrade Tuner.
These include the master file, driver file, DDL, and SQL files. These files will have `.orig` or `.old` appended.
- 3 Using the list, find and archive the files Upgrade Tuner revised. These include the master file, driver file, DDL, and SQL files.
- 4 Using the list, remove the `.orig` or `.old` from the Upgrade Tuner input files.

A

Schema Changes

This appendix includes the following topic:

- [“Important Schema Changes at Release 7.7” on page 381](#)

Important Schema Changes at Release 7.7

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

In Release 7.7.x, the upgrade includes the following important schema changes.

Customer and Order Management Applications

The following schema changes affect customer and order management applications. This includes Product Administration, Order Management, Quotes, Pricer, and Product Configurator.

New Party Payment Profile Table

The new Party Payment Profile table replaces the old Contact Payment Profile table. The new schema supports account and contact payment profiles. The upgrade migrates the data from the old Contact Payment Profile table to the new table.

Quote Teams

Quotes now support multiple sales representatives, or other positions, associated to a single quote. The existing reference to the sales representative for the quote is migrated to the new intersection table and reused as a primary child column.

Multiple Price Types

Products now support multiple price types (Siebel Business Applications only). The new Price Type Code column for existing products is set to One-Time. The new One-Time Charges Sub-Total column in Quote, Order, and Agreement line items is calculated and stored. The new Recurring Charges Sub-Total column in Quote, Order, and Agreement line items is set to 0.

Revision to the Line Item Discount Amount Field

The upgrade process changes how line item discounts are stored. If the data source for a discount amount is Pricer, the upgrade moves the value stored in the Discount Amount Field to the Pricing Adjustment Field.

If the Keep Discount flag is checked, the upgrade process does not make this change. Instead, all items that have this flag checked are treated as manual discounts.

The upgrade process determines the correct Header Discount Amount when both of the following are true:

- Before the upgrade, the line item did not have a Manual Discount Amount, Manual Discount %, or Manual Price Override specified
- The line item has a Header Discount % specified

Attribute Pricing Matrices in Pricer

Attribute pricing matrices in Siebel Pricer are replaced by attribute adjustments. After the database upgrade is complete, users must run a business service method to convert attribute pricing data to attribute adjustments. Obsolete tables and corresponding new tables are shown in [Table 36](#).

Table 36. Attribute Adjustment Tables

Obsolete Table	New Table
S_PRI_MTRX	S_DYN_MTRX
S_PRI_MTRX_ATTR	S_DYN_MTRX_DIM
S_PRI_MTRX_ITEM	S_DYN_MTRX_RL
S_PRI_MTRX_VAL	S_DYN_MTRX_DOM

Net Price, Currency Code, Exchange Date

The calculated Net Price field in Quote, Order, and Agreement line items is replaced by a new column. The upgrade calculates the value based on the basic configuration in the prior release. Quote, Order, and Agreement line items now support currency code and exchange date. The relevant data is migrated from the associated header record.

Effective Dates on Price List Line Items and CP Adjustments

Price List line items and Price List CP adjustments now support effective dates in the user key. The effective start and end date are copied from the associated header record where applicable. If this data is not available, the start date is set to 01-01-1980 and the end date is left as NULL.

Volume Discounts

Volume discounts now support discount amounts as well as the existing method of discount percentage. The new price adjustment type code column is used to specify the type. The existing discount amount column stores the associated value. Both columns are updated to support the existing data.

Bundle Discounts

Bundle factors in pricing models are converted to bundle sequences and discounts in Release 7.7. Obsolete tables and corresponding new tables are shown in [Table 37 on page 383](#).

References to pricing models in price lists and price list items are converted to refer to bundle sequences. All other pricing models must be manually reimplemented as PPS Procedures after the database upgrade.

Table 37. Bundle Discount Tables

Obsolete Tables	New Tables
S_PRI MDL	S_BUNDLE_SEQ
S_PRI MDL_FCTR	<ul style="list-style-type: none"> ■ S_BDL_SEQ_I TEM ■ S_BUNDLE_DI SCNT
S_PRI FCTR_I TM	S_BDL_DI SC_I TEM
S_PRI MDLFCTRVAL	NA
S_PRI MDL_OBJ	NA
S_PRI MDL_OBJCRT	NA
S_PRI FCTI TM_ATR	NA

Product Configurator

Release 7.7 introduces a new infrastructure for Product Configurator. The primary new table for configurator is S_VOD. This table stores the header information for products, classes, and attributes.

Other important changes are as follows:

- S_VOD_VER replaces S_PROD_CFGVER and stores the version information for product, class and attribute objects.
- S_I SS_OBJ_DEF stores the product and class definitions.
- S_I SS_ATTR_DEF stores the global (abstract) attribute definitions.
- S_I SS_ATTR_VAL replaces the concept of S_XA_ATTR. VLDTN_LOV_TYPE_CD and stores the enumerated values for the global attribute.
- S_I SS_OBJ_ATTR stores the relationship between local attributes and both classes and products. This replaces the relationships previously stored in S_XA_ATTR. CLASS_ID and S_PROD_I NT_XA.

Obsolete tables and corresponding new tables are shown in [Table 38](#).

Table 38. Product Configurator Tables

Obsolete Tables	New Tables
S_XA_ATTR	<ul style="list-style-type: none"> ■ S_VOD ■ S_I SS_ATTR_DEF ■ S_I SS_ATTR_VAL ■ S_I SS_OBJ_ATTR
S_XA_ATTR_LANG	S_I SS_OBJ_ATTR_LANG
S_XA_CLASS	S_VOD, S_I SS_OBJ_DEF
S_XA_CLASS_LANG	S_VOD_LANG
S_PROD_CFGVER	<ul style="list-style-type: none"> ■ S_VOD_VER ■ S_I SS_UI_OPTION
S_PROD_INT_XA	S_I SS_OBJ_ATTR
S_PROD_ITEM	S_I SS_SUB_OBJ
S_PROD_ITEM_LANG	S_I SS_SOBJ_LANG
S_CFG_VAR_DEF	S_I SS_OBJ_ITEM
S_CFG_PROP_DEF	S_I SS_OBJ_RSRC
S_CFG_SCRIPT_DEF	S_I SS_OBJ_SCRIPT
S_CFG_RULE_DEF	S_I SS_OBJ_CFGRL
S_CFGRLDEF_LANG	S_I SS_CFRL_LANG
S_CFG_RULENODE	S_I SS_CFRL_NODE
S_CFG_UIGROUP	S_I SS_UIOPT_GRP
S_CFGUIGRP_LANG	S_I SS_UIGR_LANG
S_CFG_UIGRP_ITEM	S_I SS_UIGR_ITEM
S_PROD_CFG_PROP	S_I SS_OBJUI_PROP
S_PRDCFGPR_LANG	S_I SS_UIPR_LANG

Like the obsolete tables, all of the new tables have versions. The upgrade migrates data from the obsolete tables to the new ones and creates new records for objects that were previously not versioned (particular classes, attributes, and products).

During the upgrade, additional data migration occurs as follows:

- S_PROD_INT to S_VOD and S_I SS_OBJ_DEF
- S_PROD_INT_LANG to S_VOD_LANG

S_PROD_INT and S_PROD_INT_LANG continue to be used in Release 7.7. Foreign key references to products still point to S_PROD_INT. The S_PROD_INT.CFG_MODEL_ID column is reused to refer to the associated S_VOD record in the Product Configurator infrastructure.

All the S_*_XA tables, for example S_QUOTE_ITEM_XA, now use the ATTR_NAME column to refer to the attribute associated to the object, for example Quote Item.

ATTR_ID is no longer used. In previous releases, it was used as a reference to S_XA_ATTR, which is obsolete in 7.7.

Captive Finance

Before Release 7.7, a contact was associated with only one bankruptcy. In Release 7.7, the relationship between contact and bankruptcy has been changed to M:M. The model also allows capturing bankruptcies for companies. The upgrade migrates data in two steps:

- Migrates bankruptcy information from S_CONTACT_FNX to S_BANKRUPTCY and S_BK_PARTY.
- Migrates associated attorney information for bankruptcies from S_PARTY_REL to S_BK_PARTY. In S_BK_PARTY, attorney type is Bank Attorney, Trustee Attorney, Debtor Attorney or Other Attorney.

