



# **Siebel Anywhere Administration Guide**

Version 8.0

December 2006

**ORACLE®**

Copyright © 2005, 2006, Oracle. All rights reserved.

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

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

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

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

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

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

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

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

# Contents

## **Chapter 1: What's New in This Release**

## **Chapter 2: Siebel Anywhere Concepts**

Features and Benefits of Siebel Anywhere	9
How Siebel Anywhere Is Used	9
Process of Using Siebel Anywhere	10
Siebel Anywhere Screens and Views	11
Siebel Anywhere Wizards and Utilities	11
Siebel Anywhere File Attachments	12
How Siebel Anywhere Versions Work	13
Crucial Siebel Anywhere Terminology	18
Siebel Anywhere Deployment Recommendations	24

## **Chapter 3: Upgrade Planning and Preliminary Tasks**

Process of Planning and Preparing for Kit Creation	25
Determining Upgrade Requirements	26
Identifying Software to Be Replaced or Added	26
Evaluating Siebel Anywhere as an Upgrade or Delivery Method	27
Identifying Computers and Users to Receive Upgrades	28
Identifying Configurations to Deliver Upgrade Components	28
Identifying Files to Include in Upgrade Kits	30
Choosing Required or Optional Upgrade Types	30
Determining Version Setting Values	31
Other Preliminary Upgrade Tasks for Specific Upgrade Kit Types	35
Determining Required Upgrade Sequences	37
Planning Upgrade Test Details	37
Creating Needed Infrastructure Elements	38
Modifying and Creating Siebel Anywhere Configurations	38
Setting Up Custom Siebel Anywhere Upgrade Components	43
Verifying Connections to the Siebel File System	47
Verifying Availability of Siebel Anywhere	47
Preparing Upgrade Kit Contents	48

Preparing Contents for a Siebel Upgrade Wizard Upgrade Kit	48
Preparing Contents for a Siebel Executables Upgrade Kit	49
About Reducing Siebel Client Executables Kit Size	51
Preparing Contents for Repository File or Third-Party Upgrade Kits	52
Process of Creating a Delta Install Siebel Client Executables Upgrade Kit	52
Creating a Delta Patch File	53
Process of Creating a Delta Repository File Upgrade Kit	55
Compiling an SRF for Use in a Delta Repository File	56
Preparing a Delta Repository File	57
Synchronizing Components	59

## **Chapter 4: Defining Upgrade Kits**

Running the Upgrade Kit Wizard	62
Defining a Siebel Upgrade Wizard Upgrade Kit	62
Defining a Siebel Configuration File (CFG) Upgrade Kit	64
Defining a Siebel Database Schema Upgrade Kit	66
Defining a Siebel Client Executables Upgrade Kit	70
Defining a Siebel Repository File Upgrade Kit	76
Defining a Third-Party Software Upgrade Kit	80
Defining a Siebel Customer Revisions Upgrade Kit	84
Defining a Custom Component Upgrade Kit	88
Viewing Upgrade Kit Properties	88

## **Chapter 5: Activating, Applying, and Distributing Upgrade Kits**

Process of Completing Upgrade Kit Creation	93
Activating an Upgrade Kit	93
Controlling the Order of Kit Installation	94
Deactivating an Upgrade Kit	95
Applying an Upgrade Kit	96
Converting an Optional Kit to a Required Kit	98
Distributing Upgrade Kits	98
Process of Limiting Distribution of an Upgrade Kit	100
Distributing Different CFG Files to Different Users	100
Disabling Siebel Anywhere	102

## Chapter 6: Retrieving, Installing, and Testing Upgrade Kits

Process of Retrieving, Installing, and Testing Upgrade Kits	105
Retrieving and Installing Upgrade Kits	106
Retrieving Optional Upgrade Kits for Mobile Web Clients	106
Retrieving Optional Upgrade Kits for Developer Web Clients	108
Retrieving Required Upgrade Kits	109
Launching the Upgrade Wizard	111
About Client Upgrade Error Recovery	111
Retrieving Upgrade Kits for Siebel Regional Node Servers	111
Testing Upgrade Kits	113
Uninstalling Siebel Maintenance Releases or Patches	115

## Chapter 7: Supplementary Information for Specific Upgrade Types

Distributing a Siebel Maintenance Release or Patch	117
Distributing Additional Language Packs	118
Example of Global Deployment with Siebel Anywhere	119
Example of Creating Multiple Kits of the Same Type	122
Performing Database Schema Updates	123
Process of Updating a Siebel Database Schema	123
About What Happens During a Database Schema Upgrade	124
About Changing the Database Schema in a Development Environment	125
About Moving Database Schema Changes Between Environments	125
About Creating Needed Upgrade Kits for Database Schema Changes	125
About Preparing Mobile Web Clients for Database Schema Changes	126
Process of Preparing Regional Node Servers for Database Schema Changes	127
Checking Regional Node Server Parameters Before a Database Schema Update	127
About Installing Database Schema Changes on Mobile Web Clients	129
About Installing Database Schema Changes on Regional Node Servers	129
Deleting Old Upgrade Kits	131
Testing for Upgrade Kit Component Consistency	132
Creating Kits That Check DLL File Versions	133
Example of Constructing a Third-Party Upgrade Kit	134

## Appendix A: Troubleshooting for Siebel Anywhere

Upgrade Kit Wizard Error at Finish	137
------------------------------------	-----

New Upgrade Kit Status “Request Submitted”	137
New Upgrade Kit Status “In Progress”	139
Unable to Download Upgrade Kit	139
Required Kit Cannot Be Installed	141
Failed Upgrade Kit Locks Administrator’s Account	141
Converting a Required Kit to an Optional Kit	142
Changing Version Settings After Distribution	143

## Index

# 1

## What's New in This Release

### **What's New in Oracle's Siebel Anywhere Administration Guide, Version 8.0**

This guide has been updated to reflect product name changes. It was previously published as *Siebel Anywhere Administration Guide, Version 7.8*.





# 2

## Siebel Anywhere Concepts

This chapter contains the following information:

- A brief description of the features and benefits provided by Siebel Anywhere. See [“Features and Benefits of Siebel Anywhere” on page 9](#).
- A high-level overview of how Siebel Anywhere is used by administrators and end users. See [“How Siebel Anywhere Is Used” on page 9](#).
- Information about how Siebel Anywhere uses version checks to perform its functions. See [“How Siebel Anywhere Versions Work” on page 13](#).
- Definitions of a few terms you must understand when administering Siebel Anywhere. See [“Crucial Siebel Anywhere Terminology” on page 18](#).
- Recommendations for the deployment of Siebel Anywhere within your organization. See [“Siebel Anywhere Deployment Recommendations” on page 24](#).

## Features and Benefits of Siebel Anywhere

Siebel Anywhere allows the Siebel system administrator to distribute software upgrades to Siebel Developer Web Clients, Siebel Mobile Web Clients, and Siebel Servers. Upgrades can include custom configurations, some new versions of Siebel Business Applications (as licensed), custom extensions to the database schema, custom files, or third-party files or applications.

**NOTE:** Siebel Anywhere is not designed to apply kits to Siebel Tools.

Siebel Anywhere provides the following mechanisms:

- Configuration control to make sure users are connecting to a system with the appropriate software components.
- Creation and packaging of upgrade components for distribution to users.
- Distribution to provide users with acceptable version levels of upgrade components.

For information about how administrators and end users work with Siebel Anywhere, see [“How Siebel Anywhere Is Used” on page 9](#). For information about how Siebel Anywhere performs its functions, see [“How Siebel Anywhere Versions Work” on page 13](#).

## How Siebel Anywhere Is Used

This section contains four types of information concerning how Siebel Anywhere is used:

- An overview of the process that administrators and end users follow when administering and using Siebel Anywhere. See [“Process of Using Siebel Anywhere” on page 10](#).

- Information about the screens and views used during this process. See [“Siebel Anywhere Screens and Views” on page 11](#).
- Information about wizards and utilities used during this process. See [“Siebel Anywhere Wizards and Utilities” on page 11](#).
- Information about how file attachments are handled during this process. See [“Siebel Anywhere File Attachments” on page 12](#).

## Process of Using Siebel Anywhere

The process for using Siebel Anywhere consists of the following steps:

- 1 Determine your upgrade requirements. For instructions concerning this step, see [“Determining Upgrade Requirements” on page 26](#).
- 2 As a Siebel system administrator, create any required configurations, dynamic configuration assignments, and components. For a discussion of components and configurations, see [“Crucial Siebel Anywhere Terminology” on page 18](#). For instructions concerning this step, see [“Creating Needed Infrastructure Elements” on page 38](#).
- 3 Define an upgrade kit to meet your requirements. For instructions concerning this step, see [Chapter 4, “Defining Upgrade Kits.”](#)
- 4 Activate the upgrade kit. Activating a kit gathers the files to be included in the upgrade kit and compresses them into a single archive on the Siebel File System. For instructions concerning this step, see [“Activating an Upgrade Kit” on page 93](#).
- 5 Apply the upgrade kit. Applying a kit updates a compiled information string with the component version information. For instructions concerning this step, see [“Applying an Upgrade Kit” on page 96](#).
- 6 Distribute the upgrade kit to test users for testing. Test users are specified by a membership in a test configuration. For instructions concerning this step, see [“Distributing Upgrade Kits” on page 98](#).
- 7 As a client belonging to the test configuration, test the upgrade kit by using one of the following methods:
  - For a required kit and a Mobile Web Client test user, test the kit by synchronizing.
  - For a required kit and Developer Web Client test user, test the kit by logging in to the Siebel application.
  - For an optional kit and either a Mobile Web Client test user or a Developer Web Client test user, test the kit by selecting and upgrading the applicable component from the Component Upgrades view in the User Preferences screen.For further instructions concerning this step, see [Chapter 6, “Retrieving, Installing, and Testing Upgrade Kits.”](#)
- 8 As a Siebel system administrator, correct any problems discovered during testing. You may find helpful information in [Appendix A, “Troubleshooting for Siebel Anywhere.”](#) Repeat client testing as necessary.

- 9 Use additional configurations to distribute the upgrade kit for wider use, and notify the appropriate users that it is available.
- 10 Mobile Web Client users retrieve the upgrade by synchronizing (if the kit is required) or by using the Component Upgrades view in the User Preferences screen (if the kit is optional). Developer Web Client users are automatically prompted to retrieve the upgrade upon logging in to the Siebel application.

**NOTE:** As part of your planning process, you may also find it helpful to consult [Chapter 7, “Supplementary Information for Specific Upgrade Types,”](#) which contains additional useful information for particular kinds of upgrades.

## Siebel Anywhere Screens and Views

Most administrative tasks concerned with Siebel Anywhere are done using the Administration - Siebel Anywhere screen. You must be logged on to the Siebel application as a Siebel administrator to access this screen. The Siebel database installation process creates a Siebel administrator account that can be used to perform the tasks described in this guide. For information about this, see *Siebel Installation Guide* for the operating system you are using.

**CAUTION:** Do *not* perform system administrative functions on your local database or in disconnected mode. Although there is nothing to prevent you from doing this, it can have serious consequences. Examples include data conflicts, potential for overly large local database, and potential for a large number of additional transactions to route.

Siebel Anywhere clients use the Component Upgrades view of the User Preferences screen for retrieving and installing optional upgrade kits. This view also allows clients to see their current update status.

This guide provides additional information about these screens and views in later chapters, as part of the information that accompanies specific procedures.

## Siebel Anywhere Wizards and Utilities

This section briefly describes the following Siebel Anywhere wizards and utilities, which are used during various stages of upgrades involving Siebel Anywhere:

- The Upgrade Kit Wizard
- The Upgrade Kit Builder
- The Upgrade Wizard

### The Upgrade Kit Wizard

The Siebel Upgrade Kit Wizard is the task-oriented feature Siebel administrators must use to define an upgrade kit. The Upgrade Kit Wizard uses a Web-enabled user interface to prompt the administrator for the required information. The Upgrade Kit Wizard performs the following functions:

- Validates the parameters supplied by the administrator

- Handles file attachment uploads for certain types of upgrade kits
- Specifies upgrade kit items
- Specifies the sequence of events in the upgrade being defined
- Submits a schedule-mode server request to invoke the *Upgrade Kit Builder* when kit definition is complete

For more information about using the Upgrade Kit Wizard, see [Chapter 4, "Defining Upgrade Kits."](#)

**NOTE:** Do not confuse the Upgrade *Kit* Wizard with the Upgrade Wizard, which is described later in this section. The Upgrade *Kit* Wizard constructs the upgrade kit. The Upgrade Wizard reads the upgrade kit and installs it.

## The Upgrade Kit Builder

Upgrade Kit Builder is part of the Siebel Anywhere Component Group. It constructs upgrade kits on the server, based upon definitions of kits collected by the *Upgrade Kit Wizard*. During the construction process, the Administrator can monitor the progress of the Upgrade Kit Builder by navigating to Administration - Server Management > Jobs and querying for Upgrade Kit Builder in the Jobs list or by checking the Upgrade Kit Builder log file. The associated log file is `UpgKitBldr_task id.log`.

## The Upgrade Wizard

The Siebel Upgrade Wizard is a stand-alone executable that reads and installs upgrade kits on Mobile or Developer Web Clients or servers. The Siebel Upgrade Wizard also tracks the upgrade process and automatically recovers from errors. In the Windows environment the associated log file is `UpgWiz.log`. In the UNIX environment the associated log file is `svrupgwiz1.log`.

**NOTE:** Do not confuse the Upgrade Wizard with the Upgrade *Kit* Wizard, which is described earlier in this section. The Upgrade *Kit* Wizard constructs the upgrade kit. The Upgrade Wizard reads the upgrade kit and installs it.

During the installation of an upgrade kit, the Upgrade Wizard creates a backup of affected files in the `\temp` or `\upgrade` folders. If an error occurs during the upgrade, the Upgrade Wizard attempts to roll back the changes and restore the machine to its original state. Subsequently, when the user starts the Siebel client, Siebel Business Applications detects that an upgrade is either in progress or has failed and notifies the user.

The Upgrade Wizard cannot roll back to the previous version after the upgrade has been installed successfully. After a successful upgrade, the Upgrade Wizard deletes the backup files. Consequently, restoring to a previous configuration is not possible.

## Siebel Anywhere File Attachments

This section briefly describes how Siebel Anywhere works with file attachments. Siebel Anywhere stores two kinds of file attachments in the Siebel File System:

- **Individual upgrade kit item files.** Upgrade kit item files are files that are included when a particular upgrade kit is built. Examples include CFG files for configuration upgrade kits and Siebel Repository Files (SRF files) for repository upgrade kits. Depending on the component type being upgraded, an upgrade kit may need to include zero, one, or multiple upgrade kit item files. These files are stored in the Siebel File System after you click Finish in the Upgrade Kit Wizard. The files should be visible in the Siebel File System as soon as the status of the upgrade kit record is Pending in the Upgrade Kits view. In the Siebel File System, individual upgrade kit item files are assigned file names that have the format `S_UPG_KIT_IARG_ROW_ID_REV_NO.SAF`.
- **Complete upgrade kit files.** These are the files that end users will download. Each upgrade kit file contains instructions for updating a specific Siebel component. Some types of upgrade kit files also contain one or more upgrade kit item files, as described earlier in this section. If upgrade kit item files are specified in the Upgrade Kit Wizard, the specified files are incorporated into the complete upgrade kit file when the kit is built. Complete upgrade kit files are stored in the Siebel File System after you activate the upgrade kit. In the Siebel File System, complete upgrade kit files are assigned file names that have the format `S_UPG_KIT_ROW_ID_REV_NO.SAF`.

The file name extension SAF is used for all Siebel file attachments, including Siebel Anywhere file attachments. In Siebel Anywhere file attachment names, *ROW\_ID* is a unique number combination that identifies the upgrade kit record in the database, and *REV\_NO* indicates whether the kit has been revised. If you deactivate and reactivate a kit, a new complete upgrade kit file is created, using the same *ROW\_ID* value but a different *REV\_NO* value in the file name.

## How Siebel Anywhere Versions Work

This section describes how Siebel Anywhere uses version settings to perform its functions, beginning with some brief definitions of essential terminology.

Siebel Anywhere packages and delivers certain kinds of software using special files called *upgrade kits*. For more information about upgrade kits, see [“Upgrade Kits” on page 19](#).

A software module that is upgraded as a single unit is called a Siebel Anywhere *component*. Examples of components include Siebel configuration files, Siebel database schemas, Siebel Executables, Siebel repository files, third-party software, and customer revisions. Any component that needs an upgrade must have its own upgrade kit. For more information about Siebel Anywhere components, see [“Upgrade Components” on page 20](#).

A Siebel application server or Siebel client that has been associated with one or more Siebel Anywhere components is called a Siebel Anywhere *subscriber*. The association between a subscriber and a set of components is not direct; the association is formed by means of the subscriber’s membership in an upgrade *configuration*, which is a definition of a setup used by a particular group of users, such as Siebel Call Center Clients or Siebel Sales Clients. A configuration associates a group of subscribers with the specific set of upgrade components that those subscribers need to have managed and maintained. For more information about configurations, see [“Upgrade Configurations” on page 22](#). For more information about subscribers, see [“Siebel Anywhere Subscribers” on page 23](#).

Siebel Anywhere stores and checks several kinds of version information to determine whether a particular subscriber can or should use a particular upgrade kit. To create upgrade kits that have the effects you want, you must understand how these versions are specified, stored, and used. The following paragraphs briefly describe the kinds of version information that Siebel Anywhere uses and how Siebel Anywhere uses them. The information is divided into the following sections:

- [“About Specifying a New Version Number for a Component” on page 14](#)
- [“About Specifying Versions That Can Use the Upgrade Kit” on page 14](#)
- [“About Specifying Acceptable Version Ranges for Upgrade Components” on page 15](#)
- [“About Storing Version Information” on page 16](#)
- [“About How Siebel Anywhere Component Versions Are Used” on page 17](#)

## About Specifying a New Version Number for a Component

An administrator who creates an upgrade kit specifies the version number that the component being upgraded will have after the kit is used to install the upgrade. This version number is called the New Version number.

For example, if the administrator sets the value of New Version to 3 when creating a new customer revision upgrade kit, a Mobile Web Client who successfully installs that upgrade kit will be upgraded to version 3 of the customer revision component.

## About Specifying Versions That Can Use the Upgrade Kit

The administrator who creates an upgrade kit also specifies which previous versions of the component are compatible with the upgrade kit. If one or more specific versions are specified, a client or server must have one of those versions already installed to download and use the upgrade kit.

The following Upgrade Kit Wizard settings are used to specify the acceptable range of versions for downloading the kit:

- Minimum Old Version
- Maximum Old Version

These settings are specified as part of the process of defining the upgrade kit. For more information about defining upgrade kits, see [Chapter 4, “Defining Upgrade Kits.”](#)

As an example, suppose you assign the following values when creating a customer revision upgrade kit:

Minimum Old Version = 1  
Maximum Old Version = 2

This causes the following results:

- A Mobile Web Client with version 0 of the customer revision component *cannot* download the upgrade kit until they install an upgrade that changes their customer revision number from 0 to 1.

- A Mobile Web Client with version 1 or 2 of the customer revision component *can* download the upgrade kit.

As another example, suppose you assign the following values when creating an upgrade kit for a custom report component:

Minimum Old Version = NULL  
 Maximum Old Version = NULL

Null values for both these settings indicate that subscribers who have any previous version of the component or no previous version of the component can download and use the kit. Therefore, null values for these settings should only be used if there are no prerequisite versions for the component, or if the upgrade kit will contain all prerequisites within itself.

As a third example, suppose you want to distribute two kits for the same component, such as a report and a batch file that will manipulate the report. To make sure that the report is installed before the batch file is run, you would create one kit for the report and a separate kit for the batch file, and you would make the kit for the batch file dependent on the kit for the report. The settings shown in [Table 1](#) would accomplish this objective.

Table 1. Example Settings for Upgrade Kits Where Kit #2 Depends on Kit #1

Kit Description	Minimum Old Version Value	Maximum Old Version Value	New Version Value
Report	NULL	NULL	1
Batch file to manipulate report	1	1	2

### About Specifying Acceptable Version Ranges for Upgrade Components

Siebel Anywhere also stores the acceptable range of versions for each upgrade component. For each subscriber, Siebel Anywhere checks the following:

- Which upgrade component versions are still acceptable for running the Siebel application
- Which components must be upgraded

The acceptable range of versions is defined by the Min Version and Max Version settings. Min Version specifies the earliest acceptable version for a component. A component must be upgraded if its version number is less than the value of Min Version. A component does not require upgrading if its version number falls between the values of Min Version and Max Version, inclusive.

Min Version and Max Version values are assigned automatically when you apply an upgrade kit (that is, when you update a compiled information string with the component version information, before distributing the kit). There are two ways to apply a kit. For information about applying a kit while using the Upgrade Kit Wizard to define the kit, see [Chapter 4, “Defining Upgrade Kits.”](#) For information about applying a kit by using the Apply Upgrade Kit Version Information dialog box, see [“Applying an Upgrade Kit” on page 96.](#)

Regardless of when the kit is applied, the values that are assigned to Min Version and Max Version are as follows:

- The value for Min Version defaults to the current Min Version value for the component to be upgraded. This makes the kit optional for all users who have at least that version of the component. If you prefer to make the kit required for all users, you can do so by performing either of the following actions:
  - Select the Required Upgrade Kit check box in the Upgrade Kit Wizard.
  - Click the Require Upgrade Kit button in the Apply Upgrade Kit Version Information dialog box.Either of these actions sets Min Version to match the value of New Version for the kit.
- The value for Max Version always matches the value of New Version for the kit.

As an example, suppose the following values are automatically assigned when you use the Apply Upgrade Kit Version Information dialog box for a customer revision upgrade kit:

Min Version= 2  
Max Version= 4

This causes the following results:

- The kit is required for any Mobile Web Client with version 1 of the customer revision component. If users have a version less than the minimum and choose not to install the upgrade, they can only access the application in a read-only mode.
- The kit is optional for any Mobile Web Client with version 2, 3, or 4 of the customer revision component. Users with these versions can either install the upgrade or continue using their current version of the component.

## About Storing Version Information

When an administrator applies an upgrade kit by using the Apply Upgrade Kit Version Information dialog box, the following settings for the kit are recorded as a compiled information string in the database:

- New Version
- Minimum Old Version
- Maximum Old Version
- Min Version
- Max Version



## About How Siebel Anywhere Component Versions Are Used

After Siebel Anywhere has stored an upgrade kit's version information in the database and the administrator has distributed the kit, eligible subscribers have access to the kit. Siebel Anywhere compares the subscriber's current component version with the version information in the database under conditions that depend on the type of the subscriber and certain settings. For detailed information about when these version checks are done, see [Table 2](#).

Table 2. Version Check Occurrence

Subscriber Type	When Version Check Occurs
Mobile Web Client and Developer Web Client users	Mobile and Developer Web Clients perform the version check when the client starts up and connects to either the server or local database. Mobile Web Clients also perform a version check during each synchronization session.
Regional Node Servers	Performed as the Regional Node Server starts up, dependent upon the version check flag on the Regional Node Server. The Replication Agent on a Regional Node Server also performs a version check per synchronization session dependent upon the same Version Check parameter.

Multiple factors affect what happens after Siebel Anywhere performs a version check. These factors include the subscriber type, the state the Siebel application is in when the versions are compared, and the relative numbers of the versions.

In general, if a version check reveals that a subscriber is required to upgrade, that subscriber is prompted to do so, and has limited or no access to the affected application until the upgrade is complete. For detailed information about responses to the version check process when the upgrade is required, see [Table 3](#).

However, if a version check reveals that a subscriber is *not* required to upgrade, that subscriber generally is not prompted to upgrade, but must voluntarily navigate to User Preferences > Component Upgrades to discover whether an upgrade is available and to request the upgrade. For more information about displaying optional upgrade kits and requesting optional upgrades, see [“Retrieving and Installing Upgrade Kits” on page 106](#).

Table 3. Response to Version Check Process Based on Application Status

Client or Server	Status of Application: Running	Status of Application: Startup
Mobile Web Client	<p>During synchronization, if version check discovers a required upgrade kit, a prompt will appear asking if you want to download the upgrade kit now. The choices are:</p> <p><i>Yes</i>. Upgrade will take place.</p> <p><i>No</i>. Synchronization ends.</p> <p>If you restart synchronization prior to exiting from the application, you will be prompted again.</p>	<p>If version check discovers a required upgrade at application startup, a message such as the following will appear:</p> <p>“Client is being started in read-only mode. Please start a Siebel Remote session to retrieve the upgrade kit required to upgrade your system.”</p>
Developer Web Client	Not applicable.	<p>If version check discovers a required upgrade at application startup, a prompt will appear asking if you want to download the upgrade kit now. The choices are:</p> <p><i>Yes</i>. Upgrade takes place.</p> <p><i>No</i>. User application starts up in read-only mode.</p>
Regional Node Server	<p>If Version Check is TRUE, Replication Agent will automatically download the upgrade kit and shutdown the Regional Node Server.</p> <p>If Version Check is FALSE, Replication Agent will stop merging but will not shut down the Regional Node Server.</p>	<p>If Version Check is set to TRUE, the Regional Node Server will download the upgrade kit and exit. The administrator needs to invoke the Upgrade wizard manually, from the command line, to carry out the upgrade.</p> <p>If Version Check is FALSE, the Regional Node Server will start up.</p>

## Crucial Siebel Anywhere Terminology

This section provides information about the following crucial terms for understanding and working with Siebel Anywhere:

- [Upgrade Kits](#) (required and optional)

- [Upgrade Components](#)
- [Upgrade Configurations](#)
- [Siebel Anywhere Subscribers](#)

## Upgrade Kits

A *Siebel Anywhere Upgrade Kit* is an archived file that contains software or database schema changes required to upgrade a specific upgrade component on a subscriber's computer. An upgrade kit contains one or more upgrade kit items, which are instructions for actions to be performed and the files associated with those actions. A kit also contains information about the sequence in which the actions are to be performed. Available actions include:

- Executing any type of program or batch script.
- Copying, moving, or deleting files.

Siebel Anywhere architecture supports creation of upgrade kits in a Web deployment. Administrators can use an HTML browser without any Siebel software installed on their machines to perform the Siebel Anywhere administrative tasks. Preparation of upgrade kits is done through the use of the Upgrade Kit Wizard and the Upgrade Kit Builder.

After the Siebel Anywhere Administrator creates an upgrade kit, it is automatically stored on the Siebel File System. From this location, it is available for retrieval and installation by subscribers. Files or scripts that are included in the upgrade kit are stored in compressed form. For more information about how Siebel Anywhere stores and identifies upgrade kits and the files that are included in upgrade kits, see ["Siebel Anywhere File Attachments" on page 12](#).

It is very important to test upgrade kits. It is recommended that you distribute each kit to selected mobile users through the use of a test configuration, and have those users attempt to download and install the kit before you distribute the kit to a wider group of users.

There are two types of upgrade kits for Siebel client subscribers—required and optional. Upgrade kits created for Siebel Servers should always be required.

### Required Upgrade Kits

A *required upgrade kit* is a kit that must be retrieved and installed on the intended Mobile or Developer Web Clients, or Siebel Servers. The Siebel clients and servers will prompt the user to upgrade if there is a required upgrade. If a required upgrade is not implemented, Siebel client can only be started in read-only mode, and Siebel Server will not start.

### Optional Upgrade Kits

An *optional upgrade kit* is a kit that is made available to subscribers, but subscribers are not required to retrieve and install the kit. For example, optional upgrade kits are useful for making third-party applications, files, or other noncritical components available for users to retrieve at their discretion.

Siebel Anywhere does not automatically prompt users to retrieve and install optional kits. Optional upgrade kits are manually retrieved using the Component Upgrades view, which is accessible from the Siebel User Preferences screen, and are installed using the Upgrade Wizard. The Component Upgrades view must be included in the responsibilities that are assigned to users. See [“Retrieving and Installing Upgrade Kits” on page 106](#).