The tables S_CONTACT_FNX and S_PARTY_REL continue to be used for other purposes.

eTraining

The upgrade makes the following schema changes:

- Release 7.7 introduces an automatic wait-list feature. Since existing wait-list records were created manually, the upgrade updates the S_SRC_EVT.AUTO_WAITLIST_FLG to N for these records.
- Moves Max Waitlist Num from S_PROD_INT_CRSE to S_SRC_EVT. This moves wait-list support from the course level to the class level.
- Moves the Allow Waitlist Flag from S_PROD_INT_CRSE to S_SRC_EVT. This moves wait-list support from the course level to the class level.

For existing class records, Max Waitlist Num and Allow Waitlist Flag are set to the values that were present for the corresponding course.

Field Service

Quote, Order, and Agreement line items now support multiple covered assets associated to a single line item. The existing reference to the covered asset for the line item is migrated to an intersection table and reused as a primary child column.

The upgrade creates the following new intersection tables:

- S_AGREE_ITEM_REL from S_AGREE_ITEM
- S_ORDER_ITEM_REL from S_ORDER_ITEM
- S_QUOTE_ITEM_REL from S_QUOTE_ITEM

The S_AGREE_ITEM, S_ORDER_ITEM and S_QUOTE_ITEM tables continue to be used.

Consumer Goods Advanced Planning

Release 7.7 introduces an Account Promotion Category in promotion planning. The new account promotion hierarchy is as follows:

- Account Plan
- Account Promotion
- Account Promotion Category
- Account Promotion Product
- Account Promotion Product Baseline/Shipment

To support this, the upgrade inserts account promotion records in S_SRC. The identifier for each record is SUB_TYPE = PLAN_ACCT_PROMOTION_CATEGORY.

The upgrade makes the following changes:

- For every Account Promotion record in S_SRC, the upgrade inserts a record for Account Promotion Category. The parent of the new record is the Account Promotion record.
- Populates S_SRC_CHNL, an extension table for S_SRC, for Account Promotion Category. S_SRC_CHNL.PAR_ROW_ID points to the new record created in S_SRC. S_SRC_CHNL stores certain attributes for Promotion, Promoted Products, and so on.
- Reparents Account Promotion Product records by pointing S_SRC.PAR_SRC_ID to Account Promotion Category.
- Revises Deals records stored in S_MDF_ALLOC that point to Account Promotion records in S_SRC to point also to Account Promotion Category records.

B

Siebel Marketing Upgrade Reference

This appendix contains the following topics:

- [“Obsolete Siebel Marketing Data” on page 387](#)
- [“Obsolete Business Objects in Siebel Marketing” on page 393](#)
- [“Obsolete or Replaced Views in Siebel Marketing” on page 394](#)

Obsolete Siebel Marketing Data

Upgrades: Releases 7.0.x, 7.5.x.

[Table 39 on page 387](#) lists marketing data that is obsolete as of Release 7.7. In the table, the Comments field explains how the data is handled.

Table 39. Obsolete Siebel Marketing Data as of Release 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
Programs (program occurrences)	Obsolete	None	None	Activities migrated to parent program.
Stages (stage occurrences)	Obsolete			

Table 39. Obsolete Siebel Marketing Data as of Release 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
Campaigns (campaign occurrences)	Obsolete	None	DBM Campaign Campaign Occurrences Campaign Occurrences Delete Campaign Results Campaign Segment Allocation Campaign Template Related Event Templates Campaign Wave List Distribution DBM Campaign Cost DBM Campaign Cost (Fixed) DBM Campaign Cost (Inbound) DBM Campaign Cost (Outbound) DBM Campaign Occurrence Cost DBM Campaign Occurrence Lists Campaign Occurrence Offer DBM Preview List	A campaign load record is created for each occurrence.

Table 39. Obsolete Siebel Marketing Data as of Release 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
eNewsletter Offers	Obsolete		Enewsletter Offer Enewsletter Offer Attachments Enewsletter Offer Attachments.Sequence Number (Sequence) Enewsletter Offer Comm Profile Parameter Enewsletter Offer Related URLs Enewsletter Offer Template Related Web Offers f/ Enewsletter Offer Related Web Surveys f/ Enewsletter Offer	Newsletter functionality is supported by conditional content formatting in email offers.
Segments	Usage history in programs is preserved. Segment criteria are obsolete.	S_CALL_LST_CRIT S_CALL_LST_DTL S_CALL_LST_QRY	Segment Detail-DD Segment Expression Campaign Segment Allocation Segment Campaign Allocation	Segments should be reconstructed in the Segment Designer.
Filters	Obsolete	S_DD_FILTER S_DD_FILTER_DTL	Filters Filters Detail Filters Expression Save Attribute Level Look Up	Filters should be applied using criteria in the Segment Designer or Segment Tree Designer or metadata constraints in the Siebel Analytics repository.

Table 39. Obsolete Siebel Marketing Data as of Release 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
List Formats	Obsolete	S_DD_LST_FMT S_DD_LST_FMTDTL S_DD_PROGLSTFMT	List Columns List Columns.Sequence (Sequence) List Formats Program List Formats Dependent List Format	Create List Formats using the List Format Designer.
Source Code Formats	Obsolete	None	None	Old source code formats are obsolete. After upgrade, source code formats need to be re-created with new elements.
Tables Joins	Obsolete	S_DD_DATA_OBJ S_DD_UNION_MBR S_DD_JOIN S_DD_JOIN_SPEC	DD Field DD Join DD Table Fields One Level Join Fields Join Fields.Sequence (Sequence) Union Tables	All data source metadata is managed in the Siebel Analytics repository.

Table 39. Obsolete Siebel Marketing Data as of Release 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
List Measures Bound Measures Custom Measures	Obsolete	S_DD_MEAS S_DD_MEASRSTRCT S_DD_SUBMEAS	Aggregation Function Available Measures Base Measure Bound Measures Custom Measures Dependent List Format Functions (Logical) Functions (Mathematical) List Measures Measures Parent Measures RowNum Measure SubMeasures Measure Aggregation (Tree) Members.Sequence (Sequence) Measure Restrict Measure Restrict (Tree)	Replaced by formulas in the Siebel Analytics repository.
Buckets (Measure Base Attributes)	Obsolete	S_DD_MEASATRP S_DD_MEAS_ATTR	Measure Attribute Members Measure Attribute Members (Tree) Measure Attribute Measure Attributes Measure Attributes (Tree)	Replaced by formulas in the Siebel Analytics repository.

Table 39. Obsolete Siebel Marketing Data as of Release 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
Attributes and Attribute Families	Obsolete	S_DD_ATTRFAMVMB S_DD_ATTRFAM S_DD_ATTRFAMLVL S_DD_ATTRFAMVAL S_DD_FIELD S_DD_HIERATR S_DD_HIERATRKEY S_DD_HIERATRLVL S_DD_HIERATRVAL	Attribute Families Attribute Family Resync VBC Attribute Family Value Attribute Filters Detail Attribute Levels Attribute Levels.Sequence (Sequence) Hierarchical Attribute Value Hierarchical Attributes Hierarchy (Fields) Hierarchy (Tree) Hierarchy Levels (Tree) Hierarchy Levels Search (Tree) Hierarchy Search (Fields) Hierarchy Search (Tree) Hierarchy Values (Fields) Hierarchy Values (Tree) Hierarchy.Sequence (Sequence)	Segment criteria are now created using presentation columns (fields) in the Segment Designer.

Table 39. Obsolete Siebel Marketing Data as of Release 7.7

Data	Effect of Upgrade	Obsolete Tables	Obsolete Business Components	Comments
Customer Hierarchies	Obsolete	S_DD_STRFAM S_DD_SRTFAM_LVL S_DD_SRTFAM_MAP	Hierarchy Hierarchy Levels Hierarchy Levels (Fields) User Defined Levels User Defined Levels.Sequence (Sequence)	No longer necessary. Customer-level targeting is supported by Target Levels in the Siebel Analytics repository.
Campaign Load Mappings (Contact Key Formats)	Obsolete			
Server tables (internal use)	Obsolete	S_DD_CUBE S_DD_DIM S_DD_DIM_DTL S_DD_EXTRACT S_DD_EXTRACT_CL S_DD_PROGEXPORT S_DD_SNPSHT_FMT S_DD_PPRPERFMEAS	Snapshot Elements	These tables were used by obsolete Siebel Server Components (Data Dictionary Manager and Marketing Server).