**CAUTION:** It is strongly recommended that you use the optional kit feature as a method for testing every Siebel Anywhere component upgrade. When an upgrade kit is created as an optional kit, test users can retrieve kit from the Component Upgrades view whenever it is convenient to do so. If the kit is created as a required kit, test users can lose read/write access to Siebel applications unless they upgrade when automatically prompted to do so. After testing, you can make the kit required or optional.

## Upgrade Components

An *upgrade component* is a logical unit of software for which Siebel Anywhere performs version checks, to determine whether that software needs to be upgraded. The determination is made by comparing the subscriber’s existing version of the component with the version requirements specified in an upgrade kit. Each upgrade component defines how to check versions for a particular software module. For example, the upgrade component Siebel Sales CFG, which is used to check the version of the Siebel Sales Client configuration file, defines how to locate the file and how to read the version from it. For more information about how Siebel Anywhere conducts version checks, see [“How Siebel Anywhere Versions Work” on page 13](#).

Upgrade components can be thought of as belonging to two broad categories:

- *History-dependent* components depend on the presence of certain previously installed software.
- *History-independent* components can be upgraded regardless of the software previously installed.

Components are associated with subscribers by means of Siebel Anywhere configurations, which are discussed in more detail later in this section. A configuration contains one or more upgrade components. The Siebel administrator can include one or more upgrade components in a configuration.

Siebel Business Applications provide several preconfigured Upgrade Components. Seeded components are available based upon the languages installed on the database server. The Siebel administrator can also register additional Siebel Anywhere components or other third-party software. Siebel Anywhere can upgrade these components, just as it does the predefined components used by Siebel Business Applications. For example, it would be possible to create an upgrade component to run a batch program in the form of a .bat or .exe file. Each upgrade component is classified by a component type, as described in [Table 4](#).

Table 4. Predefined Examples of Siebel Anywhere Components

Siebel Component Type	Siebel Anywhere Component	Comment
Customer Revisions	Siebel Client Customer Revision Siebel Client Customer Revisions_ <i>[language-code]</i>	A nonspecific component for distributing any customer-revised files, such as Online Help files, Siebel Report files, Siebel Help Files, Siebel Sample Database Files, third-party files or applications on the Siebel client. Can be either history-dependent or history-independent.
	Siebel Server Customer Revisions Siebel Server Customer Revision_ <i>[language-code]</i>	A nonspecific component for distributing any customer-revised files on the Siebel Server. Can be either history-dependent or history-independent.
Siebel Upgrade Wizard	Siebel Upgrade Wizard	The Siebel Upgrade Wizard. It is automatically associated with every upgrade kit to make sure the client uses the current version of the wizard. History-independent.
Siebel Configuration File	Siebel Sales CFG_ <i>[language-code]</i>	A component for distributing a customized CFG file for client systems. History-independent.
Siebel Database Extensions	Siebel Database Schema	The database schema used by the Siebel Regional Node and Siebel Remote user databases. History-independent.
Siebel Executables	Siebel Client Executables Siebel Client Executables_ <i>[language-code]</i>	The Siebel Business Applications client executables. History-independent.

Table 4. Predefined Examples of Siebel Anywhere Components

Siebel Component Type	Siebel Anywhere Component	Comment
Siebel Repository File	Siebel Client Repository File_ <i>[language-code]</i>	The file used by Siebel Business Applications (.srf). History-independent.
	Siebel Server Repository File_ <i>[language-code]</i>	
Third-Party Software	Third Party - Oracle 8 Client	Third-Party Software, as used on the Siebel client. Can be either history-dependent or history-independent.
	Third Party - Microsoft Word	
	Third Party - Microsoft SQL Server Driver	Some Third-Party Software upgrade components are for version checking <i>only</i> . Siebel Anywhere is not intended to deliver complex Third Party software products as upgrade kits.
	Third Party - Microsoft Internet Explorer	
	Third Party - IBM DB2 Client	
	Third Party - Adobe Acrobat Reader	For smaller and simpler software products such as Adobe Acrobat and WinZip, you can use Siebel Anywhere to deliver them as Third-Party upgrade kits.
	Third Party - Adobe Acrobat	

For more information about upgrade components, see [“Setting Up Custom Siebel Anywhere Upgrade Components” on page 43](#).

## Upgrade Configurations

An Upgrade Configuration is a definition of the setup used by a particular set of Siebel Anywhere subscribers, such as Siebel Call Center Clients or Siebel Sales Clients. A configuration associates a particular set of Siebel Anywhere subscribers with the specific set of upgrade components that those subscribers need to have managed and maintained. Each Siebel subscriber belongs to an individual Siebel Anywhere configuration. When Siebel Anywhere checks whether a particular subscriber needs an upgrade, it checks the versions of all components included in that subscriber’s configuration.

Siebel Business Applications provide configurations such as:

- Siebel Test Client
- Siebel Call Center Client
- Siebel Service Client
- Siebel Sales Client
- Siebel Marketing Client
- Siebel Partner Manager Client
- Siebel ERM Client

Each subscriber can be assigned to one of these configurations. Administrators can also create new configurations for special situations.

**NOTE:** Siebel Anywhere supports global deployments by including seeded upgrade configurations and components for each supported language included within your Siebel Business Applications. Consider carefully the receivers or subscribers to any upgrade kit you create, and use the correct upgrade component based on end-user languages.

By default, Siebel Anywhere uses the value of the ComponentName entry in the Siebel section of an employee's CFG file to determine what configuration the employee is using and thus what components it should check. The server parameter UpgComponent specifies the configuration for Siebel Server.

It is possible to override an employee's default configuration assignment by using the Administration - Siebel Anywhere screen to assign the employee to a different configuration. This method of assigning a client to a configuration is appropriate for situations in which the assignment is temporary. For example, a client might be assigned to a specific configuration to let that client test an upgrade kit before it is released for general use. These temporary assignments are also called *dynamic assignments*.

For more information about working with configurations, see [“Modifying and Creating Siebel Anywhere Configurations” on page 38](#).

## Siebel Anywhere Subscribers

A Siebel Anywhere subscriber is a Siebel client or a Siebel application server that belongs to an active configuration. It can be a laptop, desktop, or a regional node. A Siebel Anywhere subscriber belongs to one upgrade configuration. Types of subscribers include the following:

- Mobile Web Clients of Siebel Business Applications such as Siebel Sales, Partner Manager, Siebel Field Service, or Siebel Call Center
- Developer Web Clients
- Siebel Servers operating against regional databases (referred to as Regional Node Servers)

**CAUTION:** It is strongly recommended that you run *only* the Siebel Smart Web Client for user accounts that have Siebel administrator responsibilities, to make sure that administrative tasks are performed while connected to the HQ server, and to make sure that the administrator is not prevented from logging in for reasons related to component versions. However, *if* you run the Siebel Developer Client for any administrator account, it is strongly recommended that the account *not* be associated with a Siebel Anywhere configuration. This precaution also helps prevent version-related login problems.

## Siebel Anywhere Deployment Recommendations

It is recommended that the Siebel Anywhere component group be enabled on only one Siebel Applications Server in the enterprise. To create upgrade kits, the Upgrade Kit Builder server component should also be enabled.

**CAUTION:** During the installation of the first Siebel Server in the Siebel enterprise, do not enable the Siebel Anywhere component group. Wait until every Siebel Server is installed. Then, enable the Siebel Anywhere component group on only one Siebel Server. Do not enable Siebel Anywhere on a Regional Node Server. The reason for this is that upgrade Kits should be built sequentially in the correct order. If the Upgrade Kit Builder is enabled on multiple servers, upgrade kits may be built in the incorrect order because multiple upgrade kits can be created at the same time.

For details about how to enable a Component group, see *Siebel System Administration Guide*.



# 3

## Upgrade Planning and Preliminary Tasks

This chapter describes planning and other preliminary tasks that must be completed before you use Siebel Anywhere to create an upgrade kit. It includes the following sections:

- [“Process of Planning and Preparing for Kit Creation” on page 25](#)
- [“Determining Upgrade Requirements” on page 26](#)
- [“Creating Needed Infrastructure Elements” on page 38](#)
- [“Preparing Upgrade Kit Contents” on page 48](#)
- [“Synchronizing Components” on page 59](#)

### Process of Planning and Preparing for Kit Creation

This section provides an overview of how to plan a Siebel Anywhere upgrade and complete the tasks that must precede creation of one or more upgrade kits. The planning and preparation process consists of three main parts:

- 1 [“Determining Upgrade Requirements” on page 26](#). In this part of the process, you gather information about the software to be distributed and the computers that will receive it. The information you gather determines how many components will be involved and how many upgrade kits will be needed, among other important points.
- 2 [“Creating Needed Infrastructure Elements” on page 38](#). In this part of the process, you determine whether or not existing Siebel Anywhere components and configurations match your upgrade needs, create any additional components, and create or modify any configurations you will need.
- 3 [“Preparing Upgrade Kit Contents” on page 48](#). In this part of the process, you gather (and, where necessary, create) the files that will be included in each upgrade kit. The specific process to follow depends on the type of components you are planning to upgrade.

**NOTE:** Some kits do not require any files. For example, Database Schema kits do not require any files to create a kit.

When these parts of the process have been completed, you can use the information, infrastructure elements, and files you have prepared to define the upgrade kits you need. This process is described in [Chapter 4, “Defining Upgrade Kits.”](#)

**CAUTION:** Any errors in an upgrade kit, whether in the files being distributed or in application processes defined for them, can have widespread impact on your production environment. To prevent such errors, thorough planning, preparation, and testing of upgrade kits are essential.

# Determining Upgrade Requirements

Before using Siebel Anywhere to upgrade existing software or distribute new software, you must gather a considerable amount of information about the software to be distributed and the computers that will receive it. Specifically, you must collect the following kinds of information, each of which is described in further detail in following sections.

Determining upgrade requirements is a step in [Process of Planning and Preparing for Kit Creation](#) on page 25.

- **What software needs to be replaced or added?** For detailed information, see [“Identifying Software to Be Replaced or Added”](#) on page 26.
- **Is Siebel Anywhere a suitable method for upgrading or adding this software?** For detailed information, see [“Evaluating Siebel Anywhere as an Upgrade or Delivery Method”](#) on page 27.
- **Which computers need the software?** For detailed information, see [“Identifying Computers and Users to Receive Upgrades”](#) on page 28.
- **Which configurations can deliver the necessary components to the designated computers?** For detailed information, see [“Identifying Configurations to Deliver Upgrade Components”](#) on page 28.
- **What files must each component's upgrade kit contain?** For detailed information, see [“Identifying Files to Include in Upgrade Kits”](#) on page 30.
- **Will each upgrade kit be required or optional?** For detailed information, see [“Choosing Required or Optional Upgrade Types”](#) on page 30.
- **What version settings must each component's upgrade kit have?** For detailed information, see [“Determining Version Setting Values”](#) on page 31.
- **What specific preliminary tasks must be performed for particular upgrade kit types?** For detailed information, see [“Other Preliminary Upgrade Tasks for Specific Upgrade Kit Types”](#) on page 35.
- **Do multiple components need to be delivered in a prescribed sequence?** For detailed information, see [“Determining Required Upgrade Sequences”](#) on page 37.
- **How will you test each upgrade kit before distributing it to end users?** For detailed information, see [“Planning Upgrade Test Details”](#) on page 37.

## Identifying Software to Be Replaced or Added

Siebel Anywhere can upgrade or distribute many kinds of software. The nature of the software to be upgraded or distributed affects many parts of the process of using Siebel Anywhere:

- **Upgrades of existing software.** For upgrades of existing software, you must determine the version numbers used by Siebel Anywhere for previous versions of the software being upgraded. You must also decide whether to allow any of those previous versions to remain in use, or whether to require that they be upgraded.

- **Distribution of new software.** For distribution of new software, you may need to create one or more custom components. Depending on who will use the software, you may also need to create new configurations.

For language-dependent components, planning the upgrade also includes identifying every end-user language used in your Siebel implementation. For example, a CFG file associated with a specific language is a language-dependent component, such as Siebel Sales CFG\_ENU for English, Siebel Sales CFG\_DEU for German, or Siebel Sales CFG\_JPN for Japanese. Be sure to use the correct end-user language pack (or upgrade component) for each language included in your Siebel Business Applications while creating upgrade kits. For more information regarding the use of different languages, see [“Example of Global Deployment with Siebel Anywhere” on page 119](#).

The number and type of upgrade kits required for your upgrade depend on your subscribers and the components to be upgraded. For example, you may be preparing to upgrade to a new version of your custom Siebel configuration, requiring the distribution of a new SRF file to every Mobile and Developer Web Client user.

Your custom configuration may also need to apply database schema changes to your Siebel databases and to the local databases of Mobile Web Clients. Database schema changes are not distributed by a kit to Developer Web Clients. Creating a database schema kit applies the changes from the logical schema to the physical schema. These changes are visible to Developer Web Clients, without distribution through a kit, because they connect directly to the Siebel Server Database.

You must create an upgrade kit for each component to be upgraded. In the preceding example, you would need to create one upgrade kit for the Siebel Client Repository File and a second upgrade kit for the database schema extensions.

It is a good idea to compile a list of the upgrade kits you will need.

## Evaluating Siebel Anywhere as an Upgrade or Delivery Method

Siebel Anywhere is suitable for delivering many kinds of software, including many software upgrades, but it is not suitable for delivery of some types of software and some types of upgrades. Observe the following guidelines:

- Use Siebel Anywhere *only* with software that is related to the use of Siebel Business Applications.
- Do *not* use Siebel Anywhere for major Siebel release upgrades, such as upgrading a database schema from Siebel 6 to Siebel 7.
- Do *not* use Siebel Anywhere to upgrade from Siebel 7.0.x to Siebel 7.5.x, Siebel 7.7.x, or Siebel 7.8.x. The addition of Unicode support in release 7.5 puts these upgrades outside the scope that Siebel Anywhere can handle. For information about upgrades that do not use Siebel Anywhere, see the *Siebel Database Upgrade Guide*.
- Siebel Anywhere *can* be used to upgrade Siebel Developer Web Client software and Siebel Mobile Web Client software for the following Siebel version combinations:
  - From Siebel version 7.5.x to Siebel version 7.7
  - From Siebel version 7.5.x to Siebel version 7.7 maintenance releases

- From Siebel version 7.7 to Siebel version 7.7 maintenance releases
- From Siebel version 7.5.x to Siebel version 7.8
- From Siebel version 7.5.x to Siebel version 7.8 maintenance releases
- From Siebel version 7.7.x to Siebel version 7.8
- From Siebel version 7.7.x to Siebel version 7.8 maintenance releases

## Identifying Computers and Users to Receive Upgrades

To make sure that Siebel Anywhere delivers your upgrade kit to the appropriate computers and users, you must take the following steps:

- Determine whether any Siebel Servers need the upgrade. (Steps to follow for servers and expected behavior may differ from corresponding client steps and behavior.)
- Determine which clients need the upgrade. Some possibilities include:
  - All clients need the upgrade.
  - Clients who have a particular job function need the upgrade.
  - Clients who have particular prior versions of the software need the upgrade.

## Identifying Configurations to Deliver Upgrade Components

A Siebel Anywhere *configuration* is a definition of the setup used by a particular set of Siebel Anywhere subscribers, such as Siebel Call Center Clients or Siebel Sales Clients. A configuration associates a particular set of Siebel Anywhere subscribers with the specific set of upgrade components that those subscribers need to have managed and maintained. Each Siebel subscriber belongs to an individual Siebel Anywhere configuration that includes a specific set of components.

As part of the process of planning your upgrade, you must make sure that your Siebel implementation contains configurations that are related to the components you want to upgrade. You must also make sure that those configurations can be used to distribute the upgrade to the appropriate subscribers, whether those subscribers represent servers or clients.

The following procedure provides instructions for listing existing configurations and inspecting the components to which they are related.

### ***To display available Siebel Anywhere configurations and their related components***

- 1 From the application-level menu, select Navigate > Site Map > Administration - Siebel Anywhere.
- 2 From the link bar, select Upgrade Configurations.

- 3 In the Upgrade Configurations list, select a configuration that you want to inspect, and then click the Upgrade Components view tab.

The Upgrade Components list displays the components that are currently associated with the selected configuration. Table 5 describes some of the fields in this list.

Table 5. Selected Fields in the Upgrade Components List

Field	Comments
Name	Name of the component. Only single-byte, alphanumeric characters, blank space, underscore, and dash are allowed. Component names may not include special characters like periods or other invalid characters such as slash, asterisk, pipe, question mark, colon, quotes, or angle brackets.
Min Version	Minimum version required for the component on the client's system.  If the client uses a version of the component that is less than the minimum, the client must upgrade or the application runs in read-only mode.
Max Version	Maximum version allowed for the component on the client's system.  If clients use a version of the component that is between the minimum and maximum, they can still use the application in read/write mode without installing an upgrade kit, even if the upgrade kit is required.  Clients can access the system if their local version is higher than the maximum version for the component.

**NOTE:** Siebel 7.8 does not include a Required At Startup field in the Upgrade Components list. In Siebel 7.0, version checking was done automatically only if a Required at Startup parameter was checked for the particular component in question. Beginning with Siebel 7.5, every component receives automatic version checking.

The following procedure provides instructions for listing the subscribers who are dynamically assigned to a selected configuration. The procedure does *not* list subscribers who are assigned to a particular configuration by means of a CFG file setting.

***To list employees who are dynamically associated with a configuration***

- 1 From the application-level menu, select Navigate > Site Map > Administration - Siebel Anywhere > Upgrade Configurations > Employees.
- 2 In the Upgrade Configurations list, select the configuration for which you want to view dynamically assigned employees.

The Employees list displays the employees who are dynamically assigned to the selected configuration. Employees who are assigned to the configuration through their CFG file values are not listed.

If you find that existing configurations do not meet your current upgrade needs, see [“Modifying and Creating Siebel Anywhere Configurations” on page 38](#) for information on how to modify existing configurations or create new configurations.

## Identifying Files to Include in Upgrade Kits

Most Siebel Anywhere upgrade kits contain software files to be distributed to subscribers. (The exceptions are kits that contain only instructions to be executed. Such instructions are generated during the kit creation process, rather than being stored in file attachments.) As part of planning an upgrade kit, you should identify any files the kit will include and gather those files into a single directory. [Table 6](#) lists some files that are typically included in various kit types.

**NOTE:** It is recommended that you gather files for only one upgrade kit at a time.

Table 6. Typical Files Included in Upgrade Kits

Upgrade Kit Type	Included Files
Siebel Configuration File	One CFG file per kit, such as siebel.cfg or uagent.cfg
Siebel Database Schema	No files
Siebel Client Executables	All the language-independent (base) files for a Siebel patch
Siebel Client Executables_ <i>[language-code]</i>	All files for a Siebel patch that are specifically for a given language
Siebel Client Customer Revisions	Batch files, web images
Siebel Client Customer Revisions_ <i>[language-code]</i>	Reports, message templates
Siebel Repository File	siebel.srf

## Choosing Required or Optional Upgrade Types

It is possible to define an upgrade kit as either required or optional, as described in [“Upgrade Kits” on page 19](#). However, it is recommended that you initially define *every* upgrade kit as optional. Your test clients can test an optional kit at their convenience, while keeping read/write access to the Siebel application in the meantime.

When you are satisfied that an optional upgrade kit functions appropriately for test clients, you can modify the kit to make it required, retest it as a required kit, and distribute it to production users as a required kit. For more information about this process, see [“Converting an Optional Kit to a Required Kit” on page 98](#).

## Determining Version Setting Values

The exact set of information needed for defining a Siebel Anywhere upgrade kit depends on the type of kit involved. However, you must supply version information for all upgrade kit types. When determining who can and who must upgrade a particular component, Siebel Anywhere compares version information stored in the database with the versions of components actually installed on each subscriber's computer. The information in the database comes from the information you supply when you create upgrade kits.

For each upgrade kit, you will need to supply values or approve default values for three version settings:

- New Version
- Minimum Old Version
- Maximum Old Version

For a general discussion of the significance of these versions, see [“How Siebel Anywhere Versions Work” on page 13](#). In general, the values you supply will be related to the version numbers already in use for the component being upgraded. Therefore, gathering version information about existing components is an important part of planning an upgrade.

**CAUTION:** Make certain that Minimum Old Version, Maximum Old Version, and New Version settings are correctly set before finishing your upgrade kit, using the default numbering system, if possible. Incorrectly specifying the version information can prevent subscribers from upgrading successfully.

## Gathering Information on Existing Component Versions

The following procedure provides instructions for listing the components currently available in your Siebel implementation, along with information about currently acceptable versions of those components and how those versions are checked.

**CAUTION:** When implementing major upgrades, be sure to preserve the version numbers for every component. This precaution is important because Siebel Anywhere assigns increasing version numbers, but components that are included in upgrades may have default version numbers set to zero. If the version number of a new component is left lower than the version number of the corresponding preupgrade component, newly-upgraded users may be prompted, incorrectly, to install the old component. Never reset any Siebel Anywhere version numbers to zero; instead, increase the version numbers of the new components to match their preupgrade counterparts. In particular, you must use the `srfstamp` utility to stamp a current version number on a new repository (`.srf`) file. For information about using `srfstamp`, see [“To test repository components for consistency” on page 132](#). The version number of your repository is displayed as User Version when you choose Help > About Repository from the application-level menu in your Siebel application.

### *To display version information for existing components*

- 1 From the application-level menu, select Navigate > Site Map > Administration - Siebel Anywhere.

- From the link bar, select Upgrade Component List.

The Upgrade Component List appears. [Table 7 on page 32](#) describes the information available in this list.

**CAUTION:** Do not use the Upgrade Component List to modify information about an existing component—modifying information in the list can prevent version checking from working properly. If you need to modify version information for an existing component, use the Upgrade Kit Wizard, as described in [Chapter 4, “Defining Upgrade Kits.”](#) If you want to modify other component characteristics, create a custom component with the characteristics you need, as described in [“Setting Up Custom Siebel Anywhere Upgrade Components” on page 43.](#)

Table 7. Fields in the Upgrade Component List

Name	Comment
Component Type	The type of component kit; for example, Siebel Executables or third-party software.
Locate Information	Information used by Siebel Anywhere when locating version information for the subscriber’s currently installed component. For more information about how Siebel Anywhere uses this setting, see <a href="#">“About Monitoring and Verifying Siebel Anywhere Version Numbers” on page 43.</a>
Locate Method	Method used by Siebel Anywhere to locate version information for the subscriber’s currently installed component. For more information about how Siebel Anywhere uses this setting, see <a href="#">“About Monitoring and Verifying Siebel Anywhere Version Numbers” on page 43.</a>
Max Version	The latest version of the component that is available for running the application in read/write mode.
Min Version	The earliest version of the component that is acceptable for running the application in read/write mode.
Name	Name that identifies the component for Siebel Anywhere.
Version Information	Information used by Siebel Anywhere when checking the version of the subscriber’s currently installed component. For more information about how Siebel Anywhere uses this setting, see <a href="#">“About Monitoring and Verifying Siebel Anywhere Version Numbers” on page 43.</a>
Version Method	Method used by Siebel Anywhere to check the version for the subscriber’s currently installed component. For more information about how Siebel Anywhere uses this setting, see <a href="#">“About Monitoring and Verifying Siebel Anywhere Version Numbers” on page 43.</a>

## Planning Version Settings for a New Upgrade Kit

The information you gather about current version settings lets you plan version settings for your new upgrade kit.

**NOTE:** When you run the Upgrade Kit Wizard to *define* your new upgrade kit, you will need to supply values for the New Version, Minimum Old Version, and Maximum Old Version settings.



The following tables contain guidelines for determining the version values you should use in a variety of situations.

- [Table 8 on page 33](#) provides guidelines for choosing your New Version value.
- [Table 9 on page 34](#) provides guidelines for choosing your Minimum Old Version and Maximum Old Version values.

You can use these tables as worksheets by printing them out and filling in the values that you will use when you define your upgrade kit.

The New Version value specifies the version number that the component being upgraded will have after the upgrade kit is used. Look in the Situation column of [Table 8](#) to find a description of your situation, and read the adjacent guidelines.

Table 8. New Version Planning Worksheet for Upgrade Component \_\_\_\_\_

Version Setting	Situation	Value Guidelines	Your Value
New Version  (Identifies the version number of the component in the upgrade kit)	New component	Set New Version to 1.	1
	Upgrade to existing component	Set New Version to $n+1$ , where $n$ is the Siebel Anywhere version number of the latest version of the component previously installed within your Siebel System. To display previously installed versions, see <a href="#">“To display version information for existing components” on page 31.</a> )	
	Upgrade to existing component after a major Siebel upgrade	Make sure that any components provided in the major upgrade have Siebel Anywhere version numbers equal to or greater than preupgrade components. You may need to stamp new numbers on the components to achieve this, particularly for SRF files. For more information, see <a href="#">“Testing for Upgrade Kit Component Consistency” on page 132.</a>  When component version numbering is continuous across the upgrade, use instructions in <a href="#">“Upgrade to existing component,”</a> in previous row of this table.	
	Replacing a defective kit that has been distributed and deactivated	Set New Version to the same value you assigned to the defective kit.	

The Minimum Old Version and Maximum Old Version values specify the range of component versions that subscribers must have to download and install the upgrade kit. Look in the Situation column of [Table 9](#) to find a description of your subscribers' situation, and read the adjacent guidelines for choosing values.

Table 9. Minimum Old Version and Maximum Old Version Planning Worksheet for Upgrade Component

Version Setting	Situation	Value Guidelines	Your Value
Minimum Old Version  (Determines who can use the kit)	No component versions are prerequisites for subscribers using upgrade kit or all needed prerequisites are included in this kit.	Leave blank.	NULL
	Subscribers are upgrading a history-independent component, such as SRF, executable, or database schema.	Automatically set to NULL. Cannot be changed.	NULL
	Subscribers are upgrading a CFG file.	It is recommended that you leave this setting blank.	
	Subscribers are upgrading a customer revision component.	It is recommended that you accept the default value for this setting and Maximum Old setting, to make sure that your users download and install every kit of this type, without skipping any of them.	
	Subscribers are required to have a prior version of component.	Set to earliest component version that can download and install the new kit.	
	You are replacing a defective kit that has been distributed and deactivated.	Set Minimum Old Version to the same value as you used in the defective kit.	

Table 9. Minimum Old Version and Maximum Old Version Planning Worksheet for Upgrade Component

Version Setting	Situation	Value Guidelines	Your Value
Maximum Old Version  (Determines who can use the kit)	No component versions are prerequisites for subscribers using upgrade kit or all needed prerequisites are included in this kit.	Leave blank.	NULL
	Subscribers are upgrading a history-independent component, such as SRF, executable, or database schema.	Automatically set to NULL. Cannot be changed.	NULL
	Subscribers are upgrading a CFG file.	It is recommended that you leave this setting blank.	
	Subscribers are upgrading a customer revision component.	It is recommended that you accept the default value for this setting and Minimum Old setting, to make sure that your users download and install every kit of this type, without skipping any of them.	
	Subscribers are required to have a prior version of component.	Set to latest component version that can download and install the new kit. This value is always less than the value of New Version.	
	You are replacing a defective kit that has been distributed and deactivated.	Set Maximum Old Version to the same value as you used in the defective kit.	

## Other Preliminary Upgrade Tasks for Specific Upgrade Kit Types

The following paragraphs describe additional preliminary tasks to perform for specific upgrade kit types.

### Preliminary Tasks for Siebel Database Schema Upgrade Kits

Perform the following tasks before creating a Siebel Database Schema upgrade kit:

- Make sure that the changes to your database schema can be distributed by this type of upgrade kit.

**NOTE:** Due to a technical limitation in the Sybase Adaptive Server Anywhere product, you cannot use Siebel Database Schema Upgrade Kits to add required, non-null extension columns to existing local database tables. If you make this kind of database change, you must deploy it by reextracting your Mobile Web Client users, rather than using a schema upgrade kit.