Obsolete Business Objects in Siebel Marketing

Upgrades: Releases 7.0.x, 7.5.x.

[Table 40 on page 393](#) lists the Siebel Repository business objects that are obsolete as of Release 7.7.

Table 40. Obsolete Siebel Marketing Business Objects as of Release 7.7

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

Table 40. Obsolete Siebel Marketing Business Objects as of Release 7.7

Business Object Name	Tools Project
DBM Financial Modeler	Mktg Financial Rollup
Data Dictionary	Data Dictionary (DBM)
External Contacts Mapping?	Server (DD)
Filters	Filters (DBM)
Hierarchical Attributes	Attributes (DBM)
Measure Attributes	Attributes (DBM)
Measures	Measures (DBM)
Segment	Segment (DBM)

Obsolete or Replaced Views in Siebel Marketing

Upgrades: Releases 7.0.x, 7.5.x.

Table 41 on page 394 lists the Release 7.x Siebel Repository views that are obsolete or have been replaced as of Release 7.7.

Table 41. Release 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Release 7.x Screen	Release 7.x View Caption	Outcome	Replacement View
All Campaigns (DBM) View	Campaign Management Screen	All Campaign Plans across Organizations	Replaced	Campaign Administration List
All Decisions List View	Decision Administration	All Decisions	Obsolete	
All Measures View	Marketing Administration	All Measures	Obsolete	
All Segments across My Organizations	Segment Screen	All Segments	Replaced	All Marketing Segments View
All Tables View	Marketing Administration	Tables	Obsolete	
Attribute Families View	Marketing Administration	Attribute Families	Obsolete	
Bound Measures View	Marketing Administration	Bound Measures	Obsolete	

Table 41. Release 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Release 7.x Screen	Release 7.x View Caption	Outcome	Replacement View
Campaign (DBM) Activities View	Campaign Occurrence Screen	Activities	Replaced	Campaign Activity View Admin
Campaign (DBM) Offers View	Campaign Management Screen	Offers	Replaced	Campaign Literature View Admin
Campaign (DBM) Team View	Campaign Management Screen	Team	Inactivated	
Campaign (DBM) Template Activities View	Campaign Management Screen	Activity Plans	Replaced	Campaign Activity Plans View
Campaign Admin Lists (SCW)	Campaign Screen (SCW)	Lists	Replaced	Campaign Admin Lists
Campaign Admin Objectives (SCW)	Campaign Screen (SCW)	Quotas	Replaced	Campaign Admin Objectives
Campaign Administration Detail (SCW)	Campaign Screen (SCW)	Contacts/Prospects	Replaced	Campaign Contacts Admin View - Org
Campaign Administration Detail - Owner Audit Trail View (SCW)	Campaign Screen (SCW)	Owner Audit Trail	Replaced	Campaign Administration Detail - Owner Audit Trail View
Campaign Administration Explorer View (SCW)	Campaign Screen (SCW)	Campaign Explorer	Replaced	Campaign Administration Explorer View
Campaign Administration List (SCW)	Campaign Screen (SCW)	Campaigns	Replaced	My Campaigns View
Campaign Administration List-More Info (SCW)	Campaign Screen (SCW)	More Info	Replaced	Campaign Administration List-More Info
Campaign Forecast (DBM) View-Cost Allocation Inputs	Campaign Management Screen	Cost Allocation Inputs	Replaced	Campaign Expenses View
Campaign Generated Lists View	Campaign Occurrence Screen	Exported Lists	Replaced	Campaign List Distribution View
Campaign Groups View - Admin (SCW)	Campaign Screen (SCW)	Groups	Inactivated	

Table 41. Release 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Release 7.x Screen	Release 7.x View Caption	Outcome	Replacement View
Campaign List	Campaign Screen (SCW)	Campaign List	Replaced	My Campaigns View
Campaign Literature View Admin (SCW)	Campaign Screen (SCW)	Offers	Replaced	Campaign Literature View Admin
Campaign Occurrence Contacts/ Prospects	Campaign Occurrence Screen	Campaigns > All Contacts/ Prospects across Organizations	Replaced	Campaign Contacts Admin View - All
Campaign Occurrence Contacts/ Prospects - Organization	Campaign Occurrence Screen	Campaigns > All Contacts/ Prospects	Replaced	Campaign Contacts Admin View - Org
Campaign Occurrence Contacts/ Prospects - Position	Campaign Occurrence Screen	Campaigns > My Contacts/ Prospects	Obsolete	
Campaign Occurrence Contacts/ Prospects Owner Audit View	Campaign Occurrence Screen	Campaigns > Owner Audit Trail	Replaced	Campaign Administration Detail - Owner Audit Trail View
Campaign Occurrence Email Status	Campaign Occurrence Screen	Campaigns > Email Status	Replaced	Campaign System Task View
Campaign Occurrence Offers View	Campaign Occurrence Screen	Offers	Replaced	Campaign Execution History View
Campaign Occurrence Timeline Gantt Chart View	Campaign Management Screen	Timeline	Replaced	Marketing Calendar - Campaigns Ax Gantt Chart View - My
Campaign Occurrences Responses View	Campaign Occurrence Screen	Campaigns > Responses	Replaced	Campaign Responses View
Campaign Occurrences View	Campaign Management Screen	Status	Replaced	Campaign Execution History View
Campaign Plan Activities View	Campaign Management Screen	Activities	Replaced	Campaign Activity View Admin

Table 41. Release 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Release 7.x Screen	Release 7.x View Caption	Outcome	Replacement View
Campaign Responses View (SCW)	Campaign Screen (SCW)	Responses	Replaced	Campaign Responses View
Campaign Results (DBM) Input Summary View	Campaign Management Screen	Results > Results Summary	Obsolete	
Campaign Results (DBM) View- Cost Inputs	Campaign Management Screen	Results > Cost Results	Replaced	Campaign Expenses View
Campaign Results (DBM) View-Cost Allocation Results	Campaign Management Screen	Results > Cost Allocation Results	Replaced	Campaign Expenses View
Campaign Results (DBM) View-List Results	Campaign Management Screen	Results > List Results	Replaced	Campaign Segment/ List Assumptions View
Campaign Results (DBM) View-Revenue Results	Campaign Management Screen	Results > Revenue Results	Obsolete	
Campaign Results (DBM) View-Segment Results	Campaign Management Screen	Results > Segment Inputs	Replaced	Campaign Segment/ List Assumptions View
Campaign Skill View (SCW)	Campaign Screen (SCW)	Assignment Skills	Replaced	Campaign Skill View
Campaign Team View - Admin (SCW)	Campaign Screen (SCW)	Team	Inactivated	
Campaign Waves View	Campaign Management Screen	Waves	Replaced	Program Schedule Detail View
Columns View	Marketing Administration	Fields	Obsolete	
Custom Measure Aggregation View	Marketing Administration	Custom Measure > Aggregation	Obsolete	
Custom Measure Details View	Marketing Administration	Custom Measure > More Info	Obsolete	
Custom Measure Restriction View	Marketing Administration	Custom Measure > Restriction	Obsolete	