- Make sure the ODBC data source correctly points to the server database that has the modified database definition.
- Determine the values to use for the following settings. For brief descriptions of these settings, see [Table 16 on page 68](#).
  - User Name (for DBA)
  - User Password (for DBA)
  - Schema Qualifier
  - Privileged User ID
  - Privileged User Password
  - Table Space
  - Index Space
- Use SQL queries on your server database to obtain values for the following parameters specific to your database platform:
  - 16K Table Space
  - 32K Table Space
  - Table Group File
- If you are operating in a DB2 environment, drop all customized views and triggers before you run the Upgrade Kit Wizard to define a database schema upgrade kit. Otherwise, the attempt to create the kit will fail.

For more information about the overall process of deploying a database schema upgrade, see [“Performing Database Schema Updates” on page 123](#).

### Preliminary Tasks for Siebel Repository File Upgrade Kits

Perform the following tasks before creating either a Siebel Client Repository File upgrade kit or a Siebel Server Repository File upgrade kit:

- Obtain or create the new repository (.srf) file.
- Place the new repository file in a network location that is accessible to the Siebel Server where the Upgrade Kit Builder will be running.
- Make a note of the UNC path to the network location where you placed the .srf file.

## Preliminary Tasks for Third-Party Software Upgrade Kits

Perform the following tasks before creating a Third-Party upgrade kit:

- Locate or prepare the executable file or script that will be executed on the subscriber's machine when the upgrade kit is installed. Make a note of any input parameters required by the executable file or script.
- Place the executable file on the Siebel Server where the Upgrade Kit Builder is enabled, or in a network location that is accessible to that server.
- Make a note of the UNC path to the network location where you placed the executable file.
- Determine the location where the third-party software should be installed on the subscriber's machine.

## Preliminary Tasks for Siebel Customer Revisions Upgrade Kits

Perform the following tasks before creating a Siebel Customer Revisions upgrade kit:

- Locate or prepare the executable file or script that will be executed on the subscriber's machine when the upgrade kit is installed. (For example, this might be a `reset.bat` script file that resets browser security settings to the values that allow the Mobile Web Client to function properly.) Make a note of any input parameters required by the executable file or script.
- Locate any other files to be included in the upgrade kit, and make a note of the locations to which they should be installed on the subscriber's machine.

## Determining Required Upgrade Sequences

You may sometimes need to use more than one upgrade kit to complete an upgrade. For example, client software in a Siebel maintenance release or patch may consist of a base component and a language-specific component. The base component would require a Siebel Client Executable upgrade kit, and the language-specific component would require a Siebel Client Executable\_*[language-code]* upgrade kit.

In cases involving multiple upgrade kits, the kits may need to be installed in a specific sequence. (For Siebel maintenance releases, the base upgrade kit must be installed before any language-specific upgrade kit.) It is possible to automate this process by making one kit dependent on the prior installation of another kit. You can specify this dependency before activating the second kit in the sequence. For instructions on how to do this, see ["Controlling the Order of Kit Installation" on page 94](#).

**NOTE:** If you plan to specify a required upgrade sequence, do not use the Upgrade Kit wizard to activate the kits involved.

## Planning Upgrade Test Details

It is important to plan your upgrade kit testing ahead of time, to make sure that you create any needed infrastructure elements and choose appropriate settings when you create the upgrade kit.

The following general recommendations are useful for most test plans:

- Create all upgrade kits as optional kits. This facilitates testing by letting you request an upgrade at your convenience. (A required kit may or may not prompt you to upgrade when you start the application, depending on version requirements and test client version.) After you have verified that an optional upgrade kit installs correctly, you can change it to a required kit, if you wish, and retest its installation using another test client.
- Distribute a new upgrade kit to a test configuration (such as the Test Client Configuration) before distributing it to production configurations. If necessary, create such a test configuration ahead of time, and make sure that it has appropriate test employee logins assigned to it, either by means of configuration files or dynamic assignments.
- Test each upgrade kit with both Mobile Web Clients and Developer Web Clients.

## Creating Needed Infrastructure Elements

This section contains information about creating or modifying the infrastructure elements that are prerequisites for successful creation and deployment of an upgrade kit. It also contains information about verifying Siebel File System connections, which are another essential part of your infrastructure.

Creating needed infrastructure elements is a step in [Process of Planning and Preparing for Kit Creation on page 25](#).

The information in this section is divided into the following subsections:

- [“Modifying and Creating Siebel Anywhere Configurations” on page 38](#)
- [“Setting Up Custom Siebel Anywhere Upgrade Components” on page 43](#)
- [“Verifying Connections to the Siebel File System” on page 47](#)
- [“Verifying Availability of Siebel Anywhere” on page 47](#)

## Modifying and Creating Siebel Anywhere Configurations

This section contains information about modifying and creating Siebel Anywhere configurations. The information is divided into the following subsections:

- [“Adding Components to a Configuration” on page 39](#)
- [“Removing Components from a Configuration” on page 39](#)
- [“Creating a New Configuration” on page 39](#)
- [“Assigning Employees to a Configuration” on page 41](#)

**NOTE:** If a new configuration is created, a configuration is split or merged with another configuration, or the name of an existing configuration is changed, the `ComponentName` parameter on the corresponding Siebel clients must also be updated so that the clients can detect any upgrades.

## Adding Components to a Configuration

At times, you may need to add components to an existing configuration. Any custom component that you create must be added to a configuration before you can use it to distribute upgrades. The following procedure describes how to perform this task.

Adding components to a configuration is a step in [Process of Limiting Distribution of an Upgrade Kit on page 100](#).

### *To add a component to a configuration*

- 1 From the application-level menu, select **Navigate > Site Map > Administration - Siebel Anywhere**.
- 2 From the link bar, select **Upgrade Configurations**.
- 3 In the Upgrade Configurations list, select the configuration to which you want to add a component, and then click the **Upgrade Components** view tab.
- 4 In the Upgrade Components list, click **New**.
- 5 In the Upgrade Components dialog box, double-click the component you want to add.

The component appears, highlighted, in the Upgrade Components list.

Repeat [Step 3](#) through [Step 5](#) for any additional components you want to add.

## Removing Components from a Configuration

At times, you may need to remove components from an existing configuration.

### *To remove a component from a configuration*

- 1 From the application-level menu, select **Navigate > Site Map > Administration - Siebel Anywhere**.
- 2 From the link bar, select **Upgrade Configurations**.
- 3 From the Upgrade Configurations list, select the configuration from which you want to remove a component, and then click the **Upgrade Components** view tab.
- 4 In the Upgrade Components list, select the component you want to remove, and then click **Delete**.
- 5 In the dialog box, confirm that this record is the record you want to delete.

Repeat [Step 3](#) through [Step 5](#) for any additional components to be removed from this configuration.

## Creating a New Configuration

In most cases, the configurations provided with Siebel Anywhere are sufficient. However, you can create new configurations for special situations or to represent different installed clients within your company.

Creating a new configuration is a step in [Process of Limiting Distribution of an Upgrade Kit on page 100](#).

For example, if everyone in your company uses Siebel Sales and you need to distribute certain upgrades to employees in remote offices separately from the rest of the company, you would create a new configuration to accommodate this situation. The employees in the remote offices would be temporarily associated with the new configuration so that you could distribute the special upgrade just to them. (See “Assigning Employees to a Configuration” on page 41 for details on associating specific employees with a configuration.)

Also, you may need to create different configurations based upon language usage. If there is a set of the users in your company only using English (ENU) and another set using both English (ENU) and German (DEU), you should create two different configurations.

If you need to create a new configuration for long term use (as in language usage), make sure that every client CFG file is appropriately updated.

**NOTE:** The recommended method for creating a new configuration is to copy an existing configuration and modify the copy, as this method minimizes the possibility for error. The following procedure describes the use of this method. However, it is also possible to use standard Siebel application techniques to create a new record and fill in the necessary fields.

**To create a new configuration**

- 1 From the application-level menu, select Navigate > Site Map > Administration - Siebel Anywhere.
- 2 From the link bar, click Upgrade Configurations.
- 3 In the Upgrade Configurations list, select an existing configuration record that resembles the configuration you want to create, and then click the menu button and select Copy Record.

A highlighted copy of the record that you selected appears.

- 4 Modify the fields in the copied record to meet your requirements.

Some fields are described in the following table.

**CAUTION:** Limit the Name field to 91 characters or less. Exceeding the 91-character limit in the Name field causes the synchronization process to fail.

Field	Comment
Name	The name of the configuration.
Active	When this check box is selected, it indicates that the configuration is to be version-checked.  When you create a new configuration by copying an existing configuration, this check box is cleared by default. Be sure to select it before using the new configuration.
Component Type	This value is either client or server configuration.
Comments	Comments can be entered here.



- 5 Add appropriate Upgrade Components to the new configuration you just created.

To do this, see ["To add a component to a configuration" on page 39](#).

## Assigning Employees to a Configuration

The purpose of assigning employees to configurations is to handle uncommon situations, such as having one particular employee retrieve and test a kit. In most cases, testing is done using special test accounts and test configurations, and regular employees are associated with a specific configuration based on a ComponentName entry in the CFG file of an application.

Assigning employees to a configuration is a step in [Process of Limiting Distribution of an Upgrade Kit on page 100](#).

For example, in the Siebel Sales application the ComponentName in the default CFG file (siebel.cfg) would be Siebel Sales Client by default. This means that everyone using the Siebel Sales application is automatically associated with the Siebel Sales configuration.

Assigning employees to a configuration in the Employees list in the Upgrade Configurations view can be used in place of the entry in the CFG file, or as a method for overriding the entry in the CFG file.

**CAUTION:** It is strongly recommended that you run *only* the Siebel Smart Web Client for user accounts that have Siebel administrator responsibilities, to make sure that administrative tasks are performed while connected to the HQ server, and to make sure that the administrator is not prevented from logging in for reasons related to component versions. However, *if* you run the Siebel Developer Web Client for any administrator account, it is strongly recommended that the account *not* be associated with a Siebel Anywhere configuration. This precaution also helps prevent version-related login problems.

When associating employees to configurations, keep in mind the following recommendations:

- Do not dynamically assign the System Administrator to a configuration.
- Do not dynamically assign Siebel Tools users to a configuration.

Siebel Anywhere *will not* allow an Employee to be dynamically associated with more than one configuration. For example, if the employee JSMITH is dynamically associated with Configuration A and the Siebel administrator then associates JSMITH with Configuration B, JSMITH will automatically be disassociated from Configuration A and only dynamically associated with Configuration B.

**CAUTION:** If an employee is accidentally associated with an incorrect configuration, that employee is disassociated from the correct configuration, and may not have access to upgrades that are distributed to the correct configuration. For this reason, it is best to correct the configuration assignment without delay.

If an employee will use more than one installation of the Siebel client in the same Siebel software implementation, do not associate that employee with a configuration. For example, if the Employee JSMITH chooses to install and use both Siebel Call Center and Siebel Sales, then that employee should not be dynamically associated with a Siebel Anywhere configuration. The reason is the upgrade kit for one configuration will likely be different than the upgrade kit for another configuration.

In this example, the CFG file would be different for the Siebel Call Center and Siebel Sales configurations. If JSMITH were dynamically associated to the Call Center configuration and logged on using the Siebel Sales client, the version check would only detect that there was a new upgrade kit and would not differentiate between the two installations of the Siebel client.

If the login is specified under an Employees view, then Siebel Anywhere does not check the CFG file any further and only checks the version of the component listed under the associated configuration for that client. Typically, Siebel Anywhere groups users by the product they use, for example Call Center or Sales Client, which is determined by the CFG file the user uses.

**NOTE:** The first time a Mobile Web Client connects to the Remote Server to initialize the local database, Siebel Anywhere does a version check for the client using the Upgrade Configuration specified in the CFG file. Siebel Anywhere cannot detect a dynamic configuration assignment at this time. However, after the Mobile Web Client successfully initializes the local database, Siebel Anywhere can detect any dynamic configuration assignment for that client.

### *To assign an employee to a configuration*

- 1 From the application-level menu, select [Navigate > Site Map > Administration - Siebel Anywhere > Upgrade Configurations](#).
- 2 In the Upgrade Configurations list, select the configuration to which you want to add an employee, and then click the Employees view tab.
- 3 In the Employees list, click New.
- 4 In the Login field of the new record, click the select button.

From the Pick Employee dialog box, select the employee you want to add, and click OK.

Repeat [Step 2](#) through [Step 4](#) to assign additional employees to this configuration.

**NOTE:** When it initializes the local database for the first time, Siebel Anywhere uses the configuration in the CFG file. It does not use the configuration defined in the Employees view under the [Administration - Siebel Anywhere Screen](#).

## Removing Employees from a Configuration

Use the following procedure to end an employee's dynamic configuration assignment, which returns that user to the original configuration assignment defined in the user's CFG file.

Removing employees from a configuration is a step in [Process of Limiting Distribution of an Upgrade Kit on page 100](#).

### *To remove an employee from a configuration*

- 1 From the application-level menu, select [Navigate > Site Map > Administration - Siebel Anywhere > Upgrade Configurations](#).
- 2 In the Upgrade Configurations list, select the configuration from which you want to remove an employee, and then click the Employees view tab.

- 3 In the Employees list, select the record for the employee you want to remove from the configuration, and then click the menu button and select Delete Record.

## Setting Up Custom Siebel Anywhere Upgrade Components

This section contains information about creating custom components. Siebel Business Applications provide many predefined upgrade components, but you can create additional components if you need them. For example, it would be possible to create an upgrade component to run a batch program in the form of a .bat or .exe file.

Siebel Anywhere can upgrade custom components, just as it does the predefined components provided with Siebel Business Applications. For correct operation, however, you must define how Siebel Anywhere will monitor and verify the version number for any custom component. The following paragraphs describe how Siebel Anywhere monitors and verifies version numbers, followed by instructions for creating a custom component:

- [“About Monitoring and Verifying Siebel Anywhere Version Numbers”](#)
- [“Creating a Custom Component” on page 46](#)

### About Monitoring and Verifying Siebel Anywhere Version Numbers

Siebel Anywhere uses information from the Locate Method, Locate Information, Version Method, and Version Information settings to determine how it needs to check the version of a component. Depending on the component, Siebel Anywhere uses some or all of these settings.

For information about when a version check occurs for each type of subscriber, see [Table 2 on page 17](#).

The following subsections describe how the Locate Method and Version Method settings are used, along with the Locate Information and Version Information settings. An example follows.

### Locate Method

The purpose of the Locate Method setting is to build a path to a file or other location, such as a registry. This value is then passed to the Version Method to read version information. [Table 10](#) describes some available values for Locate Method, each of which specifies a different way to locate version information.

**NOTE:** For some Locate Method values, you will need to specify additional information in the Location Information setting.

Table 10. Locate Method Values

Value	Comment
Siebel Root/ Siebel Home	Determines location based on the client's Siebel root directory. Locate Information is ignored in this case. (For example, this method would be used for Siebel Upgrade Wizard components.)  Both server and client use this to retrieve the value of Siebel Root.
Registry	Determines location based on information in the Microsoft Windows registry. Locate Information would contain the path to read in the registry. (For example, this method would be used for Third Party - Microsoft Word components.)
Environment	Determines the location on the client by reading an environment variable. Locate Information would contain the name of the environment variable to read.
CCF	Determines location on server from a set of common core facility (CCF) parameters. Locate Information would contain the value from the CCF program context. (For example, this method would be used for a Siebel Server Customer Revision component.)
CFG	Determines location based on reading the client's active CFG file. Locate Information would contain the section and entry to read from the CFG file, plus an optional value to append to the value read from the CFG file. (For example, this method would be used for a Siebel Client Customer Revision component.)

### Version Method

This setting provides additional information that is used in conjunction with Locate Method to read the version of a file. [Table 11](#) describes the available values for Version Method, each of which specifies a different way to determine the version of the component.

**NOTE:** Some Version Method values require you to specify additional information in the Version Information setting.

Table 11. Version Method Values

Value	Comment
File Version	Determines version based on reading the version of a file. Version Information would contain the name of the file to version check.
Registry	Determines version based on reading the version from the registry. Version Information would contain the path to read in the registry.
DLL	Determines the version based on executing an entry point in a DLL file. Version Information would contain the name of the DLL file and the name of the entry point. For example, bin\sslcver.dll, VersionString executes the function VersionString in the file bin\sslcver.dll.
Database	Determines the version based on reading the Siebel Database Schema. Version Information is ignored in this case.
CFG	Determines version based on reading the client's active CFG file. Version Information would contain the section and entry to read from the CFG file, plus an optional value to append to the value read from the CFG file. (For example, this method would be used for a Siebel Client Customer Revision component.)
SRF	Determines the version based on reading the SRF. Version Information would contain the section name and the entry name from the active CFG file to be read to determine the name of the SRF.

### Example: How Siebel Anywhere Uses Locate and Version Methods

The following example shows how Siebel Anywhere checks component versions, using the Locate Method and Version Method settings.

[Table 12](#) shows information that appears in the Upgrade Components list for the Siebel Client Customer Revisions\_ENU component, along with comments explaining how each setting is used.

Table 12. Example of Locate and Version Method Values

Field Name	Value	Comments
Name	Siebel Client Customer Revisions_ENU	Name of the component.
Locate Method	CFG	Information about the location of component version information is stored in the client's active CFG file.

Table 12. Example of Locate and Version Method Values

Field Name	Value	Comments
Locate Information	Siebel,ClientRootDir,bin\enu\clntrev.cfg	Information about the location of component version information is stored in the ClientRootDir entry in the [Siebel] section of the client's CFG file. To find the actual version information, append \bin\enu\clntrev.cfg to the value in the ClientRootDir entry.  For example, if the ClientRootDir entry is c:\siebel, the path to the version information would be c:\siebel\bin\enu\clntrev.cfg.
Version Method	CFG	Version information is stored in the CFG file specified by the Locate Method and Locate Information settings.
Version Information	Siebel,Version	Version information is stored in the Version entry in the [Siebel] section of the specified file.

## Creating a Custom Component

Siebel Anywhere functionality allows you to define your own custom components, if necessary.

The Siebel administrator *must define* how Siebel Anywhere will monitor and verify the version number for a custom upgrade component. Siebel Anywhere maintains an individual version number for each upgrade component. The version number must be maintained on a Siebel Anywhere subscriber to compare it to the version number in the database. The Siebel administrator can maintain the version number in the application, in a file, or in the Windows registry.

**CAUTION:** It is strongly recommended that the Siebel administrator use the Copy Record functionality to copy existing upgrade components and simply rename the new upgrade component so that the correct parameters are used for the new upgrade component. This method is described in the following procedure. Make sure that you also update the Locate and Version columns appropriately.

### To create a custom component

- 1 From the application-level menu, select Navigate > Site Map > Administration - Siebel Anywhere.
- 2 From the link bar, click Upgrade Component List.
- 3 In the Upgrade Components list, select an existing component that resembles the component you want to create, and then click the menu button and select Copy Record.

A highlighted copy of the record that you selected appears.

- 4 Modify the fields in the copied record to meet your requirements.

Use a meaningful name to describe the function of the component. Limit the Name field to 91 characters or less.

**CAUTION:** Exceeding the 91-character limit in the Name field causes the synchronization process to fail.

Make sure the Min Version and Max Version fields are blank. This information is filled in when the kits are applied.

- 5 For any configuration that you will upgrade using your new custom component, you must specify that the component is a related component for the configuration. For instructions, see [“Adding Components to a Configuration” on page 39](#).

## Verifying Connections to the Siebel File System

The Siebel File System on the server stores the newly created upgrade kits. It is important to verify settings for connections to the Siebel File System. The connection settings vary depending upon the type of client. In either case these are essential to downloading the kits.

Mobile Web Clients connect to the Siebel File System through the Siebel Server. Therefore, the Siebel File System parameter in Synchronization Manager should be set to the same location as the File System parameter in the CFG file of the user who created the kit.

Developer Web Clients connect directly to the Siebel File System. Therefore, the File System parameter in the ServerDataSrc of the client's CFG file should be set to the appropriate location. Also, the user should have read/write access to that directory. Test for this by having the user open an attachment, and if that is successful, then the connection to File System is set correctly.

## Verifying Availability of Siebel Anywhere

The Siebel Anywhere component group must be available to allow the creation of upgrade kits. The following procedure describes how to verify the availability of this component group.

### *To check the availability of Siebel Anywhere*

- 1 From the application-level menu, select Navigate > Site Map > Administration - Server Management.
- 2 In the link bar, select Servers.
- 3 In the Servers list, select a Server that you want to inspect.
- 4 Click the Component Groups view tab, and select the Siebel Anywhere component group.
- 5 Verify that State is set to Online.

If State is not Online, see *Siebel System Administration Guide* for information about administering component groups, including instructions for enabling a component group.

# Preparing Upgrade Kit Contents

Preparing to create an upgrade kit involves gathering the software, files, or database schema updates required for the kit. It can also involve preparing simple batch scripts. The specific process to follow depends on the type of components you are planning to upgrade.

Preparing upgrade kit contents is a step in [“Process of Planning and Preparing for Kit Creation” on page 25](#).

**NOTE:** Do not use the upgrade kit type Customer Revisions for upgrade kits containing a Siebel SRF or CFG file. Instead, use the Siebel Client Repository File and Siebel Client CFG kit types, respectively, for these upgrades. This will allow proper version checking.

The following sections describe specific steps to take for certain upgrade kit types:

- [“Preparing Contents for a Siebel Upgrade Wizard Upgrade Kit” on page 48](#)
- [“Preparing Contents for a Siebel Executables Upgrade Kit” on page 49](#)
- [“Preparing Contents for Repository File or Third-Party Upgrade Kits” on page 52](#)

## Preparing Contents for a Siebel Upgrade Wizard Upgrade Kit

You must create a Siebel Upgrade Wizard upgrade kit whenever there are changes to the Siebel Upgrade Wizard in a software release from Oracle. For information about whether a specific release contains such changes, see the release-specific documentation for the release.

Beginning with version 7.7, a Siebel Upgrade Wizard upgrade kit must contain a set of DLL files, along with the Siebel Upgrade Wizard executable file, siebupg.exe. The following procedure describes how to prepare the contents for such a kit.

### *To prepare the contents for a Siebel Upgrade Wizard upgrade kit*

- 1 Create a new directory to hold the Upgrade Wizard executable and related DLLs.  
Make a note of the directory name, as you will need to specify it while defining the upgrade kit.
- 2 If you have not already done so, install the client software for the new Siebel software release on a test machine or an administrator's machine, and make a note of where you installed it.  
**NOTE:** In order for the necessary files to be accessible, the client software must be installed. The necessary files are *not* directly accessible from the network image.
- 3 Locate the client BIN directory for the new Siebel software release.  
The default location is c:\program files\siebel\web client\bin.
- 4 Copy the following files from the release client BIN directory into the new directory you created in [Step 1](#):
  - siebupg.exe
  - libarm.dll



- mfc70u.dll
- msvc70.dll
- msucr70.dll
- pw32a.dll
- srlcver.dll
- sslcacln.dll
- sslccore.dll
- sslcdb.dll
- sslcevt.dll
- sslclm.dll
- sslcnapi.dll
- sslcns.dll
- sslcos.dll
- sslcosa.dll
- sslcosd.dll
- sslcrsa.dll
- sslcsc.dll
- sslcscr.dll
- sslcsecm.dll
- sslcshar.dll
- sslcsnns.dll
- sslcsnsc.dll
- sslcsrd.dll
- sslcsrvr.dll
- sslcsym.dll
- sslcupg.dll
- sslcver.dll

## Preparing Contents for a Siebel Executables Upgrade Kit

You can use the Siebel Executables component type to deliver Siebel maintenance releases (patches) or additional language packs to Mobile or Developer Web Clients.

Siebel maintenance releases consist of a set of language-independent files (called *base* files) and one or more sets of language-specific files. Base files and language-specific files require different upgrade kits:

- Siebel Client Executable upgrade kits are used to distribute base files.
- Siebel Client Executable\_*[language-code]* upgrade kits are used to distribute language-specific files.

**NOTE:** The base upgrade kit must be installed before language-specific kits are installed. If you wish, you can make the language-specific kits technically dependent on the base kit, so this will be accomplished automatically. For more information see [“Controlling the Order of Kit Installation” on page 94](#).

The following procedure describes how to prepare files for inclusion in either base or language-specific client executable upgrade kits.

### *To prepare files for inclusion in Siebel Client Executable upgrade kits*

- 1 Create a new, network-accessible UNC directory to use for storing the base client executable files in the maintenance release or patch.

A typical name for this directory could be `\\machine_name\seaw\base`.

- 2 For each language in your Siebel deployment, create a separate new UNC directory.

These language-specific directories should *not* be subdirectories of the directory you created in [Step 1](#). Typical language-specific directories could include `\\machine_name\seaw\enu` for United States English or `\\machine_name\seaw\fra` for French.

- 3 Locate the network image for the Siebel Maintenance Release or patch.
- 4 Navigate to the Maintenance Release or patch subdirectory that contains the base installer.exe file for Siebel Windows clients.

A typical location for these files in the network image file hierarchy would be `Windows\Client\Siebel_Web_Client`.

- 5 Copy all of the files (but not the subdirectories) from the directory that you located in [Step 4](#) to the directory that you created in [Step 1](#).

- 6 Pick a language that is used in your Siebel deployment, and navigate to the Maintenance Release or patch directory that contains the language-specific files for that language.

For example, files specifically for use with United States English may be located in `Windows\Client\Siebel_Web_Client\enu`.

- 7 Copy all the files from the directory that you located in [Step 6](#) to the corresponding directory that you created in [Step 2](#).
- 8 Repeat [Step 6](#) and [Step 7](#) for each additional language that is used in your Siebel deployment.

### Related Topics

[“About Reducing Siebel Client Executables Kit Size” on page 51](#)

[“Defining a Siebel Client Executables Upgrade Kit” on page 70](#)

[“Controlling the Order of Kit Installation” on page 94](#)

[“Distributing a Siebel Maintenance Release or Patch” on page 117](#)

[“Distributing Additional Language Packs” on page 118](#)

## About Reducing Siebel Client Executables Kit Size

This topic describes a feature called the Delta Install kit type for a Siebel Client Executables upgrade kit. This feature is supported for versions 7.7.2 and later. Information about the Delta Install kit type does not apply for earlier releases.

Beginning with Siebel version 7.7.2, it is possible to reduce the size of some Siebel Client Executables upgrade kits, which also reduces the time it takes users to retrieve and install these kits. This reduction in upgrade kit size is accomplished by creating the *Delta Install* type of Siebel Client Executables kit. Delta Install kits contain condensed information about specific *changes* to client executable files, instead of an actual set of the new executable files.

Prior to version 7.7.2, all Siebel Client Executables kits contained sets of new executables files. This kit type is now called a *Standard Install* kit. When you create a Siebel Client Executables kit, you choose whether to create it as a Delta Install kit or a Standard Install kit.

**NOTE:** Delta Install kits are suitable for delivering Siebel Maintenance Releases, including fixpack releases, that contain relatively few changes to client executable files. For example, you could use a Delta Install kit to upgrade client executables from version 7.7.1 to version 7.7.2 or from version 7.7.2 to version 7.7.2.1. Delta Install kits are *not* suitable for major upgrades, such as upgrades from version 7.7 to version 7.8.

You can create Delta Install kits for the Siebel Client Executables base (language-independent) component and for each language-specific Siebel Client Executables\_*[language-code]* component that is for a language that your users already have installed.

**CAUTION:** If you want to add an additional language for your users, do not use a Delta Install kit to add that language. Instead, use a Standard Install kit. That is, create an upgrade kit for the Siebel Client Executables\_*[language-code]* component that has Standard Install as the Type of Installation for the kit. Delta Install kits are not supported for the addition of a new language. If a Delta Install kit is incorrectly used to install a new language, it can damage some client executable files.

Before defining a Delta Install kit, an administrator must use the Siebel Patch Utility (siebpatch.exe) to create a *delta patch file*. This is a file that contains information about the changes that were made to client executables between two specific Siebel releases.

Users retrieve and install Delta Install kits just as if the kits contained complete sets of client executable files. When a Delta Install kit is installed, the Upgrade Wizard applies the changes that are described in the kit to the user’s existing client executable files, rather than replacing the user’s files.

Depending on the release you are installing, users may need to upgrade the Upgrade Wizard before installing a Delta Install Kit. In this case, an administrator must define an Upgrade Wizard upgrade kit, and make any Delta Install kits dependent on that Upgrade Wizard kit. For information about whether this requirement applies to you, see the release-specific documentation for the release you want to install.

For information about the process of creating Delta Install upgrade kits, see [“Process of Creating a Delta Install Siebel Client Executables Upgrade Kit” on page 52.](#)

## Preparing Contents for Repository File or Third-Party Upgrade Kits

To prepare for creating an upgrade kit for a Siebel repository file, third-party application, or other non-Siebel file, copy the files to a directory on the Siebel Server from which you will run the Upgrade Kit Builder server component, or to a directory that is accessible from that server.

**NOTE:** Beginning with Siebel release 7.7, Siebel Anywhere supports two kinds of Repository File upgrade kits: *standard* repository file upgrade kits, which contain complete SRF files, and *delta* repository file upgrade kits, which contain only changes to a Siebel Repository File. For more information about how to prepare files for delta repository file kits, see [“Process of Creating a Delta Repository File Upgrade Kit” on page 55.](#))

Siebel Anywhere can install third-party applications or other files using one of the following methods:

- Executing a batch script or installation program.
- Copying files to a specified location.

Before deploying a kit through Siebel Anywhere, make sure that any files, scripts, or other programs required to install third-party applications contain the necessary data and are fully tested.

## Process of Creating a Delta Install Siebel Client Executables Upgrade Kit

Beginning with version 7.7.2, Siebel Anywhere supports the use of the Delta Install type of Siebel Client Executables upgrade kit. For information about this kit type, see [“About Reducing Siebel Client Executables Kit Size” on page 51.](#)

To create a Delta Install client executables upgrade kit, perform the following tasks:

- 1 Consult the release-specific documentation for the release you want to install, to determine whether you must define an Upgrade Wizard upgrade kit. If so, prepare and define the Upgrade Wizard upgrade kit as described in the following topics:
  - [“Preparing Contents for a Siebel Upgrade Wizard Upgrade Kit” on page 48](#)
  - [“Defining a Siebel Upgrade Wizard Upgrade Kit” on page 62](#)
- 2 Identify the machine where you will create the delta patch file. This is typically an administrator's machine. Install the version of the Siebel Client software that your mobile users *currently* use, if that version of the software is not already present.
- 3 On the same machine, but in a different directory, install the client software for the new release.

**NOTE:** The client environment is sufficient for creating the delta patch file; no servers are required at this time.

- 4 Create a delta patch file for each of the following components, as described in [“Creating a Delta Patch File” on page 53](#):
  - a Siebel Client Executables base (language-independent) component
  - b Each Siebel Client Executables\_*[language-code]* component that is in use in your Siebel implementation
- 5 Define a Siebel Client Executables upgrade kit for each of the components listed in [Step 4](#), choosing Delta Install for the Select Type of Installation setting in the Upgrade Kit Wizard.

For information about defining a Siebel Client Executables upgrade kit, see [“Defining a Siebel Client Executables Upgrade Kit” on page 70](#).
- 6 If you defined an Upgrade Wizard upgrade kit, make the Siebel Client Executables base upgrade kit dependent on the Upgrade Wizard upgrade kit, and make each Siebel Client Executables\_*[language-code]* upgrade kit dependent on the base kit.

For information about making one kit dependent upon another, see [“Controlling the Order of Kit Installation” on page 94](#).

## Creating a Delta Patch File

The following procedure describes how to create a delta patch file for use in a Delta Install type of Siebel Client Executables upgrade kit. A delta patch file contains information about how client executables have changed between two specific versions of Siebel software.

For example, you could create a delta patch file that contains information about how client executables changed between version 7.7.1 and version 7.7.2 of the Siebel software. A Delta Install type of Siebel Client Executables upgrade kit that contains that delta patch file would be used to upgrade version 7.7.1 client executables to version 7.7.2.

This task is a step in [“Process of Creating a Delta Install Siebel Client Executables Upgrade Kit” on page 52](#).

For general information about Delta Install upgrade kits, see [“About Reducing Siebel Client Executables Kit Size” on page 51](#).

### ***To create a delta patch file for use in a Delta Install Siebel Client Executables kit***

- 1 On a machine that has client software for the new release already installed, use a text editor to inspect the siebdp.rtc file.

This file is typically located in the BIN subdirectory for the client software. For example, if the client software is installed in d:\seaw, then the siebdp.rtc file would be located in d:\seaw\bin.

- 2 If your Siebel implementation contains files or directories that you do not want inspected for changes when the delta patch file is created, add those files or directories to the siebdp.rtc file, and then save the file.

The correct location for your additions is after the files and directories that are already listed in the following section of the siebdp.rtc file:

```
IGNORE *.log, *.err, *.cfg, *.file
IGNORE *_setup.reg
IGNORE Uninst*.bat
```

```
IGNOREDIR _uninst
IGNOREDIR UPGRADE
IGNOREDIR packager
IGNOREDIR temp
IGNOREDIR local
IGNOREDIR log
```

- ; You may customize the following section to include/exclude folder and files.
- ; To ignore file, use: IGNORE \$filename
- ; To ignore directory, use: IGNOREDIR \$directoryname

**CAUTION:** Do not change entries in the earlier part of the siebdp.rtc file. Changing earlier entries can prevent correct creation of the delta patch file.

If you want to add a comment to the file, place the comment on its own line, and use a semi-colon (;) as the first character of the line.

- 3 Open a Command Prompt window and navigate to the directory that contains the latest installed client software, including siebdp.rtc and siebpatch.exe, for example, d:\seaw\bin.
- 4 In the Command Prompt window, enter the following command, substituting the appropriate path to the directory that contains the siebdp.rtc file:

```
si ebpatch /c d:\seaw\bin\siebdp.rtc
```

The Siebel Patch Utility window appears.

- 5 Use the information in the following table to respond to the prompts in the Siebel Patch Utility, and then click Next to start the creation of the delta patch file and display progress indicators.

Siebel Patch Utility Field or Prompt	Comments
Location of the current Siebel client	Enter the path to the directory where the older version of the Siebel client software is installed (the version that your users have now).
Location of the new Siebel client	Enter the path to the directory where the newer version of Siebel client software is installed (the version to which your users will upgrade).

Siebel Patch Utility Field or Prompt	Comments
Language	<p>If you are creating the delta patch file for a language-independent Siebel Client Executables upgrade kit, select BASE.</p> <p>If you are creating the delta patch file for a language-specific Siebel Client Executables_<i>[language-code]</i> upgrade kit, select the appropriate language code.</p>
Output File	<p>Specify a path and filename for the delta patch file you are creating. (This is the file that will be included in the upgrade kit.) Use .rtp as the file name extension for this file. If you wish, you can click the Browse button to navigate to the directory where you want the file to be placed.</p> <p><b>NOTE:</b> If you specify <i>filename.rtp</i> as the name for the delta patch file, both <i>filename.rtp</i> and <i>filename.rtd</i> are automatically created. The .rtd file is a text file that provides information about the files that have changed between the two specified versions of the software, along with information about any compression of those files that occurred when <i>filename.rtp</i> was created.</p>

- When the Siebel Patch Utility displays a message that the output package has been generated successfully, click Finish to exit from the utility.

Repeat this entire procedure as needed to create a separate delta patch file for use in each of the following upgrade kit types:

- Delta Install Siebel Client Executables (language-independent base kit)
- Delta Install Siebel Client Executables\_*[language-code]* (for each language that is used in your Siebel implementation)

## Process of Creating a Delta Repository File Upgrade Kit

Beginning with version 7.7, Siebel Anywhere supports the use of *delta* repository file upgrade kits, which contain only repository file changes, rather than an entire SRF file. The use of delta repository file upgrade kits decreases the amount of time needed for distributing repository changes to Siebel clients.

**NOTE:** Siebel does not currently support the use of delta repository file upgrade kits to upgrade servers.

To create a delta repository file upgrade kit, perform the following tasks:

- 1 Create a complete new repository file (SRF) that contains the changes you want to distribute. For general information about compiling repository files, see *Using Siebel Tools*. For specific instructions on how to compile an SRF in order to minimize the size of a delta repository file upgrade kit, see [“Compiling an SRF for Use in a Delta Repository File” on page 56](#).
- 2 Use the *diffsrf* utility to compare the new repository file with your existing repository file and to create a delta repository file that contains only the changes between the two versions. For information about how to use the *diffsrf* utility, see [“Preparing a Delta Repository File” on page 57](#).

**NOTE:** The existing repository file that you specify must have the same version as the repositories that will be upgraded using your new delta repository file kit.

- 3 Define the repository file upgrade kit as described in [“Defining a Siebel Repository File Upgrade Kit” on page 76](#). Specify that you are creating a delta kit, not a standard kit.

**CAUTION:** The “delta” (difference) between two SRF files is specific to the two SRF files being compared. Therefore, it is very important that the base (reference) SRF that is used to compile the new SRF and to generate the delta SRF be exactly the same as the SRF on any user machine on which the delta SRF will be applied. It is strongly recommended that administrator maintain a client installation similar to that of the end user. The administrator should apply all the upgrade kits to this client just as end users do. The SRF file on this client should be used as the base whenever a delta SRF needs to be created and distributed. If, instead, you specify a base reference SRF that has not actually been distributed to a client using a Siebel Anywhere upgrade kit, then the process of applying the delta file to client machines will fail in some cases.

## Compiling an SRF for Use in a Delta Repository File

The following procedure briefly describes how to compile a repository file that will be used to create a delta repository file upgrade kit. A repository file that is compiled using this procedure minimizes both the size of the resulting delta repository file upgrade kit and the time required to upgrade. For more general information about compiling repository files, see *Using Siebel Tools*.

This task is a step in [“Process of Creating a Delta Repository File Upgrade Kit” on page 55](#).

**NOTE:** Minimizing the size of the delta repository file upgrade kit depends on specifying a Reference SRF when the SRF that contains changes is compiled, as described in the following procedure. If the SRF that contains changes is compiled without specifying a Reference SRF, the SRF that contains changes can still be used to create a delta repository file upgrade kit, but the kit size is not minimized.

### *To compile a repository file for use in creating a delta repository file upgrade kit*

- 1 In Siebel Tools, make the repository changes you want to distribute.
- 2 From the Tools menu, select Compile Projects.
- 3 In the Projects pane of the Object Compiler dialog box, click All Projects.
- 4 Click the Reference SRF button.



- 5 In the Reference SRF dialog box, enter the path and file name of the repository file that served as the base upon which you made your repository changes, and then click OK.

For the base, use a repository file that has actually been distributed to a client machine using Siebel Anywhere. (Specifying a file that has not passed through the distribution process can cause failures during attempts to apply a delta repository file to clients.)

The delta repository file upgrade kit will contain only the differences between this base version and the new SRF you are about to compile.

- 6 In the Siebel repository file field of the Object Compiler dialog box, enter the path and file name for the repository file that you are about to compile.
- 7 Click Compile.

## Preparing a Delta Repository File

This topic describes how to use the *diffsrf* utility to create a delta repository file containing only the differences between two complete repository files. This task is a step in [“Process of Creating a Delta Repository File Upgrade Kit” on page 55](#).

**NOTE:** Every delta repository file contains information that is required for version control. This version information requires approximately 1 to 2 MB of space in the delta repository file, regardless of the size of the base SRF or the changes you have made.

### *To create a delta repository file using the *diffsrf* utility*

- 1 Note the location and name of the complete Siebel repository file (SRF) that is currently in use in your Siebel implementation.

**CAUTION:** This should be the location of a file that has been distributed to an actual client machine using Siebel Anywhere. If necessary, copy the file from a client machine to the machine where you will create the delta patch file. Do not use a repository file that has not passed through the process of being distributed by Siebel Anywhere. In some cases, a delta patch file that is created from an undistributed repository will fail to apply correctly on client machines.

- 2 Note the location and name of the complete Siebel repository file (SRF) that contains the changes you want to distribute.
- 3 On the machine that you will use to run *diffsrf*, open a Command Prompt window and navigate to the appropriate directory:
  - On your Siebel Server machine, navigate to the `siebsrvr\bin` directory for your Siebel implementation.
  - On a Siebel Developer Client machine, navigate to the `web client\bin` directory for that machine.
- 4 In the Command Prompt window, enter the following command, substituting actual values for the italicized argument placeholders shown here.

For information about the arguments for the *diffsrf* command, see [Table 13 on page 58](#).

```
diffsrf -F diff -O oldsrfr -N newsrfr -P del ta -L language
```

**NOTE:** For best results, close other applications before running `diffsrf`. The `diffsrf` utility is CPU-intensive, and may take a few moments to complete its operations.

Table 13. Arguments for the `diffsrf.exe` utility

Argument	Comments
-F <i>function</i>	<p>Function that you want the utility to perform. Valid values include:</p> <ul style="list-style-type: none"> <li>■ <code>apply</code>. Applies the changes that are recorded in delta repository file <i>delta</i> to existing repository file <i>oldsrfr</i>. Normally, this operation is not required, as repository file changes are applied by using a delta repository file upgrade kit, rather than by using <code>diffsrf</code>.</li> <li>■ <code>diff</code>. (Default) Creates delta repository file <i>delta</i>, based on differences between repository files <i>oldsrfr</i> and <i>newsrfr</i>. This delta repository file can be included in a delta repository file upgrade kit.</li> </ul>
-O <i>oldsrfr</i>	<p>Complete path and filename for an existing repository file.</p> <ul style="list-style-type: none"> <li>■ If you use the -F <code>apply</code> argument to apply changes in a delta repository file to an existing repository file, <i>oldsrfr</i> should be the existing repository file.</li> <li>■ If you use the -F <code>diff</code> argument to create a delta repository file, <i>oldsrfr</i> should be the SRF that is currently used in your Siebel implementation. Use an SRF that has actually been distributed to a client machine using Siebel Anywhere. (Specifying an SRF that has not passed through the distribution process can cause failures during attempts to apply the delta repository file to clients.)</li> </ul>
-N <i>newsrfr</i>	<p>Complete path and filename for a repository file that contains changes.</p> <ul style="list-style-type: none"> <li>■ If you use the -F <code>apply</code> argument to apply changes in a delta repository file to an existing repository file, <i>newsrfr</i> is the name of the new repository file to create.</li> <li>■ If you use the -F <code>diff</code> argument to create a delta repository file, <i>newsrfr</i> should be the SRF that contains the changes that you want to distribute.</li> </ul>
-P <i>delta</i>	<p>Complete path and filename for a delta repository file.</p> <ul style="list-style-type: none"> <li>■ If you use the -F <code>apply</code> argument to apply changes in a delta repository file to an existing repository file, <i>delta</i> is the name of the delta repository file containing the changes to apply.</li> <li>■ If you use the -F <code>diff</code> argument to create a delta repository file, <i>delta</i> is the name of the delta repository file for <code>diffsrf</code> to create. Be sure to specify a location for which you have write permission.</li> </ul>
-L <i>language</i>	<p>Language code for <i>oldsrfr</i>, <i>newsrfr</i> and <i>delta</i> repository files, such as ENU for English, DEU for German, and JPN for Japanese.</p>

# Synchronizing Components

Before you create your first upgrade kit, you must synchronize server components. If synchronization has not been done, attempts to define upgrade kits fail.

## *To synchronize server components before defining upgrade kits*

- 1 From the Siebel application menu, choose Navigate > Site Map > Administration - Server Configuration > Enterprises > Component Groups.
- 2 In the Component Groups list, verify that the Siebel Anywhere component group is enabled.
- 3 Click the Component Definitions tab.
- 4 In the Component Definitions list, use standard query techniques to locate records that have Component Group set to Siebel Anywhere, and click Enable for any that are not currently active.
- 5 Click Synchronize.
- 6 Restart the Siebel Server where the Siebel Anywhere components are enabled.



# 4

## Defining Upgrade Kits

*Defining* an upgrade kit is the process of running the Upgrade Kit Wizard and entering data concerning a single Siebel Anywhere component that is to be upgraded. After all necessary data has been provided, the Upgrade Kit Wizard invokes the Upgrade Kit Builder, which uses the gathered data to build the upgrade kit.

**NOTE:** It is important to gather the data to be entered before you run the Upgrade Kit Wizard. For information on planning your upgrade kit and gathering the necessary data, see [Chapter 3, “Upgrade Planning and Preliminary Tasks.”](#) For general information about the Upgrade Kit Wizard and the Upgrade Kit Builder, see [“Siebel Anywhere Wizards and Utilities”](#) on page 11.

This chapter provides general instructions for running the Upgrade Kit Wizard, followed by specific instructions for defining several types of upgrade kits, and instructions for viewing upgrade kit properties, after the kits have been defined:

- [“Running the Upgrade Kit Wizard”](#) on page 62
- [“Defining a Siebel Upgrade Wizard Upgrade Kit”](#) on page 62
- [“Defining a Siebel Configuration File \(CFG\) Upgrade Kit”](#) on page 64
- [“Defining a Siebel Database Schema Upgrade Kit”](#) on page 66
- [“Defining a Siebel Client Executables Upgrade Kit”](#) on page 70
- [“Defining a Siebel Repository File Upgrade Kit”](#) on page 76
- [“Defining a Third-Party Software Upgrade Kit”](#) on page 80
- [“Defining a Siebel Customer Revisions Upgrade Kit”](#) on page 84
- [“Defining a Custom Component Upgrade Kit”](#) on page 88
- [“Viewing Upgrade Kit Properties”](#) on page 88

For language-specific kits, use the instructions for the appropriate kit type. For example, use instructions for defining a CFG kit to define a language-specific CFG kit.

**NOTE:** If your upgrade includes multiple components, use the instructions in this chapter to create an upgrade kit for each component.

After you define your upgrade kit, you must proceed to activate, apply, and distribute it. For information on these tasks, see [Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”](#)

Defining upgrade kits is a step in [Process of Limiting Distribution of an Upgrade Kit](#) on page 100.

## Running the Upgrade Kit Wizard

All upgrade kits must be defined using the Upgrade Kit Wizard. This section provides general instructions for running the Upgrade Kit Wizard. For background information about the Upgrade Kit Wizard, see [“The Upgrade Kit Wizard” on page 11](#).

**NOTE:** The following procedure describes how to start the Upgrade Kit Wizard for any type of upgrade kit. Subsequent steps and information required vary depending on the type of kit.

### *To start the Upgrade Kit Wizard*

- 1 From the application-level menu, select **Navigate > Site Map > Administration - Siebel Anywhere**.
- 2 From the link bar, click **Upgrade Kits**.  
The Upgrade Kits view appears.
- 3 Click **Auto Create** to start the Upgrade Kit Wizard.  
The Upgrade Kit Wizard - Upgrade Component page appears.
- 4 Follow the prompts to specify all parameters necessary to define your upgrade kit.  
The following sections of this chapter provide instructions for individual upgrade kit types. For a list of these sections, see [Chapter 4, “Defining Upgrade Kits.”](#)

## Defining a Siebel Upgrade Wizard Upgrade Kit

This section provides specific instructions for defining a Siebel Upgrade Wizard upgrade kit using the Upgrade Kit Wizard. Kits of this type are used to upgrade the Siebel Upgrade Wizard as part of software releases from Oracle.

**CAUTION:** When a software release from Oracle includes an upgrade to the Siebel Upgrade Wizard, upgrade tasks must be performed in the following order to accomplish the upgrade correctly: First, upgrade the Siebel Server and the Gateway Name Server. If these servers are not upgraded first, version numbering for subsequent upgrade tasks may be incorrect. After upgrading the Siebel Server, complete the upgrade to the Upgrade Wizard for each client before you upgrade any other software components for that client. This allows the latest version of the Upgrade Wizard to be used to accomplish the upgrades of other components.

### *To define a Siebel Upgrade Wizard Upgrade Kit*

- 1 Gather the information and files you will need, as described in [Chapter 3, “Upgrade Planning and Preliminary Tasks.”](#)
- 2 Start the Upgrade Kit Wizard using the instructions in [“Running the Upgrade Kit Wizard” on page 62](#).

- 3 Use the information you have gathered and the information in [Table 14 on page 63](#) to respond to the prompts in the Upgrade Kit Wizard.

**NOTE:** The Minimum Old Version and Maximum Old Version settings are automatically set to NULL for history-independent component types, including Siebel Upgrade Wizard, indicating that there are no prerequisite versions required for using the kit. The value for New Version is also automatically supplied for Siebel Upgrade Wizard upgrade kits.

- 4 When you have finished specifying data about the upgrade kit you are defining, click Finish to pass the request to the Upgrade Kit Builder and to exit from the Upgrade Kit Wizard.

A new row for the new upgrade kit appears in the Upgrade Kits list, with Status set to Request Submitted. For more information about the kit information available in this list and other lists, see [“Viewing Upgrade Kit Properties” on page 88](#).

After you define your upgrade kit, you must proceed to activate, apply, and distribute it. For information on these tasks, see [Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”](#)

Information in [Table 14](#) is presented in approximately the sequence used by the Upgrade Kit Wizard.

Table 14. Upgrade Kit Wizard Elements for a Siebel Upgrade Wizard Upgrade Kit

Wizard Element	Element Type	Comments
Upgrade Component	Drop-down list	Name of the component the upgrade kit will install or upgrade. Choose Siebel Upgrade Wizard for this kit type.
UNC Path for File Folder	Text field	Universal Naming Convention (UNC) path to the directory that contains the siebugp.exe file and its accompanying DLL files for the upgrade. The path should identify a network-accessible location.
New Version	Text field	Version number that the component being upgraded will have after the upgrade kit is installed. Default value is 1 greater than the current version for the component to be upgraded. For more information about choosing version values, see <a href="#">“Determining Version Setting Values” on page 31</a> .
Activate Upgrade Kit	Check box	When this check box is selected, the files to be included in the upgrade kit will be compressed into a single archive on the Siebel File System automatically. Activation can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a>  For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation” on page 94</a> .

Table 14. Upgrade Kit Wizard Elements for a Siebel Upgrade Wizard Upgrade Kit

Wizard Element	Element Type	Comments
Upgrade Kit Title	Text field	Identifier for the upgrade kit. Defaults to Upgrade Component Name value followed by a space and the New Version value, but can be modified during the kit definition process. After the kit is defined, appears in the Name field of the Upgrade Kits list.
Description	Text field	Available for comments about the upgrade kit. Comments entered here are displayed in the Upgrade Kits list.

## Defining a Siebel Configuration File (CFG) Upgrade Kit

This section provides specific instructions for defining a Siebel Configuration File upgrade kit using the Upgrade Kit Wizard. Kits of this type are used to upgrade Siebel CFG files, and are always language-specific.

**NOTE:** In Siebel releases 7.0 and release 7.5, in implementations where some clients used different installation directories than others, it was necessary to use a `cfgedit.exe` executable to change certain lines in CFG files, rather than distributing one CFG file to all users. Beginning with release 7.7, the `cfgedit.exe` executable is no longer needed. A single CFG file can be distributed to all users, regardless of differences in installation directories. The Upgrade Wizard will detect these differences and make the appropriate modifications automatically.

### *To define a Siebel Configuration File upgrade kit*

- 1 Gather the information you will need, as described in [Chapter 3, "Upgrade Planning and Preliminary Tasks."](#)
- 2 Start the Upgrade Kit Wizard using the instructions in ["Running the Upgrade Kit Wizard" on page 62.](#)
- 3 Use the information you have gathered and the information in [Table 15 on page 65](#) to respond to the prompts in the Upgrade Kit Wizard.
- 4 When you have finished specifying data about the upgrade kit you are defining, click Finish to pass the request to the Upgrade Kit Builder and to exit from the Upgrade Kit Wizard.

A new row for the new upgrade kit appears in the Upgrade Kits list, with Status set to Request Submitted. For more information about the kit information available in this list and other lists, see ["Viewing Upgrade Kit Properties" on page 88.](#)

After you define your upgrade kit, you must proceed to activate, apply, and distribute it. For information on these tasks, see [Chapter 5, "Activating, Applying, and Distributing Upgrade Kits."](#)



Information in [Table 15](#) is presented in approximately the sequence used by the Upgrade Kit Wizard.

Table 15. Upgrade Kit Wizard Elements for a Siebel Configuration File Upgrade Kit

Wizard Element	Element Type	Comments
Upgrade Component	Drop-down list	Name of the component the upgrade kit will install or upgrade. Typical values for configuration files include Siebel Sales CFG_ENU and Siebel Call Center CFG_ENU.
Files to Add	Text field	Name of the CFG file to include in the upgrade kit. This field is populated either by entering a file name and path, or by clicking Browse and choosing a file from the directory listings displayed.
Browse	Button	Displays a standard dialog box for browsing and choosing a file.
Minimum Old Version	Text field	Earliest component version that can download and install the upgrade kit. When this field is blank, indicates that there are no prerequisite versions required for using the kit. Defaults to the current component version recorded in the database. For more information about choosing version values, see <a href="#">“Determining Version Setting Values”</a> on page 31.
Maximum Old Version	Text field	Latest component version that can download and install the upgrade kit. When this field is blank, indicates that there are no prerequisite versions required for using the kit. For more information about choosing version values, see <a href="#">“Determining Version Setting Values”</a> on page 31.
New Version	Text field	Version number that the component being upgraded will have after the upgrade kit is installed. Default value is 1 greater than the current version for the component to be upgraded. For more information about choosing version values, see <a href="#">“Determining Version Setting Values”</a> on page 31.
Activate Upgrade Kit	Check box	<p>When this check box is selected, the files to be included in the upgrade kit will be compressed into a single archive on the Siebel File System automatically. Activation can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a></p> <p>For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation”</a> on page 94.</p>

Table 15. Upgrade Kit Wizard Elements for a Siebel Configuration File Upgrade Kit

Wizard Element	Element Type	Comments
Apply Versions	Check box	<p>Displayed only if Activate Upgrade Kit check box was selected in an earlier Upgrade Kit Wizard screen. When this check box is selected, the compiled information string in the database will be updated automatically with the component version information for this upgrade kit. Applying versions can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a></p> <p>For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation” on page 94.</a></p>
Required Upgrade Kit	Check box	<p>When this check box is selected, the upgrade kit will be required regardless of previous versions installed. (Min Version and Max Version are set equal to New Version, but this change is not visible in the Upgrade Kit Wizard.) Displayed only if Apply Versions check box was selected in an earlier Upgrade Kit Wizard screen.</p>
Upgrade Kit Title	Text field	<p>Identifier for the upgrade kit. Defaults to Upgrade Component Name value followed by a space and the New Version value, but can be modified during the kit definition process. After the kit is defined, appears in the Name field of the Upgrade Kits list.</p>
Comments	Text field	<p>Available for comments about the upgrade kit. Comments entered here are displayed in the Upgrade Kits list.</p>

## Defining a Siebel Database Schema Upgrade Kit

This section provides specific instructions for defining a Siebel Database Schema upgrade kit using the Upgrade Kit Wizard.

Installing the schema upgrade kit synchronizes the logical and physical schemas on Mobile Web Client and Regional Node Server databases.

**CAUTION:** In the DB2 environment before creating database schema upgrade kits, you must drop all customized views and triggers. Otherwise, the upgrade kit will fail.

### *To define a Siebel Database Schema upgrade kit*

- 1 Gather the information you will need, as described in [Chapter 3, “Upgrade Planning and Preliminary Tasks.”](#)

- 2 Start the Upgrade Kit Wizard using the instructions in [“Running the Upgrade Kit Wizard” on page 62](#).
- 3 Use the information you have gathered and the information in [Table 16 on page 68](#) to respond to the prompts in the Upgrade Kit Wizard.
- 4 When you have finished specifying data about the upgrade kit you are defining, click Finish to pass the request to the Upgrade Kit Builder and to exit from the Upgrade Kit Wizard.

A new row for the new upgrade kit appears in the Upgrade Kits list, with Status set to Request Submitted. For more information about the kit information available in this list and other lists, see [“Viewing Upgrade Kit Properties” on page 88](#).