Table 41. Release 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Release 7.x Screen	Release 7.x View Caption	Outcome	Replacement View
Customer Hierarchies View	Marketing Administration	Customer Hierarchies	Obsolete	
DBM Campaign Timeline Gantt Chart View	Campaign Management Screen	Timeline	Replaced	Marketing Calendar - Campaigns Ax Gantt Chart View - My
DD All Segments View	Segment Screen	All Segments across Organizations	Replaced	Marketing Segments Across All Organizations View
DD Segment Detail View	Segment Screen	Edit Segment	Replaced	SSO Mktg Segments Entry View
Database Synchronization	Marketing Administration	Data Retrieval	Obsolete	
Decision Wizard Player View	Decisions Administration Screen		Obsolete	
Decisions Detail Input Parameters View	Decisions Administration Screen	Identifiers	Obsolete	
Decisions Detail Output Parameters View	Decisions Administration Screen	Outputs	Obsolete	
Decisions Detail Session Parameters View	Decisions Administration Screen	Real Time Inputs	Obsolete	
Decisions Detail View	Decisions Administration Screen	More Info	Obsolete	
Enewsletter Attachments View	Offer Screen	eNewsletter > Attachments	Obsolete	
Enewsletter Email Profile View	Offer Screen	eNewsletter > Profile	Obsolete	
Enewsletter Offer Detail View	Offer Screen	eNewsletter > More Info	Obsolete	
Enewsletter Offer Related URLs View	Offer Screen	eNewsletter > Related URLs	Obsolete	
Enewsletter Offer Sections List View	Offer Screen	eNewsletter > Sections	Obsolete	

Table 41. Release 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Release 7.x Screen	Release 7.x View Caption	Outcome	Replacement View
Enewsletter Offer View	Offer Screen	eNewsletter Offers	Obsolete	
Enewsletter Rules View	Offer Screen	eNewsletter Rules	Obsolete	
External Contacts Mapping	Marketing Administration	Campaign Load Mapping	Obsolete	
Filters Detail View	Marketing Administration	Filters	Obsolete	
Filters View	Marketing Administration	Filters	Obsolete	
Financial Modeler (DBM) View	Campaign Management Screen	Forecast > Input Summary	Obsolete	
Financial Modeler (DBM) View-Fixed Costs	None	Financial Modeler	Obsolete	
Financial Modeler (DBM) View-Inbound Costs	None	Financial Modeler	Obsolete	
Financial Modeler (DBM) View-Input Costs	Campaign Management Screen	Forecast > Cost Inputs	Replaced	Campaign Expenses View
Financial Modeler (DBM) View-List Inputs	Campaign Management Screen	Forecast > List Inputs	Replaced	Campaign Segment/List Assumptions View
Financial Modeler (DBM) View-Outbound Costs	None	Financial Modeler	Obsolete	
Financial Modeler (DBM) View-Revenue Inputs	None	Financial Modeler	Obsolete	
Financial Modeler (DBM) View-Segment Inputs	Campaign Management Screen	Forecast > Segment Inputs	Replaced	Campaign Segment/List Assumptions View
Hierarchical Attribute List View	Marketing Administration	Hierarchical Attributes	Obsolete	
Joins View	Marketing Administration	Joins	Obsolete	

Table 41. Release 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Release 7.x Screen	Release 7.x View Caption	Outcome	Replacement View
List Formats View	Marketing Administration	Output List Layouts	Obsolete	
List Measures View	Marketing Administration	List Measures	Obsolete	
Marketing Plans Funds	Marketing Plans	Funds	Inactivated	
Measure Based Attributes View	Marketing Administration	Buckets	Obsolete	
My Campaigns (DBM) View	Campaign Management Screen	My Campaign Plans	Replaced	My Campaigns View
My Campaigns (DBM) View - More Info	Campaign Management Screen	My Campaign Plans	Replaced	Campaign Administration List-More Info
My Decisions List View	Decision Administration	My Decisions	Obsolete	
My Segments View	Segments	My Segments	Replaced	My Marketing Segments View
My Team's Decisions List View	Decision Administration	My Team's Decisions	Obsolete	
Program (DBM) Activities View	Program Occurrence Screen	Activities	Replaced	Program Plan Activities View
Program Container Timeline Gantt Chart View	Program Screen	Timeline	Replaced	Program Schedule Timeline View
Program Lists View	Program Stages Screen	Preview List	Replaced	Campaign List Distribution View
Program Occurrence Timeline Gantt Chart View	Program Screen	Timeline	Replaced	Marketing Calendar - Programs Ax Gantt Chart View - My
Program Occurrences View	Program Screen	Status	Obsolete	
Program Responses View	Program Occurrence Screen	Responses	Obsolete	
Program Snapshots View	Program Stages Screen	Snapshot Elements	Obsolete	
Program Stages View	Program Screen	Stages	Inactivated	

Table 41. Release 7.x: Obsolete or Replaced Siebel Marketing Views

Obsolete View Name	Release 7.x Screen	Release 7.x View Caption	Outcome	Replacement View
Related Events View for eNewsletter Offers	Offer Screen	Related Events	Obsolete	
Related Web Offers View f/ Enewsletter	Offer Screen	Related Web Offers	Obsolete	
Related Web Surveys View f/ Enewsletter	Offer Screen	Related Web Surveys	Obsolete	
Response Detail View (SME) - More Info	Response Screen	More Info	Replaced	Response Detail View (Detail)
Segment Criteria View	Segment Screen	More Info	Obsolete	
Segment Program Stages View	Segment Screen	Program Plans	Obsolete	
eChannel All Programs View (DBM)	eChannel Program Screen - OLD	All Program Plans across Organizations	Replaced	All Programs View (DBM)
eChannel All Programs across My Organizations	eChannel Program Screen	All Programs	Replaced	All Programs across My Organizations
eChannel Campaign Administration Detail	eChannel Campaign Management Screen - OLD	Contacts/ Prospects	Replaced	Campaign Contacts Admin View - Org
eChannel My Programs View (DBM)	eChannel Program Screen - OLD	My Program Plans	Replaced	My Programs View (DBM)
eChannel Program Container Timeline Gantt Chart View	eChannel Program Screen - OLD	Timeline	Replaced	Marketing Calendar - Programs Ax Gantt Chart View - My
eChannel Programs Detail View (DBM)	eChannel Program Screen - OLD	More Info	Replaced	Programs Detail View (DBM)
eChannel Segment Program Stages View	eChannel Segment Screen	Program Plans	Obsolete	

C

Upgrade Planning Worksheet

This appendix contains the following topic:

- “Master Upgrade Worksheet” on page 403

NOTE: The Upgrade Planning Worksheet should be photocopied and a copy given to each member of the upgrade team. Please do not write on the master copy in this appendix.

Master Upgrade Worksheet

- “Team Lead Summary”
- “DB2 Connect Information”
- “Siebel Development Environment Information”
- “Siebel Production Environment Information”

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 Name:

Path to Siebel File System:

Database Alias:

Schema Qualifier ID
(Max. 8 characters)

Security Group ID /
Grantee

Siebel Administrator User
Name

Siebel Administrator
Password

Siebel Administrator User
Group

Siebel User Group ID
(Max. 8 characters)

EIM User Group ID
(Max. 8 characters)

Database Name Prefix
(Max. 4 characters)

Name of Storage Control
File Used

4-KB Bufferpool

16-KB Bufferpool

32-KB Bufferpool

Index Bufferpool

Code Page/CCSID

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 Name: _____

Path to Siebel File System: _____

Database Alias: _____

Schema Qualifier ID (Max. 8 characters) Security Group ID / Grantee

Siebel Administrator User Name	Siebel Administrator Password	Siebel Administrator User Group
--------------------------------	-------------------------------	---------------------------------

Siebel User Group ID (Max. 8 characters)	EIM User Group ID (Max. 8 characters)	Database Name Prefix (Max. 4 characters)
--	---------------------------------------	--

Name of Storage Control File Used		
-----------------------------------	--	--

4-KB Bufferpool	16-KB Bufferpool	32-KB Bufferpool
-----------------	------------------	------------------

Index Bufferpool	Code Page/CCSID	
------------------	-----------------	--

NOTE: The Security Group ID is also known as the secondary authorization ID.

D

Columns Denormalized During Upgrade to Release 7.7

This appendix contains the following topics:

- “Denormalized Columns for 6.2.1 Siebel Financial Services Applications” on page 407
- “Denormalized Columns for 6.2.1 Siebel eBusiness Applications” on page 416
- “Denormalized Columns for 6.3 Siebel Industry Solutions” on page 423
- “Denormalized Columns for 7.0.4 Siebel eBusiness Applications” on page 433
- “Denormalized Columns for 7.0.4 Siebel Financial Services Applications” on page 435
- “Denormalized Columns for 7.0.4 Siebel Industry Solutions” on page 439
- “Denormalized Columns for 7.5.2 Siebel Industry Applications” on page 443

NOTE: If you reduced column lengths when you installed the Siebel Database Server on DB2 UDB for z/OS, you must review them before upgrading to Release 7.7. The upgrade does not recognize the denormalized columns.