After you define your upgrade kit, you must proceed to activate, apply, and distribute it. For information on these tasks, see [Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”](#)

**NOTE:** Be sure to check the log file after Upgrade Kit Builder creates a database schema kit—make sure that error messages or warnings, if any, are nonfatal.

Information in Table 16 is presented in approximately the sequence used by the Upgrade Kit Wizard.

Table 16. Upgrade Kit Wizard Elements for a Siebel Database Schema Upgrade Kit

Wizard Element	Element Type	Comments
Upgrade Component	Drop-down list	Name of the component the upgrade kit will install or upgrade. Select Siebel Database Schema.
ODBC Data Source	Text field	Name of the ODBC data source used to connect to the HQ database.
User Name	Text field	Siebel administrator login ID used to connect to the database.
User Password	Text field	Siebel administrator password used to connect to the database.
Schema Qualifier	Text field	For DB2/390 and AS/400 environments only, the name used to qualify all database objects created that are required by Siebel Business Applications.
Privileged User ID	Text field	Required. User account that has the necessary database authority and privileges to perform operations required to implement Siebel, including creating, accessing, and modifying Siebel database objects and native database objects. These environments have rigid controls on user identification—accounts must correspond to a real person. For a database other than DB2/390 or AS/400, this account is the same account as table owner.
Privileged User Password	Text field	Privileged User's password on the Regional Database. For a database other than DB2/390 or AS/400, this password is the table owner's password.
Table Space	Text field	Database table space or segment for Siebel tables. Table spaces are created during initial installation or by the DBA. Obtain this value from your DBA or through a SQL query on the server database. Not required for Oracle or MSSQL.
Index Space	Text field	Database table space or segment for Siebel indexes. Obtain this value from your DBA through a SQL query on the server database. Not required for Oracle or MSSQL.
16K Table Space	Text field	Obtain the optional parameter from your DBA or through a SQL query on the server database. This setting is specific to the DB2 database platform.
32K Table Space	Text field	Obtain the optional parameter from your DBA or through a SQL query on the server database. This setting is specific to the DB2 database platform.
Table Group File	Text field	Obtain the optional parameter from your DBA or through a SQL query on the server database. This setting is specific to the DB2/390 database platform.

Table 16. Upgrade Kit Wizard Elements for a Siebel Database Schema Upgrade Kit

Wizard Element	Element Type	Comments
Minimum Old Version	Read-only text field	Earliest component version that can download and install the upgrade kit. Automatically set to NULL for history-independent component types, including database schemas, indicating that there are no prerequisite versions required for using the kit. For more information about choosing version values, see <a href="#">“Determining Version Setting Values” on page 31.</a>
Maximum Old Version	Read-only text field	Latest component version that can download and install the upgrade kit. Automatically set to NULL for history-independent component types, including database schemas, indicating that there are no prerequisite versions required for using the kit. For more information about choosing version values, see <a href="#">“Determining Version Setting Values” on page 31.</a>
New Version	Text field	Version number that the component being upgraded will have after the upgrade kit is installed. Default value is 1 greater than the current version for the component to be upgraded. For more information about choosing version values, see <a href="#">“Determining Version Setting Values” on page 31.</a>
Activate Upgrade Kit	Check box	<p>When this check box is selected, the information to be included in the upgrade kit will be compressed into a single archive on the Siebel File System automatically. Activation can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a></p> <p>For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation” on page 94.</a></p>
Apply Versions	Check box	<p>Displayed only if Activate Upgrade Kit check box was selected in an earlier Upgrade Kit Wizard screen. When this check box is selected, the compiled information string in the database will be updated automatically with the component version information for this upgrade kit. Applying versions can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a></p> <p>For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation” on page 94.</a></p>

Table 16. Upgrade Kit Wizard Elements for a Siebel Database Schema Upgrade Kit

Wizard Element	Element Type	Comments
Required Upgrade Kit	Check box	When this check box is selected, the upgrade kit will be required regardless of previous versions installed. (Min Version and Max Version are set equal to New Version, but this change is not visible in the Upgrade Kit Wizard.) Displayed only if Apply Versions check box was selected in an earlier Upgrade Kit Wizard screen.  Database Schema upgrade kits should be required kits when they are distributed to users in a production environment. However, it is recommended that you create and test the kit as an optional kit before making it required. For information about changing an optional kit to a required kit, see <a href="#">“Converting an Optional Kit to a Required Kit” on page 98</a> .
Upgrade Kit Title	Text field	Identifier for the upgrade kit. Defaults to Upgrade Component Name value followed by a space and the New Version value, but can be modified during the kit definition process. After the kit is defined, this identifier appears in the Name field of the Upgrade Kits list.
Comments	Text field	Available for comments about the upgrade kit. Comments entered here are displayed in the Upgrade Kits list.

**NOTE:** To avoid unnecessary download operations, it is recommended that you keep only one upgrade kit for history-independent components, including database schema kits. Depending on your preference, you can either delete or inactivate previous kits for a history-independent component. Deleting a kit increases available space in the Siebel File System. Inactivating a kit prevents use of the kit, while keeping it available in case you need it unexpectedly.

For detailed information about using a database schema upgrade kit, see [“Performing Database Schema Updates” on page 123](#).

## Defining a Siebel Client Executables Upgrade Kit

This section provides specific instructions for defining a Siebel Client Executables upgrade kit using the Upgrade Kit Wizard. Instructions in this section apply to the following two component types:

- **Siebel Client Executables.** Used for language-independent portions of Siebel maintenance releases, known as *base* portions.
- **Siebel Client Executables\_*[language-code]*.** Used for language-specific portions of Siebel maintenance releases.

Instructions are included for defining either the Standard Install or the Delta Install type of Client Executable Upgrade Kit. For more information on Delta Install kits, see [“About Reducing Siebel Client Executables Kit Size” on page 51](#).

This task is a step in [“Process of Creating a Delta Install Siebel Client Executables Upgrade Kit”](#) on page 52.

The Delta Install type of Siebel Client Executables upgrade kit is not available for Siebel releases prior to 7.7.2. (Siebel Client Executable upgrade kits that are created with these earlier releases are equivalent to the Standard Install kit type for 7.7.2 and later.)

**NOTE:** If you are upgrading Siebel client executables from version 7.5.x to version 7.7.x or version 7.8.x, you must create a special Siebel Client Customer Revisions\_*[language-code]* kit to be used prior to your Siebel Client Executables and Siebel Client Executables\_*[language-code]* kits. If you are upgrading from version 7.7 to a 7.7 maintenance release, or from version 7.8 to a 7.8 maintenance release, or from version 7.7.x to version 7.8.x, you do not need such a kit. For information about creating the special Siebel Client Customer Revisions\_*[language-code]* kit, see *Siebel Database Upgrade Guide*. For information about making the executables kits dependent on the Siebel Client Customer Revisions\_*[language-code]* kit, see [“Controlling the Order of Kit Installation”](#) on page 94.

Each of these component types needs its own kit. For example, if you want to install a Siebel maintenance release on a Siebel system that uses both English and German, you would need at least three upgrade kits, one each for Siebel Client Executables, Siebel Client Executables\_ENU, and Siebel Client Executables\_DEU. The base kit must be installed before the language-specific kits. For instructions on enforcing this by making the language-specific kits dependent on the base kit, see [“Controlling the Order of Kit Installation”](#) on page 94. For a general overview of the process of deploying a Siebel maintenance release, see [“Distributing a Siebel Maintenance Release or Patch”](#) on page 117.

The following procedure describes how to define either a Siebel Client Executables kit for the base component, or a Siebel Client Executables\_*[language-code]* kit for a language-specific component.

#### ***To define a Siebel Client Executables upgrade kit (base or language-specific)***

- 1 Gather the information you will need, as described in [Chapter 3, “Upgrade Planning and Preliminary Tasks.”](#)
- 2 Start the Upgrade Kit Wizard using the instructions in [“Running the Upgrade Kit Wizard”](#) on page 62.
- 3 Use the information you have gathered and the information in [Table 17 on page 73](#) to respond to the prompts in the Upgrade Kit Wizard.

**NOTE:** The Minimum Old Version and Maximum Old Version settings are automatically set to NULL for history-independent component types, including Siebel Client Executables, indicating that there are no prerequisite versions required for using the kit. The value for New Version is also automatically supplied for Siebel Client Executables upgrade kits. These settings are displayed only in the final screen of the Upgrade Kit Wizard.

- 4 When you have finished specifying data about the upgrade kit you are defining, click Finish to pass the request to the Upgrade Kit Builder and to exit from the Upgrade Kit Wizard.

A new row for the new upgrade kit appears in the Upgrade Kits list, with Status set to Request Submitted. For more information about the kit information available in this list and other lists, see [“Viewing Upgrade Kit Properties” on page 88](#).

After you define your upgrade kit, you must proceed to activate, apply, and distribute it. For information on these tasks, see [Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”](#)



Information in Table 17 is presented in approximately the sequence used by the Upgrade Kit Wizard.

Table 17. Upgrade Kit Wizard Elements for a Siebel Client Executables Upgrade Kit

Wizard Element	Element Type	Comments
Upgrade Component	Drop-down list	<p>Name of the component the upgrade kit will install or upgrade:</p> <ul style="list-style-type: none"> <li>■ For the language-independent or <i>base</i> part of a Siebel maintenance release or patch for clients, select Siebel Client Executable.</li> <li>■ For the language-specific part of a Siebel maintenance release or patch for clients, select Siebel Client Executable_<i>[language-code]</i>, such as Siebel Client Executable_ENU.</li> </ul>
Select Type of Installation	Radio button	<p>Determines the size and nature of the upgrade kit.</p> <ul style="list-style-type: none"> <li>■ If you want the kit to contain a complete set of the applicable client executable files, select Standard Install.</li> <li>■ If you want the kit to contain condensed information about <i>only</i> the executable files that have changed in a particular maintenance release, select Delta Install.</li> </ul> <p>For more information about the Delta Install kit type, see <a href="#">“About Reducing Siebel Client Executables Kit Size” on page 51</a> and <a href="#">“Process of Creating a Delta Install Siebel Client Executables Upgrade Kit” on page 52</a>.</p> <p><b>NOTE:</b> The Select Type of Installation wizard element is not available for Siebel releases prior to 7.7.2. Siebel Client Executable upgrade kits that are created with earlier releases must always contain a complete set of client executable files.</p>
UNC Path for Master Installation	Text field	<p>Universal Naming Convention (UNC) path to the directory that contains the files that you prepared for the upgrade, as described in <a href="#">“Preparing Contents for a Siebel Executables Upgrade Kit” on page 49</a>. This wizard element is displayed only if Standard Install was chosen in an earlier wizard screen. The path should identify a network-accessible location.</p> <ul style="list-style-type: none"> <li>■ For the base component of a Siebel maintenance release or patch, a typical path would be \\<i>machine</i>\seaw\base.</li> <li>■ For a language-specific component of a Siebel maintenance release or patch, a typical path would be \\<i>machine</i>\seaw\ENU.</li> <li>■ For a full release in which Siebel Packager has been used to create an installation package, you must choose the directory where the package was created.</li> </ul>

Table 17. Upgrade Kit Wizard Elements for a Siebel Client Executables Upgrade Kit

Wizard Element	Element Type	Comments
UNC Path for Siebel Delta Patch File	Text field	<p>Universal Naming Convention (UNC) path to the .rtp delta patch file that was created by using the Siebel Patch Utility (siebpatch.exe). Specify the complete directory path <i>and</i> file name. This setting is displayed only if Delta Install was chosen in an earlier wizard screen.</p> <p><b>NOTE:</b> The UNC Path for Siebel Delta Patch File wizard element is not available for Siebel releases prior to 7.7.2. Use the UNC Path for Master Installation element, instead.</p>
Destination Directory	Text field	Specifies where the files of the upgrade kit are to be placed on the client machine.
Delete destination file when done	Check box	<p>When this check box is selected, the upgrade kit files will be deleted, automatically, after the kit is installed.</p> <p>If you are defining this kit for a Siebel Client maintenance release, be sure to clear this check box.</p>
Uninstall previous Siebel version	Check box	<p>When this check box is selected, the Upgrade Wizard will automatically uninstall <i>all</i> existing Siebel applications before proceeding with installation of the upgrade.</p> <ul style="list-style-type: none"> <li>■ If you are creating a kit for a Siebel Maintenance Release or patch, be sure to clear this check box.</li> <li>■ If you are creating a kit for a complete Siebel Release (such as a kit to upgrade release 7.7 client executables to release 7.8), select this check box.</li> </ul> <p><b>NOTE:</b> Siebel Maintenance Releases (patches) are designed to be applied successively, without the need for previous patches to be uninstalled. However, it can occasionally be useful for testing or troubleshooting to uninstall a patch. For instructions on uninstalling a patch <i>without</i> uninstalling the underlying base release, see <a href="#">“Uninstalling Siebel Maintenance Releases or Patches” on page 115</a>.</p>
File to execute	Read-only text field	Specifies one file in the upgrade kit to be executed as part of installation of the kit. Automatically set to install.exe for this component type, indicating that install.exe will be run from the directory identified in UNC Path for Master Installation.
Enter command line arguments	Text field	<p>Specifies any command line arguments to use when running the install.exe file.</p> <p>If you are defining this kit for a Siebel Client maintenance release, do <i>not</i> change the default command line arguments supplied in this field.</p>
Major Version	Text field	Enter a value from the first part of the release number. For example, for maintenance release version 7.5.2.211, set Major Version to 7.

Table 17. Upgrade Kit Wizard Elements for a Siebel Client Executables Upgrade Kit

Wizard Element	Element Type	Comments
Minor Version	Text field	Enter a value from the second part of the release number. For example, for maintenance release version 7.5.2.211, set Minor Version to 5.
Maintenance Version	Text field	Enter a value from the third part of the release number. For example, for maintenance release version 7.5.2.211, set Maintenance Version to 2.
Patch Version	Text field	Enter a value from the fourth part of the release number. For example, for maintenance release version 7.5.2.211, set Patch Version to 211.
Vertical Code	Drop-down list	If you are using industry-specific Siebel applications, such as Siebel Consumer Sector or Siebel Financial Services, choose SIA from the drop-down list. For all other Siebel Business Applications, leave the field blank.
Activate Upgrade Kit	Check box	<p>It is recommended that you clear this check box for all Siebel Client Executables upgrade kits.</p> <p>If this check box is selected, the files to be included in the upgrade kit will be compressed into a single archive on the Siebel File System automatically. Activation can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a></p> <p>For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation” on page 94.</a></p>
Apply Versions	Check box	<p>It is recommended that you clear this check box for all Siebel Client Executables upgrade kits.</p> <p>Displayed only if Activate Upgrade Kit check box was selected in an earlier Upgrade Kit Wizard screen. When this check box is selected, the compiled information string in the database will be updated automatically with the component version information for this upgrade kit. Applying versions can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a></p> <p>For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation” on page 94.</a></p>

Table 17. Upgrade Kit Wizard Elements for a Siebel Client Executables Upgrade Kit

Wizard Element	Element Type	Comments
Required Upgrade Kit	Check box	When this check box is selected, the upgrade kit will be required regardless of previous versions installed. (Min Version and Max Version are set equal to New Version, but this change is not visible in the Upgrade Kit Wizard.) Displayed only if Apply Versions check box was selected in an earlier Upgrade Kit Wizard screen.
Upgrade Kit Title	Text field	Identifier for the upgrade kit. Defaults to Upgrade Component Name value followed by a space and the New Version value, but can be modified during the kit definition process. After the kit is defined, this identifier appears in the Name field of the Upgrade Kits list.
Comments	Text field	Available for comments about the upgrade kit. Comments entered here are displayed in the Upgrade Kits list.

**NOTE:** To avoid unnecessary download operations, it is recommended that you keep only one upgrade kit for history-independent components, including Siebel Client Executables kits. Depending on your preference, you can either delete or inactivate previous kits for a history-independent component. Deleting a kit increases available space in the Siebel File System. Inactivating a kit prevents use of the kit, while keeping it available in case you need it unexpectedly.

## Defining a Siebel Repository File Upgrade Kit

This section provides specific instructions for defining a Siebel Repository File upgrade kit using the Upgrade Kit Wizard. These instructions can be used for the following types of upgrade components, which are always language-specific:

- Siebel Client Repository File\_*[language-code]*
- Siebel Server Repository File\_*[language-code]*

For either of these component types, two different types of repository file kits can be created:

- **Standard Repository File Upgrade Kit.** Contains a complete SRF repository file.
- **Delta Repository File Upgrade Kit.** Contains information about changes to a specific version of a Siebel repository file, rather than the complete SRF. The delta repository file must be created ahead of time, using the Siebel Delta SRF Utility (diffsrf).

**NOTE:** The upgrade kit also sets the version of the SRF in the Siebel Repository File itself. For both Standard Repository File upgrade kits and Delta Repository File upgrade kits, the kit sets the version in the final Siebel Repository File. When a Delta Repository File upgrade kit is used, the final file that receives the version number is created when the Upgrade Wizard applies the delta file to the base file.

***To define a Siebel Repository File upgrade kit***

- 1 Gather the information you will need, as described in [Chapter 3, "Upgrade Planning and Preliminary Tasks."](#)

In particular, if you are defining a Delta Repository File upgrade kit, see ["Process of Creating a Delta Repository File Upgrade Kit" on page 55.](#)

- 2 Start the Upgrade Kit Wizard using the instructions in ["Running the Upgrade Kit Wizard" on page 62.](#)
- 3 Use the information you have gathered and the information in [Table 18 on page 78](#) to respond to the prompts in the Upgrade Kit Wizard.
- 4 When you have finished specifying data about the upgrade kit you are defining, click Finish to pass the request to the Upgrade Kit Builder and to exit from the Upgrade Kit Wizard.

A new row for the new upgrade kit appears in the Upgrade Kits list, with Status set to Request Submitted. For more information about the kit information available in this list and other lists, see ["Viewing Upgrade Kit Properties" on page 88.](#)

After you define your upgrade kit, you must proceed to activate, apply, and distribute it. For information on these tasks, see [Chapter 5, "Activating, Applying, and Distributing Upgrade Kits."](#)

Information in [Table 18](#) is presented in approximately the sequence used by the Upgrade Kit Wizard.

Table 18. Upgrade Kit Wizard Elements for a Siebel Repository File Upgrade Kit

Wizard Element	Element Type	Comments
Upgrade Component	Drop-down list	<p>Name of the component the upgrade kit will install or upgrade. Select one of the following:</p> <ul style="list-style-type: none"> <li>■ Siebel Client Repository File_<i>[language-code]</i> (use when distributing repository file to clients)</li> <li>■ Siebel Server Repository File_<i>[language-code]</i> (use when distributing repository file to servers)</li> </ul>
Select File Type	Radio button	<p>Specifies whether the kit will contain a complete repository file or a “delta” repository file. Select one of the following values:</p> <ul style="list-style-type: none"> <li>■ <b>Standard Repository File.</b> Use this file type when defining a repository upgrade kit for clients if the kit will contain a complete repository file. For a repository upgrade kit for a server, always choose this file type.</li> <li>■ <b>Delta Repository File.</b> Use this file type when defining an upgrade kit for clients if the kit will contain a “delta” repository file. A delta repository file contains information about changes to a specific version of a complete repository file, rather than the complete file, itself. The delta repository file must be created ahead of time, using the Siebel Delta SRF Utility (diffsrf). For information about creating the delta repository file, see <a href="#">“Process of Creating a Delta Repository File Upgrade Kit” on page 55</a>.</li> </ul>
Files to Add	Text field	Universal Naming Convention (UNC) path and file name of the delta .srf file to include in a Delta Repository File upgrade kit. Browsing is available by clicking the Browse button described in the following row of this table.
Browse	Button	Displays a standard dialog box for browsing and choosing a delta .srf file. Available for Delta Repository File upgrade kits only.
UNC Path for SRF file	Text field	Universal Naming Convention (UNC) path and file name of the standard (complete) .srf file to include in a Standard Repository File upgrade kit. Browsing is not available for Standard Repository File upgrade kits.
Minimum Old Version	Read-only text field	Earliest component version that can download and install the upgrade kit. Automatically set to NULL for history-independent component types, including repository files, indicating that there are no prerequisite versions required for using the kit. For more information about choosing version values, see <a href="#">“Determining Version Setting Values” on page 31</a> .

Table 18. Upgrade Kit Wizard Elements for a Siebel Repository File Upgrade Kit

Wizard Element	Element Type	Comments
Maximum Old Version	Read-only text field	Latest component version that can download and install the upgrade kit. Automatically set to NULL for history-independent component types, including repository files, indicating that there are no prerequisite versions required for using the kit. For more information about choosing version values, see <a href="#">“Determining Version Setting Values” on page 31.</a>
Base Version	Text field	<p>For a delta Repository File upgrade kit, specifies the version number of the repository file that was used as a base for the changes that the delta kit will distribute.</p> <p>To determine the version number of the base repository file, open a command window, navigate to <i>SIEBEL_ROOT\bin</i> or <i>SIEBEL_CLIENT_ROOT\bin</i>, and enter the following command, substituting the name of the base repository file and the language code used for that repository:</p> <pre>srfstamp /r repository_file_name /l language-code</pre>
New Version	Text field	Version number that the component being upgraded will have after the upgrade kit is installed. Default value is 1 greater than the current version for the component to be upgraded. For a Siebel Repository File upgrade kit, the New Version Value <i>must</i> be an integer. For more information about choosing version values, see <a href="#">“Determining Version Setting Values” on page 31.</a>
Activate Upgrade Kit	Check box	<p>When this check box is selected, the files to be included in the upgrade kit will be compressed into a single archive on the Siebel File System automatically. Activation can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a></p> <p>For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. (For example, a repository kit might be dependent on a database schema kit.) For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation” on page 94.</a></p>

Table 18. Upgrade Kit Wizard Elements for a Siebel Repository File Upgrade Kit

Wizard Element	Element Type	Comments
Apply Versions	Check box	<p>Displayed only if Activate Upgrade Kit check box was selected in an earlier Upgrade Kit Wizard screen. When this check box is selected, the compiled information string in the database will be updated automatically with the component version information for this upgrade kit. Applying versions can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a></p> <p>For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation” on page 94.</a></p>
Required Upgrade Kit	Check box	<p>When this check box is selected, the upgrade kit will be required regardless of previous versions installed. (Min Version and Max Version are set equal to New Version, but this change is not visible in the Upgrade Kit Wizard.) Displayed only if Apply Versions check box was selected in an earlier Upgrade Kit Wizard screen.</p>
Upgrade Kit Title	Text field	<p>Identifier for the upgrade kit. Defaults to Upgrade Component Name value followed by a space and the New Version value, but can be modified during the kit definition process. After the kit is defined, this identifier appears in the Name field of the Upgrade Kits list.</p>
Comments	Text field	<p>Available for comments about the upgrade kit. Comments entered here are displayed in the Upgrade Kits list.</p>

**NOTE:** It is recommended that you keep only one upgrade kit for history-independent components, including repository file kits. Depending on your preference, you can either delete or inactivate previous kits for a history-independent component. Deleting a kit increases available space in the Siebel File System. Inactivating a kit prevents use of the kit, while keeping it available in case you need it unexpectedly.

## Defining a Third-Party Software Upgrade Kit

This section provides specific instructions for defining a Third-Party software upgrade kit using the Upgrade Kit Wizard.



For an example of the creation of a Third-Party software upgrade kit, see [“Example of Constructing a Third-Party Upgrade Kit” on page 134](#).

**NOTE:** Siebel Anywhere Third-Party software components are meant to support third-party software products that are needed for operation with Siebel applications. Siebel Anywhere is not a general purpose tool for software configuration management and distribution.

### *To define a Third-Party software upgrade kit*

- 1 Gather the information you will need, as described in [Chapter 3, “Upgrade Planning and Preliminary Tasks.”](#)
- 2 Start the Upgrade Kit Wizard using the instructions in [“Running the Upgrade Kit Wizard” on page 62](#).
- 3 Use the information you have gathered and the information in [Table 19 on page 82](#) to respond to the prompts in the Upgrade Kit Wizard.
- 4 When you have finished specifying data about the upgrade kit you are defining, click Finish to pass the request to the Upgrade Kit Builder and to exit from the Upgrade Kit Wizard.

A new row for the new upgrade kit appears in the Upgrade Kits list, with Status set to Request Submitted. For more information about the kit information available in this list and other lists, see [“Viewing Upgrade Kit Properties” on page 88](#).

After you define your upgrade kit, you must proceed to activate, apply, and distribute it. For information on these tasks, see [Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”](#)

Information in Table 19 is presented in approximately the sequence used by the Upgrade Kit Wizard.

Table 19. Upgrade Kit Wizard Elements for a Third-Party Software Upgrade Kit

Wizard Element	Element Type	Comments
Upgrade Component	Drop-down list	<p>Name of the component the upgrade kit will install or upgrade. Choose one of the following predefined components, or a custom component you have created:</p> <ul style="list-style-type: none"> <li>■ Third Party - Oracle 8 Client</li> <li>■ Third Party - Microsoft Word</li> <li>■ Third Party - Microsoft SQL Server Driver</li> <li>■ Third Party - Microsoft Internet Explorer</li> <li>■ Third Party - IBM DB2 Client</li> <li>■ Third Party - Adobe Acrobat Reader</li> <li>■ Third Party - Adobe Acrobat</li> </ul>
UNC Path for Software Directory	Text field	Path to the directory that contains the files to include in the upgrade kit. All files in the specified directory are included in the kit automatically.
Destination Directory	Text field	Specifies where the files from the upgrade kit are to be placed on the client machine. Default value is \$SiebelRoot\upgrade\%KitName, where \$SiebelRoot refers to the Siebel client root directory, and %KitName refers to a subdirectory with the name of the upgrade kit. Both of these variables are case-sensitive.
Delete destination file when done	Check box	When this check box is selected, the upgrade kit files will be deleted, automatically, after the kit is installed.
Specify file to execute	Text field	<p>Specifies the file that runs the software installer on the subscriber's computer, such as install.exe. Only one file per upgrade kit can be specified for execution.</p> <p>Be sure you specify a file that can be executed, such as a .bat or .exe file. If you specify a file that <i>cannot</i> be executed, such as a .txt file or a CFG file, the upgrade kit will not install properly.</p>
Enter command line arguments	Text field	Specifies any command line arguments to use when running the file chosen in the Select file to execute field.

Table 19. Upgrade Kit Wizard Elements for a Third-Party Software Upgrade Kit

Wizard Element	Element Type	Comments
Minimum Old Version	Text field	Earliest component version that can download and install the upgrade kit. When this field is blank, indicates that there are no prerequisite versions required for using the kit. Defaults to the current component version recorded in the database. For more information about choosing version values, see <a href="#">“Determining Version Setting Values” on page 31.</a>
Maximum Old Version	Text field	Latest component version that can download and install the upgrade kit. When this field is blank, indicates that there are no prerequisite versions required for using the kit. For more information about choosing version values, see <a href="#">“Determining Version Setting Values” on page 31.</a>
New Version	Text field	Version number that the component being upgraded will have after the upgrade kit is installed. Default value is 1 greater than the current version for the component to be upgraded. For more information about choosing version values, see <a href="#">“Determining Version Setting Values” on page 31.</a>
Activate Upgrade Kit	Check box	<p>When this check box is selected, the files to be included in the upgrade kit will be compressed into a single archive on the Siebel File System automatically. Activation can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a></p> <p>For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation” on page 94.</a></p>
Apply Versions	Check box	<p>Displayed only if Activate Upgrade Kit check box was selected in an earlier Upgrade Kit Wizard screen. When this check box is selected, the compiled information string in the database will be updated automatically with the component version information for this upgrade kit. Applying versions can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a></p> <p>For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation” on page 94.</a></p>

Table 19. Upgrade Kit Wizard Elements for a Third-Party Software Upgrade Kit

Wizard Element	Element Type	Comments
Required Upgrade Kit	Check box	When this check box is selected, the upgrade kit will be required regardless of previous versions installed. (Min Version and Max Version are set equal to New Version, but this change is not visible in the Upgrade Kit Wizard.) Displayed only if Apply Versions check box was selected in an earlier Upgrade Kit Wizard screen.
Upgrade Kit Title	Text field	Identifier for the upgrade kit. Defaults to Upgrade Component Name value followed by a space and the New Version value, but can be modified during the kit definition process. After the kit is defined, this identifier appears in the Name field of the Upgrade Kits list.
Comments	Text field	Available for comments about the upgrade kit. Comments entered here are displayed in the Upgrade Kits list.