Denormalized Columns for 6.2.1 Siebel Financial Services Applications

Table 42 lists columns that are denormalized during upgrades from Release 6.2.1 Siebel Financial Services applications to Release 7.7 Siebel Industry applications.

Table 42. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services

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_C D	[ACTIVITY_ID].[CAL _TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ACT_CAL_RSRC	ACT_TEMPLATE_F LG	[ACTIVITY_ID].[TEM PLATE_FLG]	S_EVT_ACT	TEMPLATE_FLG
S_ACT_CAL_RSRC	ACT_TODO_PLNEN D_DT	[ACTIVITY_ID].[TOD O_PLAN_END_DT]	S_EVT_ACT	TODD_PLAN_END_DT

Table 42. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
S_CASE_BU	CASE_TYPE_CD	[CASE_ID].[TYPE_CD]	S_CASE	TYPE_CD

Table 42. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
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

Table 42. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
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

Table 42. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
S_LOY_TXN_BU	TXN_TYPE_CD	[TXN_ID].[TYPE_CD]	S_LOY_TXN	TYPE_CD
S_MDF_BU	MDF_UID	[MDF_ID].[MDF_UID]	S_MDF	MDF_UID
S_OPTY_BU	OPTY_NAME	[OPTY_ID].[NAME]	S_OPTY	NAME

Table 42. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
S_OPTY_BU	SUM_WIN_PROB	[OPTY_ID].[SUM_WIN_PROB]	S_OPTY	SUM_WIN_PROB
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	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

Table 42. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
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_PROVIDR_FLG]	S_ORG_EXT	SRV_PROVIDR_FLG
S_ORG_DIST_LST	OU_ID	[ORG_PROD_ID].[OU_ID]	S_ORG_PROD	OU_ID

Table 42. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services

Target Table	Target Column	Denorm Path	Source Table	Source Column
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_PRIMDL_FCTR_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
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

Table 42. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
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

Table 42. Columns Denormalized During Upgrades from Release 6.2.1 Siebel Financial Services

Target Table	Target Column	Denorm Path	Source Table	Source Column
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 eBusiness Applications

Table 43 lists columns that are denormalized during upgrades from Release 6.2.1 Siebel eBusiness applications to Release 7.7 Siebel Industry applications.

Table 43. Columns Denormalized During Upgrades from Release 6.2.1 Siebel eBusiness 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	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

Table 43. Columns Denormalized During Upgrades from Release 6.2.1 Siebel eBusiness Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
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_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_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
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

Table 43. Columns Denormalized During Upgrades from Release 6.2.1 Siebel eBusiness Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
S_CTLGCT_RESI TM	RI_TYPE_CD	[RES_ITEM_ID].[TYPE_CD]	S_RESI TEM	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

Table 43. Columns Denormalized During Upgrades from Release 6.2.1 Siebel eBusiness Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
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

Table 43. Columns Denormalized During Upgrades from Release 6.2.1 Siebel eBusiness Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
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 43. Columns Denormalized During Upgrades from Release 6.2.1 Siebel eBusiness Applications

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_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
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

Table 43. Columns Denormalized During Upgrades from Release 6.2.1 Siebel eBusiness Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
S_PROJ_BU	PROJ_TYPE_CD	[PROJ_ID].[PROJ_TY PE_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_NA ME]	S_VOD	VOD_NAME
S_QUOTE_POSTN	QUOTE_NUM	[QUOTE_ID].[QUOT E_NUM]	S_DOC_QUOTE	QUOTE_NUM
S_SALES_MTHD_BU	SALES_METHOD_N AME	[SALES_METHOD_ID].[NAME]	S_SALES_METHOD	NAME
S_SRC_BU	SRC_NAME	[SRC_ID].[NAME]	S_SRC	NAME
S_SRV_REQ_BU	SR_TIT LE	[SRV_REQ_ID].[SR_ TITLE]	S_SRV_REQ	SR_TIT LE
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_I D].[NAME]	S_TMPL_PLANITE M	NAME
S_TMPL_PLNIT_BU	PLANITEM_TYPE_ CD	[TMPL_PLANITEM_I D].[TYPE_CD]	S_TMPL_PLANITE M	TYPE_CD
S_TMPL_PLNIT_BU	PLNIT_TMPL_TYP E_CD	[TMPL_PLANITEM_I D].[TMPL_TYPE_CD]	S_TMPL_PLANITE M	TMPL_TYPE_CD
S_USERLIST_BU	UL_NAME	[USERLIST_ID].[NA ME]	S_USERLIST	NAME

Table 43. Columns Denormalized During Upgrades from Release 6.2.1 Siebel eBusiness 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 6.3 Siebel Industry Solutions

Table 44 lists columns that are denormalized during upgrades from Release 6.3 Siebel Industry Solutions to Release 7.7 Siebel Industry applications.

Table 44. 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
S_ACCNT_POSTN	SRV_PROVDR_FLG	[OU_EXT_ID].[SRV_PROVDR_FLG]	S_ORG_EXT	SRV_PROVDR_FLG

Table 44. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
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_C D	[ACTIVITY_ID].[CAL _TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ACT_CAL_RSRC	ACT_TEMPLATE_F LG	[ACTIVITY_ID].[TEM PLATE_FLG]	S_EVT_ACT	TEMPLATE_FLG
S_ACT_CAL_RSRC	ACT_TODO_PLNEN D_DT	[ACTIVITY_ID].[TOD O_PLAN_END_DT]	S_EVT_ACT	TODO_PLAN_END_DT
S_ACT_CAL_RSRC	ACT_TODO_PLNST RTDT	[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_RPTEN D_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_DI SP_F LG	[ACTIVITY_ID].[CAL _DISP_FLG]	S_EVT_ACT	CAL_DI SP_FLG
S_ACT_EMP	ACT_CAL_TYPE_C D	[ACTIVITY_ID].[CAL _TYPE_CD]	S_EVT_ACT	CAL_TYPE_CD
S_ACT_EMP	ACT_EVT_STAT_C D	[ACTIVITY_ID].[EVT _STAT_CD]	S_EVT_ACT	EVT_STAT_CD
S_ACT_EMP	ACT_TEMPLATE_F LG	[ACTIVITY_ID].[TEM PLATE_FLG]	S_EVT_ACT	TEMPLATE_FLG
S_ACT_EMP	ACT_TODO_PLNEN D_DT	[ACTIVITY_ID].[TOD O_PLAN_END_DT]	S_EVT_ACT	TODO_PLAN_END_DT
S_ACT_EMP	ACT_TODO_PLNST RTDT	[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
S_AGREE_POSTN	AGREE_VALID_FL G	[AGREE_ID].[VALID _FLG]	S_DOC_AGREE	VALID_FLG

Table 44. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
S_CTLG_CAT_ISS	ISS_NAME	[ISS_ID].[NAME]	S_ISS	NAME
S_CTLG_CAT_LIT	LIT_NAME	[LIT_ID].[NAME]	S_LIT	NAME

Table 44. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
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

Table 44. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
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
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

Table 44. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
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_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
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	CONSUMER_OPTY_FLG	[OPTY_ID].[CONSUMER_OPTY_FLG]	S_OPTY	CONSUMER_OPTY_FLG

Table 44. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
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_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

Table 44. 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_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
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_PROVIDR_FLG]	S_ORG_EXT	SRV_PROVIDR_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_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
S_PERIOD_BU	PERIOD_NAME	[PERIOD_ID].[NAME]	S_PERIOD	NAME
S_POS_BU	POS_NUM	[POS_ID].[POS_NUM]	S_POS	POS_NUM

Table 44. Columns Denormalized During Upgrades from Release 6.3 Siebel Industry Solutions

Target Table	Target Column	Denorm Path	Source Table	Source Column
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_PRIMDL_FCTR_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_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

Table 44. 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_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 eBusiness Applications

Table 45 lists columns that are denormalized during upgrades from Release 7.0.4 Siebel eBusiness applications to Release 7.7 Siebel Industry applications.

Table 45. Columns Denormalized During Upgrades from Release 7.0.4 Siebel eBusiness 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].[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].[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_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 45. Columns Denormalized During Upgrades from Release 7.0.4 Siebel eBusiness 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

Table 45. Columns Denormalized During Upgrades from Release 7.0.4 Siebel eBusiness Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
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 46 lists columns that are denormalized during upgrades from Release 7.0.4 Siebel Financial Services applications to Release 7.7 Siebel Industry applications.

Table 46. 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].[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	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

Table 46. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
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_INSCLM_BU	INSCCLAIM_NUM	[INSCLM_ID].[INSCCLAIM_NUM]	S_INSCCLAIM	INSCCLAIM_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

Table 46. 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_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_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

Table 46. Columns Denormalized During Upgrades from Release 7.0.4 Siebel Financial Services Applications

Target Table	Target Column	Denorm Path	Source Table	Source Column
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_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 47 lists columns that are denormalized during upgrades from Release 7.0.4 Siebel Industry Solutions to Release 7.7 Siebel Industry applications.

Table 47. 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 47. 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].[INSC LAIM_NUM]	S_I NS_CLAI M	I NSCLAI M_NUM
S_I NSCLM_BU	REV_NUM	[INSCLM_ID].[REVI SION_NUM]	S_I NS_CLAI M	REVI SI ON_NUM
S_I NSCLMEL_BU	SEQ_NUM	[INSCLM_ELMNT_ID].[SEQ_NUM]	S_I NSCLM_ELMN T	SEQ_NUM
S_I NSCLMEL_BU	TYPE_CD	[INSCLM_ELMNT_ID].[TYPE_CD]	S_I NSCLM_ELMN T	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 47. 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 47. 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	PROVIDER_FLG	[CON_ID].[PROVIDER_FLG]	S_CONTACT	PROVIDER_FLG
S_PROD_INT_BU	PROD_CD	[PROD_INT_ID].[PROD_CD]	S_PROD_INT	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_ID	[SRV_REQ_ID].[SR_STAT_ID]	S_SRV_REQ	SR_STAT_ID

Table 47. 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 48 lists columns that are denormalized during upgrades from Release 7.5.2 Siebel Industry applications to Release 7.7 Siebel Industry applications.

Table 48. 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 48. 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

Index

Symbols

.srf file

See configuration file

Numerics

6.x

duplicate Row IDs, identifying and resolving 46, 54
repository, running Repository Preparation Wizard 259
Siebel Configuration data, preparing for upgrade 141

7.7

new features 18
rev A, new features 17
rev B, new features 17

7.7, planning upgrade

new database schema structure 86

7.x, planning upgrade

backup and recovery stages, planning 90
capacity planning 88
code page support, considering 88
IBM utilities, obtaining required 89
Job Stream Optimization utility 88
new database schema structure in 7.7 86
schedule, creating 91
security privileges, obtaining required 89
Siebel-scheduled mode or vendor-schedule mode 88
software and hardware, required 89
testing before production upgrade 87

A

access control

about 95
S_CONTACT and S_ORG-GROUP, about multi-org visibility for 95

adding

Siebel user 118

aggregate categories (new), reviewing 321

AIX memory allocation segment space, securing 104, 137

applet functionality, reviewing 314

applets

7.x applet scripts, about migrating to 302

applet models, about and list of 289
customizations, migrating to applets and views 295
migrating applets to release 7.x 285
new applet migration, reviewing 298
scripts, about migrating to release 7.x 291, 302

Application scripts

7.x, about migrating to 291
migrating 305

attachments

file system attachments, updating 374

audience for guide 19, 83

B

backing up

development database 118

backup and recovery stages, planning 90

business component definitions, changes to party model 98

Business Component scripts

7.x, about migrating to 291
migrating 305

business components

conflicts, resolving 331
custom business component configurations, migrating 188
join conflicts, resolving 276, 310
visibility properties, changing 344

Business Services scripts, migrating 305

C

call report attachments

file names, updating 353, 364
updating (procedure) 353, 364

capacity planning 88

Catalog visibility

See visibility modes, setting for access control

Cell Center, locating user logins that require resolution of duplicates 347

client configuration, migrating to Web client

See also postupgrade tasks; pre-upgrade tasks

about 285

applet and view models 289

applet migration, reviewing 298

- applet scripts, migrating to 7.x 302
- applets and views, migrating, about and wizards 285
- client-side interfaces, migration tasks 292
- hidden properties, exposing 294
- migrating Business Components, Business Services, and Application scripts 305
- outbound COM interfaces, migrating 306
- scripts, analyzing for migration 300
- scripts, migration tasks 291
- views, remigrating 299
- client-side interfaces**
 - outbound COM interfaces 292
 - outbound COM interfaces, migrating 306
- clustered indexes**
 - considerations for 99
- clustered indexes, considerations** 102
- code page**
 - considering support 88
- collation sequence**
 - See database sort order, verifying
- columns, denormalized**
 - 6.2.1 Siebel eBusiness applications 416
 - 6.2.1 Siebel Financial Services applications 407
 - 6.3 Siebel Industry Solutions 423
 - 7.0.4 Siebel eBusiness applications 433
 - 7.0.4 Siebel Financial Services applications 435
 - 7.0.5 Siebel Industry Solutions 439
 - 7.5.2 Siebel Financial Services applications 443
- configuration file**
 - new .srf file, compiling 282
 - new custom configuration file, producing 282
- copied objects**
 - upgrade inheritance, configuring objects for 253
- copied objects, automatic upgrade of**
 - upgrade ancestor, choosing 246
- custom business component configurations**
 - migrating 188
- custom database schema**
 - license keys, adding new 176
- custom indexes**
 - clustered indexes, considerations 102
 - custom temporary tables, dropping 102
- custom objects, remapping opportunity**
 - product migration objects** 355
- custom repository, safeguarding export file** 280
- Customer Repository, preparing for merge**

- See *also* repository merge, performing inheriting upgrade behavior, about 245
- New Customer repository, preparing to import 278, 281
- customized triggers, disabling** 150
- customized views, dropping** 150

D

- data migration**
 - party model, support of 98
- database schema**
 - new structure in 7.7 86
 - testing before production upgrade 87
- database server**
 - resetting, about 335
- database server configuration, verifying** 103
 - See *also* database sort order, verifying; database, preparing for upgrade
 - Siebel database expected growth during upgrade (table) 91
- database sort order, verifying**
 - about 94
- database template file, regenerating** 334
- database, preparing for upgrade**
 - AIX memory allocation segment space, securing 104, 137
 - database sort order, verifying 94
 - mobile and dedicated users, preparing for upgrade 132
 - New Customer Repository, preparing to import 278, 281
 - RDBMS software, upgrading 92
- DB2, upgrade planning**
 - new database schema structure in 7.7 86
- dedicated users, preparing for upgrade** 132
- deleting**
 - duplicate EIM mappings 332
- denormalized columns**
 - See columns, denormalized
- development environment upgrade**
 - integration objects, upgrading 343
 - new custom repository export file, safeguarding 280
- development environment, post-upgrade tasks**
 - business component and join conflicts, resolving 331
 - database server, about resetting 335
 - database template file, regenerating 334
 - developers or Siebel Tools clients, extracting 334