## Defining a Siebel Customer Revisions Upgrade Kit

This section provides specific instructions for defining a Siebel Customer Revisions upgrade kit using the Upgrade Kit Wizard. These instructions can be used for the following types of upgrade components, where (base) indicates a language-independent component, and *\_[language-code]* indicates a component that applies to a particular language represented by a code such as ENU (English) or DEU (German):

- Siebel Client Customer Revisions (base)
- Siebel Client Customer Revisions\_*\_[language-code]*

These instructions can also be used for custom components.

**CAUTION:** Do not use the upgrade kit type Customer Revisions for upgrade kits containing a Siebel SRF or CFG file. (Using a Customer Revisions kit type to deliver a Siebel SRF or CFG file can result in upgrade failures or incorrect version checking and version numbering.) Instead, use the Siebel Client Repository File and Siebel Client CFG kit types, respectively, for these upgrades.

### To define a Siebel Customer Revisions upgrade kit

- 1 Gather the information you will need, as described in [Chapter 3, “Upgrade Planning and Preliminary Tasks.”](#)
- 2 Start the Upgrade Kit Wizard using the instructions in [“Running the Upgrade Kit Wizard” on page 62.](#)
- 3 Use the information you have gathered and the information in [Table 20 on page 85](#) to respond to the prompts in the Upgrade Kit Wizard.

- 4 When you have finished specifying data about the upgrade kit you are defining, click Finish to pass the request to the Upgrade Kit Builder and to exit from the Upgrade Kit Wizard.

A new row for the new upgrade kit appears in the Upgrade Kits list, with Status set to Request Submitted. For more information about the kit information available in this list and other lists, see “Viewing Upgrade Kit Properties” on page 88.

After you define your upgrade kit, you must proceed to activate, apply, and distribute it. For information on these tasks, see Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”

Information in Table 20 is presented in approximately the sequence used by the Upgrade Kit Wizard.

Table 20. Upgrade Kit Wizard Elements for a Siebel Customer Revisions Upgrade Kit

Wizard Element	Element Type	Comments
Upgrade Component	Drop-down list	Name of the component the upgrade kit will install or upgrade: For customer revisions, select one of the following, as needed: <ul style="list-style-type: none"> <li>■ Siebel Client Customer Revisions</li> <li>■ Siebel Client Customer Revisions_<i>[language-code]</i></li> </ul> For a custom component, select the name of the component you created.
Select File Uploading Method	Radio button	Method for specifying files to include in the upgrade kit. Select one of the following: <ul style="list-style-type: none"> <li>■ <b>Add Files.</b> Use when the number of files to be added is small.</li> <li>■ <b>Use UNC Path.</b> Use when the number of files to be added is large. Gather all the files to be added into one directory before proceeding, as you will specify the directory in a following screen of the wizard.</li> </ul>
Files to Add	Text field	Displayed only if the Add Files radio button was selected in an earlier Upgrade Kit Wizard screen. Names of files to include in the upgrade kit. This field is populated either by entering one file name and path, or by clicking Browse and choosing a file from the directory listings displayed. In either case, click the Add button after specifying each path and file name combination. Repeat to add more files.
Browse	Button	Displayed only if the Add Files radio button was selected in an earlier Upgrade Kit Wizard screen. Displays a standard dialog box for browsing and choosing a file.
Add	Button	Displayed only if the Add Files radio button was selected in an earlier Upgrade Kit Wizard screen. When this button is clicked, the information in the Files To Add field is saved. If the Add button is not clicked, the information is discarded when you click Next, and an error message is displayed.

Table 20. Upgrade Kit Wizard Elements for a Siebel Customer Revisions Upgrade Kit

Wizard Element	Element Type	Comments
Added Files	Drop-down list	Displayed only if the Add Files radio button was selected in an earlier Upgrade Kit Wizard screen. Displays the names of the files added to the kit so far. Automatically populated when you enter information in Files to Add and then click the Add button.
Remove	Button	Displayed only if the Add Files radio button was selected in an earlier Upgrade Kit Wizard screen. When clicked, the Upgrade Kit Wizard discards the file information currently selected and displayed in the Added Files drop-down list. Other items in the drop-down list are not affected.
UNC Path for File Directory		Displayed only if the Use UNC Path radio button was selected in an earlier Upgrade Kit Wizard screen. Specifies the location of a directory that contains all the files to include in the upgrade kit. All the files in the directory you specify will be included in the kit.
Destination Directory	Text field	Specifies where the files of the upgrade kit are to be placed on the client machine.
Delete destination file when done	Check box	When this check box is selected, the upgrade kit files will be deleted, automatically, after the kit is installed.
Select file to execute	Drop-down list	Specifies one file in the upgrade kit to be executed as part of installation of the kit. Available values are the values stored in the Added Files field.  Be sure you specify a file that can be executed, such as a .bat or .exe file. If you specify a file that <i>cannot</i> be executed, such as a .txt file or a CFG file, the upgrade kit will not install properly.
Enter command line arguments	Text field	Specifies any command line arguments to use when running the file chosen in the Select file to execute drop-down list.
Acceptable Return Code	Numeric field	Optionally specifies the numeric code that the selected file to execute will return upon successful execution. Default value is zero. After the file is executed, Upgrade Wizard checks the actual return code against the value you specify in this setting.
Minimum Old Version	Text field	Earliest component version that can download and install the upgrade kit. When this field is blank, indicates that there are no prerequisite versions required for using the kit. Defaults to the current component version recorded in the database. For more information about choosing version values, see <a href="#">“Determining Version Setting Values”</a> on page 31.

Table 20. Upgrade Kit Wizard Elements for a Siebel Customer Revisions Upgrade Kit

Wizard Element	Element Type	Comments
Maximum Old Version	Text field	Latest component version that can download and install the upgrade kit. When this field is blank, indicates that there are no prerequisite versions required for using the kit. For more information about choosing version values, see <a href="#">“Determining Version Setting Values” on page 31.</a>
New Version	Text field	Version number that the component being upgraded will have after the upgrade kit is installed. Default value is 1 greater than the current version for the component to be upgraded. For more information about choosing version values, see <a href="#">“Determining Version Setting Values” on page 31.</a>
Activate Upgrade Kit	Check box	<p>When this check box is selected, the files to be included in the upgrade kit will be compressed into a single archive on the Siebel File System automatically. Activation can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a></p> <p>For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation” on page 94.</a></p>
Apply Versions	Check box	<p>Displayed only if Activate Upgrade Kit check box was selected in an earlier Upgrade Kit Wizard screen. When this check box is selected, the compiled information string in the database will be updated automatically with the component version information for this upgrade kit. Applying versions can also be performed manually, as described in <a href="#">Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”</a></p> <p>For most upgrade kits, it is recommended that you select this check box. However, if you are creating an upgrade kit that is dependant on another upgrade kit, be sure to clear this check box. For information about working with dependent kits, see <a href="#">“Controlling the Order of Kit Installation” on page 94.</a></p>
Required Upgrade Kit	Check box	When this check box is selected, the upgrade kit will be required regardless of previous versions installed. (Min Version and Max Version are set equal to New Version, but this change is not visible in the Upgrade Kit Wizard.) Displayed only if Apply Versions check box was selected in an earlier Upgrade Kit Wizard screen.

Table 20. Upgrade Kit Wizard Elements for a Siebel Customer Revisions Upgrade Kit

Wizard Element	Element Type	Comments
Upgrade Kit Title	Text field	Identifier for the upgrade kit. Defaults to Upgrade Component Name value followed by a space and the New Version value, but can be modified during the kit definition process. After the kit is defined, this identifier appears in the Name field of the Upgrade Kits list.
Comments	Text field	Available for comments about the upgrade kit. Comments entered here are displayed in the Upgrade Kits list.

## Defining a Custom Component Upgrade Kit

The instructions for defining a custom component upgrade kit depend on the component type you choose when you create the custom component. For information about which instructions to use for each component type, see [Table 21](#).

Table 21. Instructions for Defining a Custom Component Upgrade Kit

If Custom Component Type Is	Use the Following Instructions
Siebel CFG File	<a href="#">“Defining a Siebel Configuration File (CFG) Upgrade Kit” on page 64</a>
Siebel Database Extension	<a href="#">“Defining a Siebel Database Schema Upgrade Kit” on page 66</a>
Siebel Executables	<a href="#">“Defining a Siebel Client Executables Upgrade Kit” on page 70</a>
Siebel Repository File	<a href="#">“Defining a Siebel Repository File Upgrade Kit” on page 76</a>
Third-Party Software	<a href="#">“Defining a Third-Party Software Upgrade Kit” on page 80</a>
Customer Revision	<a href="#">“Defining a Siebel Customer Revisions Upgrade Kit” on page 84</a>

## Viewing Upgrade Kit Properties

After you use the Upgrade Kit Wizard to define an upgrade kit, information about the kit becomes visible in your Siebel application. There are two general types of information available:

- Information that applies to the kit as a whole, and the component it will upgrade. See [“Properties of the Entire Upgrade Kit and Its Components.”](#)
- Information that applies to items and item parameters within the kit. See [“Properties of Upgrade Kit Items and Upgrade Kit Item Parameters” on page 91.](#)



## Properties of the Entire Upgrade Kit and Its Components

The following procedure describes how to view and interpret information that applies to the upgrade kit as a whole, and the components it will upgrade.

### *To view properties of an existing upgrade kit and its components*

- 1 From the application-level menu, select **Navigate > Site Map > Administration - Siebel Anywhere**.
- 2 From the link bar, select **Upgrade Kits**.

The Upgrade Kits list and the Upgrade Kit Components list appear.

- 3 In the Upgrade Kits list, select the record for the upgrade kit for which you want to view properties.

For information about the fields in the Upgrade Kits list, which displays each upgrade kit defined in the system, see [Table 22](#).

For information about how to interpret various values that are displayed in the Status field of this list, see [Table 23 on page 90](#).

For information about the fields in the Upgrade Kit Components list, which displays the components associated with the selected upgrade kit, see [Table 24 on page 90](#).

Table 22. Fields in the Upgrade Kits List

Name	Comment
Name	Name of the upgrade kit (called Upgrade Kit Title in the Upgrade Kit Wizard). The default name includes the component name and version in the kit. You can specify a different name when defining the kit. However, it is recommended that you use descriptive names if you do not accept the default wording. Also see <a href="#">Table 5 on page 29</a> for additional limitations for a name. The name cannot be modified once the kit has been defined.
Status	For definition of Status and the associated values and descriptions, see <a href="#">Table 23 on page 90</a> .
Archive Kit Name	Name of the compressed file containing the upgrade kit.
File Size	Size of the physical file containing the upgrade kit.
Comments	This field is available for comments.
Modified	The date and time this record was last modified.
Compiled Information	Compiled string that includes all information about the upgrade kit.

The Status column indicates whether the Upgrade Kit is Request Submitted (awaiting for Server to pick up) or In Progress (in process of being created). The status column may have other values as well such as Pending, Active, Error, and so on. It is not limited to just Request Submitted and In Progress. [Table 23](#) describes the values and description of the Status column.

Table 23. Status Column Values and Definitions, Upgrade Kit View

Values	Description
Active	Kit is ready to be applied and distributed.
Error	Indicator that there was an error while creating the kit.
Inactive	Kit on hold and not ready for further processing. First step to activate the kit is to change status to Pending. Then you can activate it.
In Progress	Creation of new upgrade kit is underway on the server.
Pending	Kit successfully created and ready to be activated.
Request Submitted	Request has been submitted to the server to create the new upgrade kit.

[Table 24](#) describes the fields in the Upgrade Kit Components list, which displays the components associated with the upgrade kit. The components are automatically created by the Upgrade Kit Wizard when you click Auto Create on the Upgrade Kits list.

When you select the new upgrade kit in the Upgrade Kit list, you will notice that two records appear in the Upgrade Kit Components list. The first record is the component to be upgraded. The second is the Siebel Upgrade Wizard. Including the Upgrade Wizard as a component reinforces using the correct version of the Upgrade Wizard for the upgrade.

**CAUTION:** Do not delete the default records in the Upgrade Kit Components list. These records store key information about how components work, and provide models you may eventually need for setting up custom components.

Table 24. Fields in the Upgrade Kit Components List

Name	Comment
Name	Name of the component.
Min Old Version	Minimum component version that a subscriber must have to install this upgrade kit.
Max Old Version	Maximum component version that a subscriber may have to install this upgrade kit.
New Version	Version of this component the user will have after installing the kit.
Comments	This field is available for comments.

This list also controls the upgrade kit sequence (the order of upgrades), when several upgrade kits are used together. For example, if Kit 1 should be installed before Kit 2, you can add component Kit 1 upgrades as Upgrade Kit Components of Kit 2. For details, see [“Controlling the Order of Kit Installation” on page 94](#).

## Properties of Upgrade Kit Items and Upgrade Kit Item Parameters

Each upgrade kit item defines an action in an upgrade kit and requires certain parameters to operate. For example, a file copy action needs the file to copy and the destination for the copy. The following procedure describes how to view and interpret information that applies to items and item parameters within a selected upgrade kit.

**NOTE:** It is recommended that you do not add, update, or delete any records in the Upgrade Kit Items list and Upgrade Kit Item Parameters list in the Upgrade Kit Items view. The records in these lists are created automatically, and changing them in any way can damage the selected upgrade kit.

### *To view properties of an existing upgrade kit’s items and item parameters*

- 1 From the application-level menu, select **Navigate > Site Map > Administration - Siebel Anywhere**.
- 2 From the link bar, select **Upgrade Kit Items**.

The Upgrade Kits list, the Upgrade Kit Items list, and the Upgrade Kit Item Parameters list appear.

For information about the fields in the Upgrade Kits list, see [Table 22 on page 89](#) and [Table 23 on page 90](#).

For information about the fields in the Upgrade Kit Items list, which displays the individual items that define actions to be executed by a selected upgrade kit, see [Table 25](#).

For information about the fields in the Upgrade Kit Item Parameters list, which displays the parameters associated with a selected upgrade kit item, see [Table 26 on page 92](#).

Table 25. Fields in the Upgrade Kit Items List

Field	Comment
Comments	This field is available for comments.
Estimated Disk Space	The disk space required to download and execute an item (optional). When the upgrade kit is created, the value for this field defaults to 0. You can enter a new value (in bytes) before activating the kit.  Before a subscriber installs the upgrade kit, the Upgrade Wizard validates that the specified amount of free disk space is available. If adequate disk space is not available, the Upgrade Wizard returns an error message.
Item Type	The type of item to execute.
Sequence	The order in which the Upgrade Wizard executes the item.
Title	The title of the item, which is displayed in the Upgrade Wizard.

Each upgrade kit item defines an action in an upgrade kit and requires certain parameters to operate. For example, a file copy action needs the file to copy and the destination for the copy. The Upgrade Kit Item Parameters list includes the parameters associated with each upgrade kit item. [Table 26](#) describes the fields of this list.

Table 26. Fields in the Upgrade Kit Item Parameters List

Field	Comment
Attachment Name	The name of the file associated with this item.
Comments	This field is available for comments.
Date	Date and time when the file is attached.
File Ext	The extension of the file associated with this item.
Name	Name of the parameter.
Size	The size of the file associated with this item.
Value	The value of the parameter.

# 5

## Activating, Applying, and Distributing Upgrade Kits

This chapter discusses activating, applying, and distributing upgrade kits, plus related tasks. It includes the following topics:

- [“Process of Completing Upgrade Kit Creation” on page 93](#)
- [“Activating an Upgrade Kit” on page 93](#)
- [“Controlling the Order of Kit Installation” on page 94](#)
- [“Deactivating an Upgrade Kit” on page 95](#)
- [“Applying an Upgrade Kit” on page 96](#)
- [“Converting an Optional Kit to a Required Kit” on page 98](#)
- [“Distributing Upgrade Kits” on page 98](#)
- [“Process of Limiting Distribution of an Upgrade Kit” on page 100](#)
- [“Distributing Different CFG Files to Different Users” on page 100](#)

### Process of Completing Upgrade Kit Creation

The remaining three steps in the process of creating an upgrade kit are activating, applying, and distributing the kit.

- 1 [“Activating an Upgrade Kit” on page 93](#). This step gathers all the files for the upgrade and creates a single compressed archive file.
- 2 [“Applying an Upgrade Kit” on page 96](#). This step updates the compiled information string in the database with the new version information. During this step, if you have not already done so, you indicate whether or not a kit will be required.
- 3 [“Distributing Upgrade Kits” on page 98](#). This step makes the upgrade kit available to designated subscribers as either a required kit or an optional kit, depending on your selection in [Step 2](#) of this process.

**NOTE:** There are additional instructions for distributing database schema upgrade kits. For information, see [“Performing Database Schema Updates” on page 123](#).

### Activating an Upgrade Kit

After you define the upgrade kits for an upgrade, you need to activate each kit. The status of a kit must be Pending before activating it. Activating a kit gathers the files to be included in the upgrade kit and compresses them into a single archive on the Siebel File System. It also sets the status of the upgrade kit to Active.

This task is a step in [Process of Completing Upgrade Kit Creation on page 93](#) and in [Process of Limiting Distribution of an Upgrade Kit on page 100](#).

### *To activate an upgrade kit*

- 1 From the application-level menu, select **Navigate > Site Map > Administration - Siebel Anywhere**.
- 2 From the link bar, click **Upgrade Kits**.  
The Upgrade Kits view appears.
- 3 Select the kit you want to activate.  
The Status field must show that it is Pending.
- 4 Click **Activate**.  
A dialog box appears indicating activation is in progress.

If you are activating a large kit, such as a repository file (SRF) kit, your browser may appear unresponsive for a few minutes after you click **Activate**. The delay in response occurs because the browser must wait for a reply from the server before proceeding. During the wait, no hourglass is displayed. When the browser receives the server's reply, the Status field changes to **Active**.

Repeat [Step 3](#) and [Step 4](#) to activate each kit required for your upgrade.

**NOTE:** Activating the upgrade kit does not affect the version information stored in the database. Activating a kit populates the File Size field for that kit. The kit is not available to subscribers until it has been applied and distributed.

When you activate the upgrade kit, the kit items you see in this view turn into item entries in the upgrade.ucf, which is part of every Siebel Anywhere upgrade kit.

## Controlling the Order of Kit Installation

Usually, when you distribute upgrades to a configuration requiring several related components, the order in which kits are installed is undefined. If you need to control the installation order of kits, you can do this by making one kit dependent upon another. If this is the case, do not use the Upgrade Kit Wizard to activate the kit.

An example of this type of situation is when you have one Siebel Repository File kit (Kit 1) and another Siebel Client Customer Revision kit (Kit 2), and you need Kit 1 to be installed first.

### *To set the order of kit installation*

- 1 From the application-level menu, select **Navigate > Site Map > Administration - Siebel Anywhere**.
- 2 From the link bar, click **Upgrade Kits**.  
The Upgrade Kits list appears.
- 3 In the Upgrade Kits list, select the kit you need installed second (for example, Kit 2, the Siebel Client Customer Revision kit).

- 4 If the kit is already activated, perform one of the following actions:
  - Manually set the status to Pending.
  - Deactivate the kit and change the status to Pending.
- 5 In the Upgrade Kit Component list (lower list in the same view), make sure that the kit to be installed second is still selected in the list above, and then click New.  
The Upgrade Components dialog box appears.
- 6 Select the component Kit 1 upgrades (for example, Siebel Client Repository\_<Language>).
- 7 In the new record, set the Min Old Version to the New Version of the kit (the Siebel Repository File).
- 8 Click to activate the Upgrade Kits applet.
- 9 Select the kit you need installed second again and click Activate.

This procedure forces first the installation of Kit 1 (Siebel Repository File), followed by Kit 2 (Siebel Client Customer Revision). This interlinked series can be extended to three or more kits.

## Deactivating an Upgrade Kit

There are times when you may need to deactivate an active kit—perhaps there was a last minute decision to delete a kit. (You must first deactivate a kit before deleting it.)

Observe the following guidelines when deciding whether to deactivate a kit:

- If there was a problem with the kit, then the kit can be deactivated and a replacement kit should be created to upgrade the users who had problems with the initial kit.
- If later kits depend on the kit you are considering for deactivation, perform the following actions to make sure that users will be able to download and install the later kits:
  - Make sure that every user who needs the upgrade kit that will be deactivated has already installed the kit. This includes any infrequent users who may log on too infrequently to have been prompted to install the kit.
  - **CAUTION:** Deactivating a previously distributed kit could cause mobile users to have an unsuccessful synchronization, and place those users unnecessarily into a read-only state.
  - Make sure that you can clone new client installations from existing up-to-date clients. This method is the preferred method for providing new users with material that was formerly delivered in one or more upgrade kits. New users cannot download later kits if a prerequisite kit has been deactivated.

### *To deactivate an upgrade kit*

- 1 From the application-level menu, select Navigate > Site Map > Administration - Siebel Anywhere.
- 2 From the link bar, click Upgrade Kits.  
The Upgrade Kits view appears.

- 3 Select the kit you want to deactivate.

The Status field should show that it is Active.

- 4 Click the menu button and select Deactivate.

A warning message will ask you to confirm. If appropriate, click OK.

The Status field should change to Inactive.

Repeat [Step 3](#) and [Step 4](#) for each kit that you want to deactivate.

## Applying an Upgrade Kit

After you activate the upgrade kits for an upgrade, you need to apply each kit. Applying a kit updates the compiled information string in the database with the component version information. This is an essential step in creating an upgrade kit. This is also the step in which you indicate whether a kit is a required upgrade kit.

This task is a step in [Process of Completing Upgrade Kit Creation on page 93](#) and in [Process of Limiting Distribution of an Upgrade Kit on page 100](#).

Upgrade.ucf is the driver file for the Upgrade Wizard to apply the upgrade kit. It contains an ordered list of action items for the Upgrade Wizard to execute during the installation of the upgrade kit.

The following procedure describes how to apply an upgrade kit. This updates the compiled information string in the database with the component version information.

**NOTE:** If a replacement kit uses the same values for New Version, Minimum Old Version, and Maximum Old Version as the deactivated kit it replaces, and if the deactivated kit was previously distributed, you do not need to apply or distribute the replacement kit.

### *To apply an upgrade kit*

- 1 Navigate to the Administration - Siebel Anywhere screen, and from the link bar, click Upgrade Kits.

The Upgrade Kits view appears.

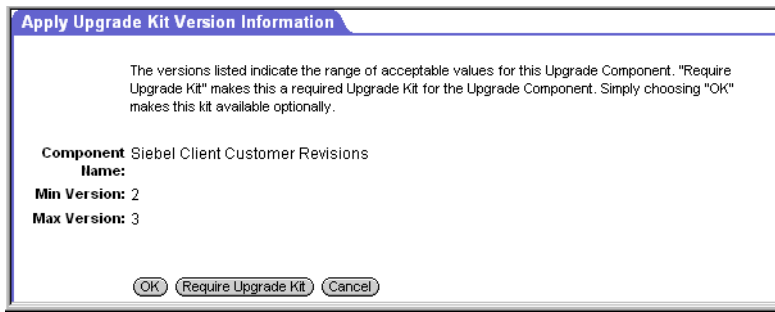
- 2 Select the kit you want to apply.

The Status field must show that it is Active.



3 Click Apply Versions.

The Apply Upgrade Kit Version Information dialog box appears, as shown in the following figure. Review the information in this dialog box. The paragraphs following the figure provide additional information.



- The Min Version and Max Version numbers in this dialog box apply to the version of the component that can be used to bring up the application in read/write mode. If users have a version below the minimum and choose not to install the upgrade, they can only access the application in a read-only mode.
- If you click OK without clicking Require Upgrade Kit, you are making it an optional kit (that is, after the kit is distributed, your subscribers can use Siebel Business Applications without upgrading—provided their version is between the minimum and maximum).
- If you click Require Upgrade Kit, the minimum version changes to match the maximum version, which is the new version. In this case, the subscribers will have to upgrade after the configuration is distributed; otherwise, they can only bring up the application in a read-only mode.

**NOTE:** It is strongly recommended that you create upgrade kits as optional and then test to make sure the kit is functioning properly. After you thoroughly test a kit using retrieval and installation, you can return to the Upgrade Kits view and reapply and distribute the kit as a required kit. See [“Distributing Upgrade Kits” on page 98](#) for more information.

4 Click OK if the version information is correct or click Cancel to exit the dialog box without applying the upgrade kit.

If you click OK, a prompt reminds you that you need to distribute the kit to make it available to subscribers. For information on distributing kits, see [“To distribute an upgrade kit” on page 99](#).

Repeat [Step 2](#) through [Step 4](#) for each kit in this upgrade. Continue to [“Distributing Upgrade Kits” on page 98](#) for distributing updated version information for upgrade kits.

**NOTE:** If there are multiple kits of the same type that are either SRF, Database Schema, or Client Executable, it is strongly recommended you deactivate the old kits. This will save time for your users by preventing the downloading of outdated kits. (The version information for a deactivated kit can be retrieved, but a deactivated kit, itself, cannot be retrieved. When a user tries and fails to retrieve a deactivated kit, an automatic attempt is made to retrieve a later kit that is currently activated.)

## Converting an Optional Kit to a Required Kit

After thoroughly testing an optional kit, you can reapply the kit to convert it to a required upgrade kit.

### *To convert an optional kit to a required kit*

- 1 From the application-level menu, select **Navigate > Site Map > Administration - Siebel Anywhere**.
- 2 From the link bar, click **Upgrade Kits**.  
The Upgrade Kits view appears.
- 3 From the Upgrade Kits list, select the optional kit you want to convert.  
The current status should be **Active**.
- 4 Click **Apply Versions**.  
The Apply Upgrade Kit Version Information dialog box appears. Review the information in this dialog box.
- 5 Click **Require Upgrade Kit** to make the upgrade kit required.  
The Minimum and Maximum Version numbers are now the same because the kit is now required.
- 6 Click **OK** if the version information is correct or click **Cancel** to exit the dialog box without applying the upgrade kit.  
If you click **OK**, a prompt reminds you that you need to distribute the kit to make it available to subscribers. See [“Distributing Upgrade Kits” on page 98](#).

## Distributing Upgrade Kits

After applying an upgrade kit, you need to distribute it to one or more configurations. For general information about configurations, see [“Upgrade Configurations” on page 22](#).

This task is a step in [Process of Completing Upgrade Kit Creation on page 93](#) and in [Process of Limiting Distribution of an Upgrade Kit on page 100](#).

**NOTE:** If a replacement kit uses the same values for **New Version**, **Minimum Old Version**, and **Maximum Old Version** as the deactivated kit it replaces, and if the deactivated kit was previously distributed, you do not need to distribute the replacement kit.

Upgrade kits are distributed to one configuration at a time. The distribution process makes the kit’s component version information available to all the subscribers who are associated with that configuration—this gives the subscribers their first opportunity to download and install the kit. Neither required nor optional kits are available to subscribers until the version information is distributed.

The version information that is made available during the distribution process includes version information for *all* the related components of the configuration, not just the component that the new kit is designed to upgrade. Technically, therefore, when you distribute a new upgrade kit to a configuration, you are making that kit's version information available to the configuration, and you are redistributing the version information of other active kits for the configuration's related components. In practice, however, you generally can proceed as if you were distributing only the one kit.

After distribution, if the kit is required, subscribers are automatically prompted to retrieve and install it. If the kit is optional, subscribers use the Component Upgrades view and Upgrade Wizard to locate and install the kit.