- EIM mappings, deleting duplicate 332
- new custom repository export file, exporting 333
- obsolete tables, reapplying custom extensions 329
- Siebel Configurator, postupgrade tasks 369
- development environment, pre-upgrade tasks**
 - virtual memory, about allocating sufficient 372
- development environment, upgrading**
 - 6.x duplicate Row IDs, identifying and resolving 46, 54
 - custom business component configurations, migrating 188
 - foreground performance, increasing 265
 - parameter validation, about 37
 - physical schema, about upgrading 185
 - repositories for an ASCII database, merging 265
 - repositories for an DB2 EBCDIC database, merging 269
 - repository merge, about 263
 - repository merge, categories of object differences 262
 - repository merge, optimizing performance 264
 - Repository Preparation Wizard, running 259
 - schema, synchronizing 186
 - Siebel Upgrade Wizard, resuming after first pause 178
 - Siebel Upgrade Wizard, resuming after second pause 182
 - Siebel Upgrade Wizard, resuming after third pause 185
 - SQL files, executing after first pause 177
 - SQL output files, executing after second pause 180
 - SQL output files, executing after third pause 184
 - statistics, updating 187
 - stored procedures and user-defined functions, deploying 167
 - upgrade inheritance, configuring objects for 253
- development upgrade**
 - clustered indexes, considerations 102
 - custom temporary tables, dropping 102
 - customized triggers, disabling 150
 - customized views, dropping 150
 - database server configuration, verifying 103
 - DB2 software, upgrading 146
 - development database, backing up 118
 - development repository projects, checking in 145
 - EIM tables, preparing for upgrade 146
 - interface table data, saving 145
 - intersection table maintenance and deduplication, preparing for 148
 - prerequisites 143
 - repositories, preparing for upgrade 150
 - repositories, renaming 152
 - schema, preparing storage layout 104
 - Siebel Configurator data 6.x, preparing for upgrade 141
 - Siebel user, adding 118
 - storage control file, extracting 110
 - storage control file, modifying 106
 - storage control file, reviewing 113
 - storage control file, validating 112
 - storage groups, creating 104
 - tablespace size, reviewing 115
- E**
 - eChannel, about preparing for upgrade 137**
 - EIM mapping**
 - deleting duplicates 332
 - EIM tables**
 - preparing for upgrade 146
 - Encryption Upgrade Utility**
 - RC2 encryption, about upgrading to 93
 - Enterprise Integration Manager**
 - integration object, upgrading 343
 - exporting**
 - new custom repository export file 333
- F**
 - features, new 17**
 - file system**
 - attachments, updating 374
 - Forecasting data, preparing for upgrade 138**
- G**
 - Generate Reporting Relationships button, exposing 340**
 - global deployment, postupgrade tasks**
 - global time zone, about setting up environment to support 342
 - global time zone, enabling after an upgrade 342
 - UTC delta columns, upgrading 343
 - global time zone**

See global deployment, postupgrade tasks
grid-based applets, reviewing 316
guide
 audience for 19, 83
 naming and typographical conventions 21
 z/OS, about 19

H

hardware, obtaining and software 89
hidden properties, exposing 294
household notes, updating file system 354

I

IBM DB2 UDB, preparing database for upgrade
 clustered indexes, considerations for 99
 RDBMS software, upgrading 92

IBM utilities
 obtaining required 89

ICL, about 248

importing
 development repository and schema definition 155

Incorporate Custom Layouts, about indexes 248

clustered indexes, considerations 102
 custom temporary tables, dropping 102

inheriting upgrade behavior, about 245

integration objects, upgrading 343

interface table
 development upgrade, saving data 145
 production upgrade, saving data 155

intersection tables
 preparing for maintenance and deduplication 148

item identifier, Web template 327

J

Job Stream Optimization utility
 using 88

join conflicts, resolving 276, 310

joins
 resolving conflicts 331

L

license keys
 custom database schema, adding new keys 176

log files
 archiving manually 243
 SiebSrvr log files, about saving 175

M

Master Upgrade Worksheet 403

Microsoft SQL Server
 Siebel Software Configuration utility, changing language utility runs 159

migrating, client configuration to Web client
 See *also* postupgrade tasks; pre-upgrade tasks

about 285
 applet and view models 289
 applet migration, reviewing 298
 applet scripts, migrating to 7.x 302
 applets and views, migrating 285
 client-side interfaces, migration tasks 292
 hidden properties, exposing 294
 migrating Business Components, Business Services, and Application scripts 305
 outbound COM interfaces, migrating 306
 scripts, analyzing for migration 300
 scripts, migration tasks 291
 views, remigrating 299

mobile users, preparing for upgrade 132

multilingual deployment, installing additional languages 92

multilingual seed data
 Siebel Database Server for DB2, installing 92

multi-value group shuttle applets, reviewing 318

N

naming conventions 21

O

objects
 opportunity product migration, remapping objects 355
 upgrade inheritance, configuring objects for 253

outbound COM interfaces, migrating 306

P

party model
 about 96
 business component definitions 98
 data migration, support of 98
 data model, changes to (diagram) 97
 obsolete tables in 7.x 97

physical schema
 upgrading, about 185

picklist visibility properties, changing 345

planning

- Employee Relationship Management (ERM),
about 120
- guidelines 83
- handheld devices 127
- Resonate Central Dispatch 128
- Workflow Designer 126

postupgrade tasks

- See also* pre-upgrade tasks; RC2 encryption,
upgrading to
- asset-based ordering 361
- business components and join conflicts,
resolving 276, 310
- call report attachments, updating 353,
364
- dock object and rule definitions, verifying are
correct 353, 371
- file system attachments, updating 374
- Generate Reporting Relationships button,
exposing 340
- household notes and attachments, updating
file system 354
- inbound workflows, updating 368
- integration objects, upgrading 343
- log files, manually archiving 243
- new custom configuration file,
producing 282
- new custom repository export file,
safeguarding 280
- Purchase Orders 360
- quotes 363
- reporting relationships, generating
(procedure) 340
- reporting relationships, generating,
about 340
- seeded workflows, incorporating prior
customizations into 368
- seeded workflows, using 368
- symbolic string model, upgrading to 335
- Transaction Amount field in Payment lines,
updating 360
- Universal Inbox 359
- user logins, locating that require resolution of
duplicates 347
- visibility modes, setting for access
control 344

pre-upgrade tasks

- AIX memory allocation segment space,
securing 104, 137
- database sort order, verifying 94
- mobile and dedicated users, preparing for
upgrade 132
- New Customer Repository, preparing to
import 278, 281

- RDBMS software, upgrading 92

procedures

- stored procedures, deploying and user-
defined functions 167

production environment repositories

- preparing for upgrade 156

production environment upgrade

- development environment, upgrading
without 175
- global time zone, about operating in 342
- New Customer Repository, preparing to
import 278, 281
- preparing production upgrade on
UNIX 164
- Transaction Amount field in Payment Lines,
updating 360
- tuning production upgrade scripts 377

production environment, upgrading

- channel data, migrating 215
- configuration data, migrating 219
- control cards and schema DDL files,
transferring to zSeries 200
- FINS, migrating configuration data 220
- Gen Primary 7.7 indexes. rebuilding 231
- Gen Primary 7.7 SQL, migrating 232
- household data, migrating 216
- index DDL, executing and rebuild
indexes 214
- index schema, transferring to zSeries 224
- interface tables, generating and
dropping 197
- intersection tables, maintaining 234
- ISS data, migrating 221
- JCL, preparing 225
- JCL, SQL, and binary files, transferring to
zSeries 191
- load jobs, running on the target
database 213
- log tables, loading on the source
database 210
- log tables, loading on the target
database 211
- new EIM indexes, creating 233
- old indexes, dropping and creating new
indexes 229
- preschm data, migrating 217
- preschm data, migrating for FINS 218
- schema and JCL files, preparing on
zSeries 201
- Siebel Upgrade Wizard, resuming after first
pause 199
- Siebel Upgrade Wizard, resuming after second
pause 222
- Siebel-Scheduled jobs, preparing on

- zSeries 206
- source database, running unload jobs 212
- stored procedure, installing on zSeries 205
- table creating maintenance jobs,
 - running 198
- table edit maintenance jobs, running 198
- target Siebel log tables, loading 228
- zSeries setup dataset, defining and
 - allocating 191
- zSeries upgrade environment, setting
 - up 193
- zSeries, prerequisites for upgrade 190
- production environments, postupgrade tasks**
 - mobile and connected clients,
 - upgrading 337
 - regional servers, upgrading 337
- production upgrade**
 - testing before 87
- production upgrade, prerequisites**
 - about 153
 - development repository and schema
 - definition, importing 155
 - interface table data, saving 155
 - production environment repositories,
 - preparing for upgrade 156
 - source database values 154
 - target database values 154
- Q**
- quotes**
 - upgrade, preparing for 138
- Quotes data, preparing for upgrade** 138
- R**
- RC2 encryption, upgrading to**
 - about 93
- RDBMS software, preparing for upgrade** 92
- recovery**
 - planning and backup stages 90
- reporting relationships**
 - about generating 340
 - Generate Reporting Relationships button,
 - exposing 340
 - generating (procedure) 340
- repositories**
 - development environment repositories,
 - preparing for upgrade 150
 - development environment repositories,
 - renaming 152
 - new custom repository export file,
 - exporting 333
 - production environment repositories,
 - preparing for upgrade 156
 - repositories for an ASCII database,
 - merging 265
 - repositories for an DB2 EBCDIC database,
 - merging 269
 - repository merge, about 263
 - repository merge, categories of object
 - differences 262
 - repository merge, increasing the foreground
 - performance 265
 - repository merge, optimizing
 - performance 264
- repositories, upgrading**
 - about 38
 - EIM temporary columns, generating 277
 - Incorporate Custom Layouts (ICL) 248
 - postmerge utilities, about 248
 - renaming 254
 - upgrade behavior, inheriting 245
- repository merge results**
 - determining success 271
 - reviewing conflicts 273
 - reviewing deleted objects 275
 - reviewing obsolete objects 275
- Repository Preparation Wizard**
 - running on 6.x repository 259
- rich text controls, revising** 320
- Row IDs**
 - 6.x, identifying and resolving 46, 54
- RTCs, revising** 320
- S**
- S_CONTACT, multi-org visibility for** 95
- S_LIT files, upgrading to S_CB_ASSET_VER files** 374
- S_ORG_GROUP, multi-org visibility for** 95
- S_PARTY**
 - See party model
- S_SRC_PAYMENT table, about columns added** 98
- saving**
 - development upgrade interface table
 - data 145
 - production upgrade interface table
 - data 155
- schedule**
 - planning version 7.x 91
- schema**
 - physical schema, about upgrading 185
 - storage layout, preparing 104
 - synchronizing 186
- scripts**
 - applet scripts, about migrating to 7.x 302

- business components, business services, and applications scripts, migrating 305
- migrating to release 7.x 291
- migration, analyzing for 300
- security**
 - obtaining required privileges 89
- seed data, installing multilingual seed data** 92
- shuttle applets, reviewing** 318
- Siebel Application Integration (EAI), about using SAP and upgrading** 27
- Siebel Business application, upgrading**
 - multilingual deployment, installing additional languages 92
- Siebel Configurator**
 - 6.x data, preparing for upgrade 141
 - postupgrade tasks 369
- Siebel data model**
 - access control, about 95
 - additional considerations 98
 - party model, business component definitions 98
 - party model, changes to data model (diagram) 97
 - party model, data migration in support of 98
 - party table, about 96
 - party table, obsolete tables in 7.x 97
 - S_SRC_PAYMENT Table, columns added to 98
- Siebel Database Schema, upgrading**
 - restarting upgrade 165
- Siebel Database Server software**
 - multilingual deployment, installing additional languages 92
- Siebel eBusiness applications**
 - denormalized columns for 6.2.1 416
 - denormalized columns for 7.0.4 433
- Siebel eChannel, about preparing for upgrade** 137
- Siebel Enterprise Integration Manager**
 - deleting duplicate EIM mappings 332
- Siebel Financial Services**
 - application tables, preparing for upgrade 141
 - call report attachments, updating 353, 364
 - call report attachments, updating file names 353, 364
 - denormalized columns for 6.2.1 407
 - denormalized columns for 7.0.4 435
 - denormalized columns for 7.5.2 443
 - household data integrity, verifying 139
 - household notes and attachments, updating
 - file system 354
 - opportunity product migration 355
- Siebel Industry Solutions**
 - denormalized columns for 6.3 423
 - denormalized columns for 7.0.4 439
- Siebel Marketing**
 - obsolete 7.x business objects 393
 - obsolete 7.x data 387
 - obsolete 7.x views 394
- Siebel Purchase Orders, updating Transaction Amount field** 360
- Siebel Quotes, about preparing upgrade for** 138
- Siebel Repository**
 - See also* Siebel Repository, preparing for upgrade; repository merge, performing
 - new custom repository, backing up 280
- Siebel Repository, preparing for upgrade**
 - See also* Siebel Repository; repository merge, performing
 - development environment repositories, preparing 254
 - dock objects and visibility rules, recording 103
- Siebel Software Configuration utility**
 - language, changing 159
 - The Siebel Upgrade Wizard, about 38
- Siebel Upgrade Wizard**
 - about 38
 - first pause, resuming after 178
 - second pause, resuming 182
 - third pause, resuming after 185
- Siebel user**
 - adding 118
- Siebel Workflow**
 - inbound workflow, updating 368
 - seeded workflows, incorporating prior customizations into 368
 - seeded workflows, using 368
- SiebSrvr log files, about saving** 175
- software, obtaining and hardware** 89
- SQL**
 - output files, about executing after second pause 180
 - output files, about executing after second pause stage 2 182
 - output files, executing after second pause stage 1 180
 - output files, executing after third pause 184
- .srf file**
 - See* configuration file
- statistics, updating** 187

storage control file

- extracting 110
- modifying 106
- reviewing 113
- validating 112

storage groups

- creating 104

stored procedures

- deploying and user-defined functions 167

T**tables**

- custom temporary tables, dropping 102
- EIM tables, preparing for upgrade 146
- interface table data, saving 155
- intersection tables, preparing maintenance and deduplication 148
- obsolete tables, reapplying custom extensions 329
- table creation maintenance jobs, running 198
- table edit maintenance jobs, running 198

tables, preparing for upgrade

- clustered indexes, considerations 99
- DB2 UDB long columns, identifying for truncation 91, 92, 94, 102, 132, 278, 281

tablespace

- size, reviewing 115

testing

- before production upgrade 87

Tools repository, renaming 254**Transaction Amount field in Payment lines, updating 360****triggers**

- disabling customized triggers 150

tuning production upgrade scripts 377**typographical conventions 21****U****upgrade inheritance**

- configuring objects for 253

Upgrade Planning Worksheet 403**upgrade, planning for**

- resources 23

upgrading ancestor, choosing an 246**upgrading inheritance**

- parent to descendents, propagating changes from 247

user

- adding Siebel user 118

UTC conversion utility, upgrading 343**utilities**

- IBM utilities, obtaining required 89
- Job Stream Optimization utility, using 88

V**version 6.x**

- duplicate Row IDs, identifying and resolving 46, 54
- repository, running Repository Preparation Wizard 259
- Siebel Configurator data, preparing for upgrade 141

version 7.7

- new features 18
- rev A, new features 17
- rev B, new features 17

version 7.7, planning upgrade

- new database schema structure 86

version 7.x, planning upgrade

- backup and recovery stages, planning 90
- capacity planning 88
- code page support, considering 88
- IBM utilities, obtaining required 89
- Job Stream Optimization utility 88
- new database schema structure in 7.7 86
- schedule, creating 91
- security privileges, obtaining required 89
- Siebel-scheduled mode or vendor-schedule mode 88
- software and hardware, required 89
- testing before production upgrade 87

view tabs, verifying display 324**views**

- customization, migrating to applets and views 295
- dropping customized views 150
- migrating to release 7.x 285
- remigrating views 299
- view models, about and list of 289
- visibility properties, changing 346

virtual memory

- development environment, about allocating sufficient 372

visibility filters, reviewing 322**visibility modes, setting for access control**

- about and Catalog visibility 344
- Auto Query Mode, picklists that start in (list of) 345
- business component visibility properties, changing 344
- business components, visibility type used (list of) 344
- picklists visibility properties, changing 345
- views visibility properties, changing 346

views, visibility type used (list of) 346

W

Web Client Migration Wizard

customizations, migrating to applets and views 295

Web client scripting, about in release

7.x 291

Web template item identifier 327

workflows

inbound, updating 368
 seeded, incorporating prior customizations into 368
 seeded, using 368
 test to production, transitioning 369

worksheet

Upgrade Planning 403

Z

z/OS

upgrade guide, about 19

zSeries

control cards and schema DDL files, transferring 200
 index schema, transferring to 224
 JCL, SQL, and binary files, transferring to zSeries 191
 schema and JCL files, preparing 201
 setup dataset, defining and allocating 191
 Siebel-Scheduled jobs, preparing on 206
 stored procedures, installing on 205
 upgrade environment, setting up 193
 upgrade prerequisites 190