**CAUTION:** Before you distribute a kit, you must define, activate, and apply it. Before you distribute a kit to any configuration containing production users, it is strongly recommended that you use the Siebel Test Client configuration to test the kit thoroughly. It is particularly important to test any required kit before distributing for general use, because Mobile Web Client users can suffer unnecessary delays while downloading a defective required kit, and they can lose read/write access to Siebel Business Applications until a correctly functioning kit is available.

The following procedure provides the steps to distribute a kit from the Upgrade Configurations view. The same procedure can also be accomplished from the Employees view.

**NOTE:** There are additional instructions for distributing database schema upgrade kits. For information, see [“Performing Database Schema Updates” on page 123](#).

### *To distribute an upgrade kit*

- 1 From the application-level menu, select [Navigate > Site Map > Administration - Siebel Anywhere](#).
- 2 From the link bar, click [Upgrade Configurations](#).
- 3 In the Upgrade Configurations view, select the appropriate Anywhere subscriber configuration.
- 4 Inspect the Upgrade Components list to verify that the components in the upgrade kit are related components for the selected configuration.

For example, for a Siebel maintenance release, the related components must include the Siebel Client Executables component for the base upgrade kit and the Siebel Client Executables\_*[language-code]* component for each language-specific upgrade kit.

- 5 Click [Distribute](#).

This action makes the version information in the database available to subscribers with that particular configuration.

A dialog box appears to confirm the intent to distribute the related components to the configuration.

Repeat [Step 3](#) through [Step 5](#) for other configurations, as applicable.

The following sections provide additional information about special cases of distribution:

- [“Process of Limiting Distribution of an Upgrade Kit” on page 100](#)
- [“Distributing Different CFG Files to Different Users” on page 100](#)

## Process of Limiting Distribution of an Upgrade Kit

You can limit the distribution of a upgrade kit to certain subscribers by temporarily assigning subscribers to a new configuration. For example, you might have a situation where you want to send a set of reports only to management, or you want to limit the users for testing a kit.

To limit kit distribution for a new configuration, perform the following tasks:

- 1 [“Defining Upgrade Kits” on page 61](#). Define the upgrade kit that you will be distributing, if you have not already done so.
- 2 [“Activating an Upgrade Kit” on page 93](#). Activate the upgrade kit that you will be distributing, if you have not already done so.
- 3 [“Applying an Upgrade Kit” on page 96](#). Apply the upgrade kit that you will be distributing, if you have not already done so.
- 4 [“Creating a New Configuration” on page 39](#). Either create a new configuration that you will use to distribute the kit to specific subscribers, or use the Siebel Test Client configuration for this purpose.
- 5 [“Adding Components to a Configuration” on page 39](#). Associate the configuration with one or more components that correspond to the upgrade kit or kits that you want to distribute.
- 6 [“Assigning Employees to a Configuration” on page 41](#). Add the login for the users you want to receive this kit to the Employee view for this configuration.
- 7 [“Distributing Upgrade Kits” on page 98](#). Distribute active upgrade kits to the configuration. When you select a configuration and click Distribute, the subscribers associated with the configuration gain access to version information for all of the related components for the configuration. This gives those subscribers potential access to all active upgrade kits for those components, including any new upgrade kits that have been defined, activated, and applied. Subscribers associated with other configurations do not gain access to your new kit until you distribute to their configurations.
- 8 [“Removing Employees from a Configuration” on page 42](#). Give the subscribers who are associated with the configuration enough time to download the kit. However, you need to disassociate these subscribers from this special configuration as soon as possible. Ending a user’s dynamic configuration assignment returns that user to the original configuration assignment defined in that user’s CFG file. Kits targeted for the original configuration are again available to the user.

## Distributing Different CFG Files to Different Users

It is possible to distribute different CFG files to different sets of users. For example, you might have two groups of users, Sales users and Engineering users, each group using a different CFG file with their respective Siebel Applications. The following procedure describes how to use Siebel Anywhere to send a new CFG file to each group.

### *To distribute different CFG files to different groups of users*

- 1 Prepare all of the CFG files your various groups will use.

- 2 Launch the Siebel Application.
- 3 From the application-level menu, select **Navigate > Site Map > Administration - Siebel Anywhere**.
- 4 From the link bar, click **Upgrade Component List**.

The Upgrade Components list appears.

- 5 Create a component for each language, such as **Sales CFG\_ENU**, similar to the Siebel Call Center CFG component. Use the **Copy Record** option. Set the values for the columns for each record as shown in the following tables.

Example values for Record 1:

Field	Comment
Name	Sales CFG
Component Type	Siebel CFG_ENU File
Min Version	100
Max Version	Should be blank for a new component
Locate Method	CFG
Locate Information	Siebel,ClientRootDir,bin\enu\sales.cfg
Version Method	CFG
Version Information	Siebel,Version

Example values for Record 2:

Field	Comment
Name	Engineering CFG
Component Type	Siebel CFG_ENU File
Min Version	100
Max Version	Should be blank for a new component
Locate Method	CFG
Locate Information	Siebel,ClientRootDir,bin\enu\eng.cfg
Version Method	CFG
Version Information	Siebel,Version

- 6 Create a configuration for each language combination for each group of users (such as Sales and Engineering), and add the appropriate CFG components you just created as related components. See [Chapter 3, "Upgrade Planning and Preliminary Tasks"](#) for details.

- 7 Go to Administration - Siebel Anywhere > Upgrade Kits and use Auto Create to create one kit for each group of users.

For example, you might create one kit for Sales users, using the Sales CFG component and the Sales.cfg file, and another for Engineering users, using the Engineering CFG component and Eng.cfg file.

- 8 Apply the kits, and be sure to click on the Require Upgrade Kit button in the Apply Upgrade Kit Version Information dialog box.

The following procedure describes how to test the distribution of different CFG files to different groups of users.

### *To test distributing CFG files to different groups of users*

- 1 From the application-level menu, select Navigate > Site Map > Administration - Siebel Anywhere > Upgrade Configurations.
- 2 Add the new CFG components (such as Sales CFG and Engineering CFG) to the Siebel Test Client configuration.
- 3 Add two or more users to the Employee list for Siebel Test Client.
- 4 Distribute the Siebel Test Client to verify that you can retrieve the kit successfully.
- 5 Make sure that your test users' CFG files reflect the correct ComponentName in their CFG files.  
For example, Sales users should have this parameter set to Sales Configuration while Engineering users should have this parameter set to Engineering Configuration. Also, users should verify an exact match between the ComponentName in the CFG file and Configuration name in the Administration - Siebel Anywhere > Upgrade Configurations view.
- 6 Distribute the final configurations (such as Sales Configuration and Engineering Configuration) to send the kit to all users.

## Disabling Siebel Anywhere

There may be times when you do not want your users to be prompted to install upgrade kits when they log in to Siebel applications. The following procedures describe how to disable Siebel Anywhere version checking and the ability to install Siebel Anywhere upgrade kits for Developer Web Clients and Mobile Web Clients.

### *To disable Siebel Anywhere for Developer Web Clients*

- 1 Navigate to the Administration - Siebel Anywhere screen.
- 2 From the Show drop-down menu, select Upgrade Configurations.
- 3 Use one of the following methods to disable Siebel Anywhere version checking and related user prompts:

- Rename the appropriate configurations. For example, rename the Siebel Sales Client as Old Siebel Sales Client, and so on. Then, redistribute current version information to the renamed configuration.
- Delete all the related components in a configuration and redistribute current version information to that configuration.

### ***To disable Siebel Anywhere for Mobile Web Clients***

- 1 Follow the instructions in ["To disable Siebel Anywhere for Developer Web Clients."](#)
- 2 In addition, use one of the following methods to disable Siebel Anywhere version checking and related user prompts:
  - Change ComponentName parameter in the CFG file to None.
  - Reextract local database.

If Mobile Web Clients are associated with a particular configuration through the Administration - Siebel Anywhere > Upgrade Configurations > Employees view, reextracting is the only choice.





# 6

## Retrieving, Installing, and Testing Upgrade Kits

After you distribute an upgrade kit to a configuration, the subscribers associated with that configuration can retrieve and install that kit. Kits should be retrieved, installed, and tested by test subscribers before being distributed to production subscribers.

This chapter discusses retrieval, installation and testing of upgrade kits. It includes the following topics:

- [“Process of Retrieving, Installing, and Testing Upgrade Kits” on page 105](#)
- [“Retrieving and Installing Upgrade Kits” on page 106](#)
  - [“Retrieving Optional Upgrade Kits for Mobile Web Clients” on page 106](#)
  - [“Retrieving Optional Upgrade Kits for Developer Web Clients” on page 108](#)
  - [“Retrieving Required Upgrade Kits” on page 109](#)
  - [“Launching the Upgrade Wizard” on page 111](#)
  - [“About Client Upgrade Error Recovery” on page 111](#)
- [“Retrieving Upgrade Kits for Siebel Regional Node Servers” on page 111](#)
- [“Testing Upgrade Kits” on page 113](#)
- [“Uninstalling Siebel Maintenance Releases or Patches” on page 115](#)

## Process of Retrieving, Installing, and Testing Upgrade Kits

To retrieve, install, and test an upgrade kit, perform the following tasks:

- 1 [“Retrieving and Installing Upgrade Kits” on page 106](#). The procedure for retrieving an upgrade kit depends on its type. Kits should be retrieved by test subscribers before being distributed to production subscribers.
- 2 [“Launching the Upgrade Wizard” on page 111](#). Install the upgrade kit using the Upgrade Wizard. The process for invoking the Upgrade Wizard depends on the subscriber type. Kits should be installed by test subscribers before being distributed to production subscribers.
- 3 [“Testing Upgrade Kits” on page 113](#). Kits should be tested by test subscribers before being distributed to production users. (Among other possible problems, incorrectly constructed kits can prevent production users from writing data to Siebel Business Applications.)

If you encounter errors in the retrieval, installation, or testing process, see the following topics for additional information:

- [“About Client Upgrade Error Recovery” on page 111](#).
- [“Retrieving Upgrade Kits for Siebel Regional Node Servers” on page 111](#).

## Retrieving and Installing Upgrade Kits

This task is a step in [Process of Retrieving, Installing, and Testing Upgrade Kits on page 105](#).

The procedure for retrieving an upgrade kit depends on its type:

- Optional upgrade kits can be manually retrieved and installed by Siebel client subscribers through the Component Upgrades view.
- Required upgrade kits are automatically retrieved when mobile users synchronize or log into the application and when the server starts up or RepAgent synchronizes with Siebel Server.

It is a common misunderstanding that Mobile Web Clients retrieve kits directly from the Siebel File System. Instead:

- Mobile Web Clients request the files from the Siebel Remote Server. The Remote Server retrieves the files from the Siebel File System.
- Developer Web Clients retrieve kits (.saf files) directly from the Siebel File System or by using the File System Manager, depending on the settings defined in the active CFG file.

Subscribers use the Siebel Anywhere Upgrade Wizard to install upgrade kits after they are retrieved. The process for invoking the Upgrade Wizard varies by subscriber type:

- For client subscribers, the Upgrade Wizard is invoked automatically after an upgrade is retrieved.
- For Siebel Server subscribers, the Upgrade Wizard is invoked manually (see [“Retrieving Upgrade Kits for Siebel Regional Node Servers” on page 111](#)).

The process for retrieving and installing upgrade kits by Siebel Client subscribers is described in the following sections:

- [“Retrieving Optional Upgrade Kits for Mobile Web Clients” on page 106](#)
- [“Retrieving Optional Upgrade Kits for Developer Web Clients” on page 108](#)
- [“Retrieving Required Upgrade Kits” on page 109](#)

**NOTE:** When a Siebel Repository File upgrade kit is retrieved and installed, the size of the downloaded SRF file may differ from the original size of the SRF file included in the upgrade kit. The size change is related to details of how compression and decompression are implemented, and does not normally indicate an error. To check for correct file size and timestamp, use the size and timestamp from a test Mobile Web Client as the standard for other subscribers.

## Retrieving Optional Upgrade Kits for Mobile Web Clients

This section describes how to retrieve and install an optional upgrade kit for a Mobile Web Client. Optional upgrade kits are only for client subscribers. Upgrade kits for server subscribers should always be required.

**NOTE:** Siebel Smart Web Clients are not Siebel Anywhere subscribers.

The following procedure describes how to retrieve and install an optional kit for a Mobile Web Client.

**NOTE:** The following procedure restarts the Mobile Web Client computer. It is recommended that users save their work and exit from other applications before starting this procedure.

**To retrieve an optional kit for a Mobile Web Client**

- 1 Invoke the Mobile Web Client.
- 2 From the application-level menu, select File > Synchronize > Database.
- 3 In the Siebel Remote dialog box, click Synchronize.  
Synchronization takes place and is complete when the opposing arrows (red and blue) in the lower-right corner of the screen disappear.
- 4 From the application-level menu, select Navigate > Site Map > User Preferences > Component Upgrades.
- 5 In the Component Upgrades list, select the desired upgrade component, and check the check box under the Upgrade column.  
A check appears. The Status field value should be Upgrade Available for those components where optional kits are available. For information about possible values for the Status field, see [Table 27 on page 107](#).
- 6 Click Upgrade Selected Components to display the Siebel Anywhere Kit Download dialog box, and then complete the following substeps:
  - a Follow the instructions in the Siebel Anywhere Kit Download dialog box to start a Siebel Remote session and retrieve the Upgrade Kit necessary to upgrade your system.
  - b Click OK to close the Siebel Anywhere Kit Download dialog box.
- 7 Synchronize *again* as in [Step 2](#) and [Step 3](#) to retrieve the corresponding upgrade kits.
- 8 Click Upgrade Selected Components again to invoke the Upgrade Wizard for the desired upgrade component.  
After the Upgrade Wizard completes, it will automatically restart the Mobile Web Client.
- 9 Log in, repeat [Step 4](#) and, from the Component Upgrades list, verify the Status is Version OK for the desired upgrade component.

[Table 27](#) describes the fields in the Component Upgrades list.

Table 27. Fields in the Component Upgrades List

Field	Comment
Upgrade	Check this field to upgrade the component.
Component Name	Name of the upgrade component.

Table 27. Fields in the Component Upgrades List

Field	Comment
Status	Status of the upgrade component. Values can include the following:  <i>Upgrade Available.</i> An optional upgrade kit is available for this component.  <i>Upgrade Required.</i> A required upgrade kit is available for this component.  <i>Version OK.</i> Version of this upgrade component is current.  <i>Upgrade Not Available.</i> The upgrade component is out-dated and needs to be updated; however, no kit is available to do this. Contact your system administrator.
Current Version	The version of the component on the client machine.
Minimum Version	The minimum required version of the component.
Maximum Version	The maximum required version of the component.
Upgrade Size (bytes)	The size of the upgrade kit.
Download Time (mins)	The estimated time to download the kit on a 28.8 modem.

## Retrieving Optional Upgrade Kits for Developer Web Clients

Developer Web Clients can retrieve optional upgrade kits as described in the following procedure.

### *To retrieve an optional kit for a Developer Web Client*

- 1 Invoke the Developer Web Client.
- 2 From the application-level menu, select **Navigate > Site Map > User Preferences > Component Upgrades**.

In the Component Upgrades list, the upgrade status should be Upgrade Available for those components where optional kits are available. If you do not see the Upgrade Available status for the component you want to upgrade, contact your Siebel administrator, as this discrepancy may indicate a configuration problem.

After the administrator corrects the problem, refresh your own view of the Component Upgrades list and reinspect the upgrade status for components that have optional kits.

- 3 In the Component Upgrades list, select the desired upgrade component, and select the check box in the Upgrade column for the desired upgrade component.

4 Click Upgrade Selected Components.

This will download the upgrade kit, shut down the Developer Web Client application, and launch the Upgrade Wizard. After the Upgrade Wizard completes, it will automatically restart the Developer Web Client application. For some Third-Party upgrade kits, the client computer may shut down and restart.

5 Log in and, from the Component Upgrades list, verify the Status is Version OK for the desired upgrade component.

## Retrieving Required Upgrade Kits

Both Mobile and Developer Web Clients perform a version check during login to a Siebel application. Mobile Web Clients also perform a version check during synchronization.

If the version check detects that an upgrade is required, a dialog appears asking the user whether to upgrade or not. If the user chooses Yes, the upgrade kit will be downloaded. The following procedures for both the Mobile and Developer Web Clients illustrate this point.

Upgrade kits are *not* automatically retrieved upon version check. A prompt appears, asking if the user would like to retrieve the upgrade kit. The kit is downloaded only if the user clicks yes to download the kit.

The version check verifies the components used by the client configuration; multiple required components can be displayed in the dialog box.

**NOTE:** Beginning with Siebel version 7.5, synchronizing a Mobile Web Client automatically includes copying version information for Siebel Anywhere from the server to the local database. This allows each Mobile Web Client to detect new required and optional upgrades for the client's configuration without waiting for individual database transactions concerning versions to be synchronized.

If the user answers Yes, every required upgrade kit for the components used by the client configuration is automatically retrieved and the Upgrade Wizard is launched.

If the user answers No, the Developer Web Client starts in read-only mode against the server database, and the Mobile Web Client starts in read-only mode against the local database.

If the user answers No, the user's session continues in read-only mode against either the local or server database, until the upgrade has been completed. A prompt notifies the user of this status.

While operating in read-only mode, both Developer and Mobile Web Client users can view screens, views and data as usual, but they cannot make changes. Mobile Web Client users can synchronize, but can only download files; they cannot send, receive, or apply database changes. Read-only mode prevents users from corrupting data with an outdated or invalid Siebel application configuration.

Both Developer and Mobile Web Client users exit read-only mode when the Siebel application is restarted following a successful upgrade. At that point, a Mobile Web Client user can perform a full synchronization, sending to the Siebel Server any local database changes that predate the upgrade.

## Retrieving Required Upgrade Kits for Mobile Web Clients

If a user starts the Mobile Web Client in read-only mode, the user is prompted to upgrade at each synchronization session.

The version check for Mobile Web Clients occurs during synchronization. If the version check detects the need for an upgrade, the user receives a prompt indicating a possible upgrade. The remote user logs in again to synchronize the server and local database.

### *To retrieve required upgrade kits for Mobile Web Clients*

- 1 Invoke the Mobile Web Client.
- 2 From the application-level menu, select File > Synchronize > Database.  
A dialog box appears indicating there is a required upgrade available.
- 3 From the dialog box, click Yes and follow the prompts to invoke the Upgrade Wizard.  
The Upgrade Wizard will shut down the Siebel client, install the upgrade kit, and automatically restart the client unless there is an error while installing the kit.
- 4 Log in, and from the application-level menu, select Navigate > Site Map > User Preferences > Component Upgrades.
- 5 Verify that the Status and Current Version columns contain the updated information.

## Retrieving Required Upgrade Kits for Developer Web Clients

Retrieving Developer Web Clients is similar to that of Mobile Web Clients, except version checking only occurs during login.

### *To retrieve required upgrade kits for Developer Web Clients*

- 1 Invoke a Developer Web Client.  
Logging on will trigger a version check and will prompt for retrieval of a required upgrade kit. If you choose not to upgrade at this time, you can only operate in read-only mode.
- 2 Click Yes.  
The Siebel client will download the appropriate upgrade kits.  
After the retrieval is complete, it will invoke the Siebel Upgrade Wizard before exiting.  
After the client exits, the Siebel Upgrade Wizard takes over and applies the upgrade kits and automatically restarts the client when it is done.
- 3 Log in, and from the application-level menu, select Navigate > Site Map > User Preferences > Component Upgrades.
- 4 In the Component Upgrades list, verify that Status is Version OK.

## Launching the Upgrade Wizard

Subscribers use the Siebel Anywhere Upgrade Wizard to install upgrade kits after they are retrieved. This task is a step in [Process of Retrieving, Installing, and Testing Upgrade Kits on page 105](#).

The process for invoking the Upgrade Wizard varies by subscriber type:

- For Siebel Server subscribers, the Upgrade Wizard is launched manually (see [“Retrieving Upgrade Kits for Siebel Regional Node Servers” on page 111](#)).
- For client subscribers, the Upgrade Wizard is launched automatically after an upgrade is retrieved.

The wizard automatically shuts down the Siebel client. It then displays a progress page as it completes the upgrade kit items for the upgrade kits that were retrieved.

**NOTE:** The items displayed on the Upgrade Wizard page vary, depending on the number and type of kits.

When the Upgrade Wizard has successfully installed the upgrade kit, the Siebel client is automatically restarted.

## About Client Upgrade Error Recovery

Before beginning the upgrade process, the Upgrade Wizard makes a backup copy of every file affected by the upgrade. If errors are encountered in the upgrade process, the Upgrade Wizard stops processing. A message indicates that the Upgrade Wizard did not complete successfully and that you should examine the log file for details. It also indicates to restart the Siebel application and attempt the upgrade again.

The next time the user restarts the Siebel client, a prompt appears that states there had been an upgrade in process.

After the user clicks OK, another prompt appears where the user can cancel the upgrade or restart the upgrade from the point of failure. Canceling the upgrade will result in a cleanup. The user will not be able to restart this upgrade.

The Siebel Upgrade Wizard functionality includes error recovery for upgrades on individual files, such as database schema extensions, client configurations or SRF files. However, rolling back upgrades always requires manual intervention.

## Retrieving Upgrade Kits for Siebel Regional Node Servers

Upgrade kits for Siebel Regional Node Server configurations are retrieved automatically by the Siebel Server. These upgrade kits may include a database schema or user-defined files to be installed with version tracking.

For regional databases, the Siebel administrator should be especially careful to distribute the database schema kit under the correct configuration. The default configuration that ships with Siebel Business Applications is Siebel Regional Server. You can create your own configuration for Regional Node Servers to suit the particular needs of your organization.

### ***To set the correct configuration for Regional Node Servers***

- 1** From the application-level menu, select **Navigate > Site Map > Administration - Server Management > Servers**.
- 2** From the Siebel Servers list, select the appropriate server, and click on the **Server Parameters** tab.  
The Server Parameters list appears.
- 3** Modify the parameters as necessary.  
Modify the server parameter **Upgrade Component** to your configuration name. This configuration name can be **Siebel Regional Server** or the name of your custom configuration.

### ***To set the correct configuration for Regional Node Servers using command line***

- From the `srvrmgr` command line, enter:

```
srvrmgr.exe /g gateway_srvr_host /e enterprise /s server_name /u user_name /p password
```

```
change param UpgComponent="Siebel Regional Server" or "name of your custom configuration"
```

Should the version check recognize a required upgrade, the Siebel Server fails to start and logs an error message in the `siebsrvr.log` or `repagent_$taskid.log` similar to the following:

Please contact HQ administrator. An upgrade may be required.

With the Siebel Regional Node Server and all tasks stopped, the Siebel administrator must manually invoke the Upgrade Wizard.

### ***To manually invoke the Upgrade Wizard on Windows***

- 1** Open a command prompt window on the Windows server where the Siebel Regional Node Server is installed.
- 2** Navigate to the `SIEBSRVR_ROOT\BIN` directory, where `SIEBSRVR_ROOT` is the root directory of the Siebel Regional Node Server installation.
- 3** Copy the file `siebug.exe` to another directory, and navigate to that directory.
- 4** Execute the copy of `siebug.exe` to launch the Upgrade Wizard.  
Follow the prompts on the wizard pages.
- 5** Restart the Siebel Regional Node Server.  
See *Siebel System Administration Guide* for details.



Repeat the process from [Step 1](#) on any other Siebel Regional Node Server as appropriate.

### *To manually invoke the Upgrade Wizard on UNIX*

- 1 Open a shell on the UNIX server where the Siebel Regional Node Server is installed.
- 2 Navigate to the `SIEBSRVR_ROOT/BIN` directory, where `SIEBSRVR_ROOT` is the root directory of the Siebel Regional Node Server installation.
- 3 Execute the file `svrupgwi z` to launch the Upgrade Wizard.  
Follow the prompts on the wizard pages.
- 4 Restart the Siebel Regional Node Server.  
See *Siebel System Administration Guide* for details.

Repeat the process from [Step 1](#) on any other Siebel Regional Node Server as appropriate.

Should the Upgrade Wizard fail, there will be an error message to that effect.

You should exit the Upgrade Wizard, correct the condition that caused the error, and restart the Upgrade Wizard using the *copy* of the `siebugg.exe` file. Upon restarting, Upgrade Wizard prompts you either to cancel the upgrade, restoring the Siebel Regional Node Server to its preupgrade state (if possible), or to retry the upgrade from the point of failure. You will need to restart the Siebel Regional Node Server to initiate a version check to restart the upgrade process. If you want to restart, you should choose `Retry`.

The Siebel Regional Node Server will be down until you successfully recover from the failed upgrade. This downtime can affect many users.

After a successful installation of an upgrade, it will take manual intervention to roll it back.

## Testing Upgrade Kits

This task is a step in [Process of Retrieving, Installing, and Testing Upgrade Kits on page 105](#).

The recommended method for testing kits is to create and distribute a kit to a limited and controlled number of users. In this method, you can assign specific testing users to the Test Client configuration so your regular subscribers will not be allowed to install any kits in the testing phase. Distribute the kit only to the Siebel Test Client configuration. See [“Using the Siebel Test Client Configuration for Testing” on page 114](#).

After distribution to the Test Client configuration, test users should retrieve and install the upgrade kit. Test users should then run the application that has been upgraded, and report any unexpected behavior in the application, particularly in areas that have changed due to the upgrade.

**CAUTION:** When testing new upgrade kits, it is strongly recommended that you perform tests using both Developer and Mobile Web Clients. If for any reason Mobile Web Client users experience problems installing or downloading the new kit, the problems will be significant.

After a kit has been thoroughly tested under the Siebel Test Client configuration, you can then distribute it to the appropriate production configurations.

To optimize the setup process for new users, it is recommended that you create a “master installation image” for each configuration your Siebel Anywhere subscribers use, and keep these images up to date. If all current upgrade kits are installed to create a current image, you can use the image to set up new subscribers, so each new subscriber will not need to download and install many upgrade kits. This method also makes it safer to deactivate old upgrade kits, since new subscribers will receive the necessary software by means of installation images, rather than being required to install old kits.

**NOTE:** Siebel Anywhere does not deliver database schema upgrades to individual Developer Web Client users, because they connect to shared databases. Database extensions should be tested in the development environment.

### Using the Siebel Test Client Configuration for Testing

Siebel Anywhere provides a special configuration called Siebel Test Client to serve as your test user. By default, the Siebel Test Client configuration is representative of the other subscriber configurations initially provided, but it can be modified to reflect your active configuration.

#### *To test an upgrade kit*

- 1 Create a user and assign this user with the Siebel Test Client configuration in either of the following ways:
  - Assign the user to the configuration, as described in [“Assigning Employees to a Configuration” on page 41](#).
  - Install a client and set ComponentName = Siebel Test Client in the active CFG file.
- 2 Define, activate and apply the kit as an optional kit and distribute to the appropriate configuration.
- 3 Verify that the Siebel Test Client configuration contains the components being upgraded with the kit.
- 4 Notify the selected users assigned to the Siebel Test Client configuration to log on. Instruct them to use the procedures described in [“Retrieving and Installing Upgrade Kits” on page 106](#) to retrieve and install the upgrade kit.
- 5 After test users install the upgrade kit, have them test it by running the application that was upgraded.

Have all test users report any unexpected behavior in the application. Special attention should be paid to areas of the application that changed due to the upgrade.

If the test is unsuccessful, see [Appendix A, “Troubleshooting for Siebel Anywhere,”](#) for possible solutions.

**NOTE:** If you want to change your optional kit to a required kit after testing, see [“Converting an Optional Kit to a Required Kit” on page 98](#).

## Uninstalling Siebel Maintenance Releases or Patches

Occasionally, as part of testing, troubleshooting, or as preparation before installing a new release, you may want to uninstall the client files associated with a Siebel Maintenance Release or patch. Depending on what you want to accomplish, different techniques are required for successful uninstallation. The following paragraphs describe these variations.

### Uninstalling a Patch Before Installing a Complete Release

Before installing the client files for a complete Siebel Release, such as version 7.8, it is appropriate to uninstall the client files for the previous base release and all Maintenance Releases and patches that have been applied to that base release. You can do this by selecting "Uninstall previous Siebel Version" when you create the Siebel Client Executables upgrade kit for the new release. For more information about creating a Siebel Client Executables kit, see ["Defining a Siebel Client Executables Upgrade Kit" on page 70](#).

### Uninstalling a Patch but Retaining Previous Patches and Base Release

As part of testing or troubleshooting, sometimes you may want to uninstall a single Maintenance Release or patch, while leaving previous Maintenance Releases or patches and the underlying base release intact. For example, you might want to remove version 7.7.2, but retain version 7.7.0 and version 7.7.1.

Before uninstalling a single Maintenance Release or patch, you must determine whether the Maintenance Release or patch that you want to remove was installed as a standard patch or a delta patch:

- A *standard patch* meets any one of the following criteria:
  - The version of the software is earlier than version 7.7.2, *or*
  - The software was installed *directly* from a Siebel software distribution, *or*
  - The software was installed using a Siebel Anywhere upgrade kit of type Siebel Client Executables with Select Type of Installation set to Standard Install.
- A *delta patch* is software that was installed using a Siebel Anywhere upgrade kit of type Siebel Client Executables with Select Type of Installation set to Delta Install.

If necessary, consult your Siebel administrator for information about how the software was installed. Then use the appropriate uninstallation method described in the following paragraphs.

The following procedure describes how to uninstall a standard patch.

#### ***To remove a standard patch but retain previous patches and base release***

- Use the Windows Add/Remove Programs Control Panel to select and uninstall the standard patch.

The following procedure describes how to uninstall a delta patch.

***To remove a delta patch but retain previous patches and base release***

- 1 Locate the following directories in the existing client installation that contains the software you want to uninstall:
  - *\$SiebelRoot*\PATCHBACKUP\_BASE, where *\$SiebelRoot* is the root directory of the client installation. (For example, the directory might be called D:\seaw\PATCHBACKUP\_BASE.)
  - *\$SiebelRoot*\PATCHBACKUP\_*\$Language*, where *\$SiebelRoot* is the root directory of the client installation and *\$Language* is the code for each language that is used in the client installation. (For example, the directory might be called D:\seaw\PATCHBACKUP\_ENU for United States English.)
- 2 In each of the directories that you located in [Step 1](#), rename the file unpatch.bak to unpatch.bat, and then execute unpatch.bat.

**NOTE:** Note: The unpatch.bat files can be executed from the various directories in any order.

# 7

## Supplementary Information for Specific Upgrade Types

This chapter provides supplementary information for specific types of upgrade kits and other special circumstances in which Siebel Anywhere may be used. It includes the following topics:

- [Distributing a Siebel Maintenance Release or Patch on page 117](#)
- [Distributing Additional Language Packs on page 118](#)
- [Example of Global Deployment with Siebel Anywhere on page 119](#)
- [Example of Creating Multiple Kits of the Same Type on page 122](#)
- [Performing Database Schema Updates on page 123](#)
- [Deleting Old Upgrade Kits on page 131](#)
- [Testing for Upgrade Kit Component Consistency on page 132](#)
- [Creating Kits That Check DLL File Versions on page 133](#)
- [Example of Constructing a Third-Party Upgrade Kit on page 134](#)

### Distributing a Siebel Maintenance Release or Patch

This section provides an overview of the process of deploying a Siebel maintenance release or patch, including links to more specific instructions.

**CAUTION:** Do *not* use *Packager* for Siebel maintenance releases. *Packager* is designed for use with full releases. It does not include the necessary functionality to install a Siebel release that depends on the existence of a previous release.

**NOTE:** Only Mobile and Developer Web Clients can use Siebel Anywhere to download the kits needed for a Siebel maintenance release or patch. For other types of clients, such as Siebel Tools clients, users can only receive notification and then manually apply the patch. For information about applying patches manually, see the *Maintenance Release Guide* for the patch, which is available on Siebel SupportWeb.

The process of distributing a Siebel maintenance release or patch consists of the following steps:

- 1 Determine your upgrade requirements. For instructions concerning this step, see [“Determining Upgrade Requirements” on page 26](#).
- 2 Prepare any needed infrastructure elements, such as configurations. For instructions concerning this step, see [“Creating Needed Infrastructure Elements” on page 38](#).

- 3 Prepare Siebel maintenance release files or patch files for inclusion in upgrade kits. Check the release-specific documentation for any special requirements concerning this step.

**NOTE:** In this step, it is important to keep the language-independent base files and the files specific to each language in separate directories. If you want to create a Delta Install upgrade kit, rather than a Standard Install upgrade kit, this is also the point at which you create the delta patch file. For information about Delta Install upgrade kits, see [“About Reducing Siebel Client Executables Kit Size” on page 51](#) and [“Process of Creating a Delta Install Siebel Client Executables Upgrade Kit” on page 52](#).

- 4 Define the following upgrade kits:

**NOTE:** Although the following kits can be created in any order, it is important to wait for the status of each kit to become Pending before defining the next kit. Press ALT+ENTER to refresh the view. You may need to refresh the view several times before you will see the Status change.

- a Define an upgrade kit for the language-independent (base) portion of the maintenance release or patch. For instructions concerning this step, see [“Defining a Siebel Client Executables Upgrade Kit” on page 70](#).
- b For each language that you use in your Siebel implementation, define an upgrade kit for the language-specific part of the maintenance release or patch. For instructions concerning this step, see [“Defining a Siebel Client Executables Upgrade Kit” on page 70](#).

**NOTE:** In Step 4, you define each upgrade kit to execute a different install.exe file.

- 5 Make each language-specific upgrade kit dependent upon the base upgrade kit. For instructions concerning this step, see [“Controlling the Order of Kit Installation” on page 94](#).
- 6 Activate each base and language-specific kit you created. For instructions concerning this step, see [“Activating an Upgrade Kit” on page 93](#).
- 7 Apply each base and language-specific kit you created. For instructions concerning this step, see [“Applying an Upgrade Kit” on page 96](#).
- 8 Distribute the base and language-specific kits to a test configuration. For instructions concerning this step, see [“Distributing Upgrade Kits” on page 98](#).
- 9 Test the upgrade kits by retrieving and installing them as both local and remote members of the test configuration, and by running the software that has been upgraded. For instructions concerning this step, see [Chapter 6, “Retrieving, Installing, and Testing Upgrade Kits.”](#)
- 10 After each base and language-specific kit is operating correctly for test configuration members, distribute the kit to one or more additional configurations, for general use.

## Distributing Additional Language Packs

When you need to deliver additional language packs to Mobile or Developer Web Clients, follow these steps:

- System administrator installs the new language packs and tests these thoroughly before distributing.
- For a major release, you can use Siebel Packager to package the language packs to be distributed. Do not use Siebel Packager to deliver a Siebel Maintenance Release or patch.

- For each language pack, create Siebel Client Executables\_*[language-code]* upgrade kits using the package created in the previous step.
- Distribute the upgrade kits to the appropriate configurations. You must have separate configurations for each combination of languages for your users.

## Example of Global Deployment with Siebel Anywhere

This section provides an example of a global deployment using Siebel Anywhere. The objective is to allow users to access and use the same application in different languages.

You have upgrade components for the languages you installed with your database server. You will map these to the applicable configurations. These configurations will be based upon the languages that different sets of users require to access the same application.

### Assumptions for Global Deployment Example

This example makes the following assumptions:

- You have five sets of users for five different languages.
- You will base your application on Siebel Call Center.
- You will have five SRF files compiled, one for one each language.
- You will have customer revisions (reports for example) depending on the language.
- You will use the same database.
- Each user only uses the application in one language.

### Requirements for Global Deployment Example

You will need:

- Five components for the Siebel Repository Files (SRF)
- Five components for the CFG files
- Five components for Client Customer Revisions
- Five configurations to distribute the upgrade kits
- Six components for Client Executables—one for Client Executables (base), and five for Client Executables\_*\$Language* (language pack)

The language pack component has same locate method, locate information, and version method as the base Client Executables component, but with different Version Information:  
`bin\sslcver.dll,VersionString,$Language`

## Global Deployment Example Records

For the records available, use the values in [Table 28](#), [Table 29](#), [Table 30](#), and [Table 31](#).

**NOTE:** Other language codes can be substituted for ENU in these example records.

Table 28. Example CFG Record in the Upgrade Component List View Files

Field	Comment
Name	CFG ENU
Component Type	Siebel CFG File
Min Version	100
Max Version	Should be blank
Locate Method	CFG
Locate Information	Siebel,ClientRootDir,bin\Language\uagent.cfg
Version Method	CFG
Version Information	Siebel,Version

Table 29. Example SRF Record in the Upgrade Component List View

Field	Comment
Name	Siebel Client SRF ENU
Component Type	Siebel Repository File
Min Version	Should be blank
Max Version	Should be blank
Locate Method	CFG
Locate Information	Siebel,ClientRootDir,objects\Language
Version Method	SRF
Version Information	Siebel,RepositoryFile

Table 30. Example Client Customer Revision Record in the Upgrade Component List View

Field	Entry
Name	Customer Revision ENU
Component Type	Customer Revision
Min Version	Should be blank
Max Version	Should be blank



Table 30. Example Client Customer Revision Record in the Upgrade Component List View

Field	Entry
Locate Method	CFG
Locate Information	Siebel,ClientRootDir,bin\clntrev.cfg
Locate Information - language dependent component	Siebel,ClientRootDir,bin\Language\clntrev.cfg
Version Method	CFG
Version Information	Siebel,Version

Table 31. Example Client Configuration Record in the Siebel Anywhere Upgrade Configurations View

Field	Comment
Name	Siebel Call Center ENU
Configuration Type	Siebel Client Configuration
In the related components applet, make sure that the following components are included:	
Siebel Call Center CFG_ENU	Required
Siebel Client Customer Revisions	Required
Siebel Client Customer Revisions_ENU	Required
Siebel Client Executables	Required
Siebel Client Executables_ENU	Required
Siebel Client Repository File_ENU	Required
Siebel Database Schema	Required
Siebel Upgrade Wizard	Required
Third Party - File Attachment	Not Required

**NOTE:** You must create a configuration for each language combination. For example, if a user installed both English (ENU) and French (FRA) on a system, create a configuration with the components related to these two languages. For more information regarding multilingual deployments, see *Siebel Global Deployment Guide*.

## Example of Creating Multiple Kits of the Same Type

If you need to distribute multiple files to different directories, you should create a kit of the appropriate type Client Customer Revision for each destination directory. You should set the values of the Minimum Old Version and Maximum Old Version settings of the upgrade kits so that they create an unbroken sequence.

The following example uses the Customer Revision kit type. Keep in mind that the same logic can be used for other types of kits when there is a need to distribute multiple kits of the same type.

Create the first kit as usual (for example, Minimum Old Version and Maximum Old Version should be set to blank, New Version = 1).

For the next kit set Minimum Old Version = 1, Maximum Old Version = 1, New Version = 2, and so forth.

If you have to skip a version number for some reason, then set the Minimum Old Version and Maximum Old Version values appropriately. Always set the Minimum Old Version so that it matches the New Version of the previous active kit.

In this example, you need three kits, but Kit 2 was skipped because it contained one wrong file. The version information would be as shown in [Table 32](#).

Table 32. Example of Version Settings for Multiple Kits with One Skipped Kit

Kit	Minimum Old Version	Maximum Old Version	New Version
Kit 1	Null	Null	1
Kit 2 (This kit contains one wrong file, was never activated, and should be skipped.)	1	1	2
Kit 3 (This kit replaces Kit 2.)	1	2	3
Kit 4	3	3	4

For more information about the version numbers used by Siebel Anywhere, see [“How Siebel Anywhere Versions Work”](#) on page 13.

# Performing Database Schema Updates

This section provides special instructions for updating a Siebel database schema using Siebel Anywhere. Typical database schema changes include:

- Adding extension columns to Siebel base tables

**NOTE:** Due to a technical limitation in the Sybase Adaptive Server Anywhere product, you cannot use Siebel Database Schema Upgrade Kits to add required, non-null extension columns to existing local database tables. If you make this kind of database change, you must deploy it by reextracting your Mobile Web Client users, rather than using a schema upgrade kit.

- Adding extension tables to the Siebel database
- Creating or modifying indexes

This section includes the following subsections:

- [“Process of Updating a Siebel Database Schema” on page 123](#)
- [“About What Happens During a Database Schema Upgrade” on page 124](#)
- [“About Changing the Database Schema in a Development Environment” on page 125](#)
- [“About Moving Database Schema Changes Between Environments” on page 125](#)
- [“About Creating Needed Upgrade Kits for Database Schema Changes” on page 125](#)
- [“About Preparing Mobile Web Clients for Database Schema Changes” on page 126](#)
- [“Process of Preparing Regional Node Servers for Database Schema Changes” on page 127](#)
- [“Checking Regional Node Server Parameters Before a Database Schema Update” on page 127](#)
- [“About Installing Database Schema Changes on Mobile Web Clients” on page 129](#)
- [“About Installing Database Schema Changes on Regional Node Servers” on page 129](#)

## Process of Updating a Siebel Database Schema

The overall process for using Siebel Anywhere to update a Siebel database schema is as follows:

- 1 Make and test database schema changes in development environment, using Siebel Tools. For more information about this step, see [“About Changing the Database Schema in a Development Environment” on page 125](#).
- 2 Move changes to test environment, using Siebel Configuration Utility. For more information about this step, see [“About Moving Database Schema Changes Between Environments” on page 125](#).
- 3 Create needed upgrade kits in test environment. For more information about this step, see [“About Creating Needed Upgrade Kits for Database Schema Changes” on page 125](#).
- 4 Prepare test Mobile Web Client users and Regional Node Servers for changes. For more information about this step, see the following topics:
  - [“About Preparing Mobile Web Clients for Database Schema Changes” on page 126](#)
  - [“Process of Preparing Regional Node Servers for Database Schema Changes” on page 127](#)

- 5 Install changes on test Mobile Web Client users and Regional Node Servers. For more information about this step, see the following topics:
  - [“About Installing Database Schema Changes on Mobile Web Clients” on page 129](#)
  - [“About Installing Database Schema Changes on Regional Node Servers” on page 129](#)
- 6 Move changes to production environment, using Siebel Configuration Utility. For more information about this step, see [“About Moving Database Schema Changes Between Environments” on page 125](#).
- 7 Create needed upgrade kits in production environment. For more information about this step, see [“About Creating Needed Upgrade Kits for Database Schema Changes” on page 125](#).
- 8 Prepare production Mobile Web Clients and Regional Node Server for changes. For more information about this step, see the following topics:
  - [“About Preparing Mobile Web Clients for Database Schema Changes” on page 126](#)
  - [“Process of Preparing Regional Node Servers for Database Schema Changes” on page 127](#)
- 9 Install changes on production Mobile Web Clients and Regional Node Server. For more information about this step, see the following topics:
  - [“About Installing Database Schema Changes on Mobile Web Clients” on page 129](#)
  - [“About Installing Database Schema Changes on Regional Node Servers” on page 129](#)

## About What Happens During a Database Schema Upgrade

This section provides background information about what happens during a database schema upgrade.

During database schema upgrades, new and changed objects are identified by comparing the current physical schema with the new virtual schema. Based on this comparison, Upgrade Wizard adds new objects and alters existing objects by using the following:

- To create the new objects: create table, create index, and so on.
- To alter the existing objects: alter table, alter index, and so on.

**CAUTION:** Downsizing existing columns is not recommended. Do not drop columns or tables. These operations can cause transactions that were created before the schema change to be rejected, because data that was included in the earlier transactions may not fit into the new database schema.

Normally, the database does not allow downsizing a column that already contains data that could not fit into a new smaller column. If you downsize a column, the column representation in the repository would be different from the physical schema. Under these conditions, Siebel Remote would fail, among other possible errors.

For example, after downsizing existing columns, one of the columns in the new database is shorter (it is created from the representation in the repository). Then, a local database is reextracted (this includes a new schema). After the Remote user synchronizes, initialization occurs. Upgrade Wizard first applies the new local database (including a new schema) to a Remote user's machine. Next, Upgrade Wizard applies any server transactions downloaded during synchronization. If these transactions were generated before the schema changed, Upgrade Wizard tries to insert data that does not fit into the new column, causing the transaction to be rejected. Furthermore, if this occurs on a regional node, the error may cause an outage.

## About Changing the Database Schema in a Development Environment

The first step in deploying Siebel database schema changes is to make the schema changes in a development environment, using Siebel Tools. For information about this step, see *Configuring Siebel Business Applications*.

Keep track of whether your changes include creation or modification of views or applets. This information will affect later steps in the deployment.

Test the changes in the development environment before proceeding to the next part of the process. For links to other steps in this process, see ["Process of Updating a Siebel Database Schema" on page 123](#).

## About Moving Database Schema Changes Between Environments

When you are ready to move database schema changes from a developmental environment to a test environment, or from a test environment to a production environment, do *not* use Siebel Tools to make the changes. Instead, use the Migrate Repository option in the Siebel Configuration Utility to migrate the entire Siebel Repository containing the schema changes to the new environment.

For information about this step, see *Going Live with Siebel Business Applications*.

Test the changes in the new environment before proceeding to the next part of the process. For links to other steps in this process, see ["Process of Updating a Siebel Database Schema" on page 123](#).

## About Creating Needed Upgrade Kits for Database Schema Changes

The creation of a database schema kit has two consequences beyond the creation of the kit itself:

- Schema changes from the logical schema (in Siebel Tools) are automatically applied to the physical schema of the HQ database.
- The custom schema version number in the HQ database is automatically incremented by one.

The number and type of upgrade kits you need for deploying database schema changes depend on the kinds of changes you made to the schema:

- **Major schema changes.** Siebel Anywhere is suitable for distributing *minor* database schema updates that have been created using database extensibility. If you are performing a major upgrade (such as migrating from Siebel 6 to Siebel 7), or if you are migrating from Siebel 7.0.x to 7.5.x, do *not* use Siebel Anywhere to distribute your changes.

For information about performing major upgrades, see the *Siebel Database Upgrade Guide*. For information about extracting database changes for Mobile Web Clients, see *Siebel Remote and Replication Manager Administration Guide*, and for detailed information regarding database extensibility, see *Configuring Siebel Business Applications*.

- **Incremental schema changes.** If your database schema changes are *incremental* (that is, if you have added tables, added columns, or changed the names of existing tables or columns), you can create a Siebel Anywhere upgrade kit to upgrade Mobile Web Clients or a Siebel Regional Node Server. You do not need to reextract.
- **Preference for reextraction rather than Siebel Anywhere.** It is possible to reextract databases for Mobile Web Clients and Regional Databases, rather than updating them using Siebel Anywhere, if you prefer.
- **Mandatory reextraction for nonincremental schema changes.** If your database schema changes are *nonincremental* (that is, if you have removed or deactivated tables or columns, which are changes that are *not* recommended), then you *must* reextract all Mobile Web Clients and Regional Node Servers. Otherwise, synchronization might fail due to transactions that contain data for columns or tables that are no longer available. Reextraction makes it unnecessary to create a database schema upgrade kit.

After you define the kit or kits for your upgrade, you need to activate and apply them as described in [Chapter 5, “Activating, Applying, and Distributing Upgrade Kits.”](#) If you created both a Siebel Database Schema kit and a Siebel Repository File kit, make sure both are activated and applied before you proceed to the next part of the process.

For links to other steps in this process, see [“Process of Updating a Siebel Database Schema” on page 123.](#)

**CAUTION:** Do not make any kits dependent upon the database schema component. If you create this type of dependency, you will receive an error when first time Mobile Web Client users initialize their local databases. The error will indicate the system was unable to check initialization status.

## About Preparing Mobile Web Clients for Database Schema Changes

In either a test environment or a production environment, it is recommended that you have your Mobile Web Client users synchronize before you distribute your Siebel Database Schema upgrade kit.

**NOTE:** If any of your mobile users connect to a Regional Node Server, rather than a HQ server, the Regional Node Server must install the database schema changes before those mobile users can obtain and install the changes.

After Mobile Web Client users have synchronized, you can distribute your active upgrade kits to an appropriate test or production configuration for your Mobile Web Clients. For information about distributing kits, see [“Distributing Upgrade Kits” on page 98](#).

For links to other steps in this process, see [“Process of Updating a Siebel Database Schema” on page 123](#).

## Process of Preparing Regional Node Servers for Database Schema Changes

Before you distribute your Siebel Database Schema upgrade kit to either test Regional Node Servers or production Regional Node Servers, take the following steps:

- 1 Have your Regional Node Servers synchronize with the HQ server.
- 2 Determine which configuration you will use to distribute the upgrade kit or kits to your Regional Node Servers.

The default configuration that ships with Siebel Business Applications is Siebel Regional Server. You can create your own configuration for Regional Node Servers to suit the particular needs of your organization. Make sure the Siebel Database Schema component is associated with your custom configuration. For information on these tasks, see [“Identifying Configurations to Deliver Upgrade Components” on page 28](#), and [“Creating Needed Infrastructure Elements” on page 38](#).

- 3 Inspect the Upgrade Component server parameter value for each Regional Node Server, and change it if necessary. For instructions, see [“Checking Regional Node Server Parameters Before a Database Schema Update” on page 127](#).

- 4 Distribute your upgrade kit (or kits) to a configuration for your Regional Node Servers.

These may be either test Regional Node Servers, or production Regional Node Servers, depending on where you are in the process. For information about distributing kits, see [“Distributing Upgrade Kits” on page 98](#).

For links to other steps in the process of updating a database schema, see [“Process of Updating a Siebel Database Schema” on page 123](#).

## Checking Regional Node Server Parameters Before a Database Schema Update

This section describes how to check Regional Node Server parameters before you distribute a database schema update kit to a Regional Node Server.

Before you distribute a database schema to a Regional Node Server, you must make sure that the server parameter Upgrade Component on the Regional Node Server is set to the correct value to match either Siebel Regional Server or any custom configuration you have made for your Regional Node Server. The simplest way to modify the parameter is to use the Server Parameters view, as described in the following procedure.

**CAUTION:** Before installing a database schema upgrade kit on a Regional Node Server in a DB2 environment, you must drop all customized views and triggers in the regional database. Otherwise, the Siebel Upgrade Wizard will encounter errors when applying the changes in the logical schema to the physical schema.

### ***To set Regional Node Server parameters before distributing a database schema update***

- 1 On the Regional Node Server, from the application-level menu, select **Navigate > Site Map > Administration - Server Configuration**.
- 2 From the link bar, select **Servers**.  
The Siebel Servers list appears.
- 3 From the Siebel Servers list, select the appropriate server, and click the **Parameters** tab.  
The Server Parameters list and the Additional Parameter Information form appear.
- 4 In the Server Parameters list, select **Upgrade Component** in the Parameter field, and set Value to Siebel Regional Server or the name of your custom configuration.  
The Value on Restart setting is automatically populated with the same value when you leave the Value field.
- 5 Still in the Server Parameters list, select **Table Owner Password** in the Parameter field, and make sure that Value and Value on Restart are not blank.
- 6 Still in the Server Parameters list, select **Version Check** in the Parameter field, and set Value and Value on Restart to True.

**NOTE:** You can also check and modify the Upgrade Component, Table Owner Password, and Version Check parameters in Server Manager.

You can also check and modify Regional Node Server parameters using Server Manager, as described in the following two procedures.

### ***To check server parameters using Server Manager***

- From the command-prompt, enter:  

```
list param <UpgComponent, VersionCheck, TableOwnPass>
```

### ***To modify server parameters using Server Manager***

- Enter:



```
change param param_alias=parameter_value
```

Where *param\_alias* is UpgComponent, VersionCheck or TableOwnPass.

For information about other steps in the process of preparing a Regional Node Server for a database schema update, see [“Process of Preparing Regional Node Servers for Database Schema Changes” on page 127](#).

For links to other steps in the process of updating a database schema, see [“Process of Updating a Siebel Database Schema” on page 123](#).

## About Installing Database Schema Changes on Mobile Web Clients

In a test environment, install the Siebel Database Schema upgrade kit (and any accompanying kits) on test Mobile Web Clients. In a production environment, install the Siebel Database Schema upgrade kit (and any accompanying kits) on production Mobile Web Clients. For instructions, see [Chapter 6, “Retrieving, Installing, and Testing Upgrade Kits.”](#)

**NOTE:** The Database Schema kit will only be downloaded by Mobile Web Client users.

Installing the schema upgrade kit synchronizes the logical and physical schemas on Mobile Web Client databases.

### *To distribute a kit for Mobile Web Clients*

- For Mobile Web Clients assigned to the HQ Server, distribute the kit under the appropriate configurations so each Mobile Web Client can receive the kit during synchronization.
- For Mobile Web Clients assigned to a Regional Node Server, distribute the database schema update to the configuration for the Regional Node Server, and install the update on the Regional Node Server before distributing the kit to the configuration for that server’s Mobile Web Clients.

Schema changes must also be complete on a Regional Node Server before any Dedicated Web Clients connected to that server can resume normal operations.

For links to other steps in this process, see [“Process of Updating a Siebel Database Schema” on page 123](#).

## About Installing Database Schema Changes on Regional Node Servers

In a test environment, install the Siebel Database Schema upgrade kit (and any accompanying kits) on test Regional Node Servers. In a production environment, install the Siebel Database Schema upgrade kit (and any accompanying kits) on production Regional Node Servers. The following procedure describes how to install a database schema update on a Regional Node Server.

Installing the schema upgrade kit synchronizes the logical and physical schemas on Regional Node Server databases.

**NOTE:** If any of your mobile Web Client users connect to a Regional Node Server, rather than a HQ server, the Regional Node Server must install the database schema changes before those mobile users can obtain and install the changes.

### *To install a database schema update kit for regional databases*

- 1 Disconnect all users from the Regional Node Servers and the HQ server.
- 2 Stop all tasks on the applications servers except the Replication Agent, Transaction Processor, Router, and Merger.
- 3 Wait until the replication backlog has been processed.  
  
Check the download and upload statistics in the Administration - Siebel Remote screen for each Regional Node. The three Last File and Last Transaction numbers should match in the lower applet. Make sure that there are no outstanding transactions by checking that \docking\inbox and \docking\outbox directories are empty on the Regional Applications Servers.
- 4 Verify the server parameter Version Check is set to True on the Regional Applications Server using one of the following options.
  - In the Server Parameters View, verify that the server parameter Version Check is set to True on the Regional Applications Server.
  - Use the following commands to start srvmgr and verify the value of the Version Check parameter:  

```
srvmgr /g gateway_host /e enterprise_server /s server_name  
/u user_name /p password
```

```
list param VersionCheck
```
- 5 The Replication Agent should now perform a version check and shut down itself and the Regional Applications Server so the Administrator can perform the upgrade manually.
- 6 Check the appropriate log files (RepAgent.log and syncthrd.log) to verify the need to run the Upgrade Wizard.
- 7 Verify that the Siebel Server is properly shut down. If not, shut it down manually.
- 8 Use one of the following methods to run the Upgrade Wizard:
  - In a Windows environment, in the server bin directory, run siebugp.exe to carry out the upgrade.
  - In a UNIX environment, in the server bin directory, run srvrupgwiz to carry out the upgrade.
- 9 After the upgrade, bring up the Regional Siebel Server.
- 10 You can change the server parameter Version Check back to False, which is its default value.

**NOTE:** When you install the Database Schema kit, the schema changes are applied to the server and the custom version is incremented, which makes the database template obsolete. If there are Mobile Web Client users who connect to the Regional Node Server, you must run Generate New Database task to create a new database template.

The Mobile Web Clients should be able to synchronize and perform the Schema Upgrade kit in much the same manner as any other kits by using the Siebel Upgrade Wizard.

If the Siebel Developer created or modified applets or views to display data in new extension columns or new extension tables, it is necessary to create and distribute the accompanying Siebel Repository File upgrade kit. Installing the combination of the Siebel Database Schema upgrade kit and the Siebel Repository File upgrade kit allows users to view these data after the schema upgrade.

After the Database Schema upgrade is finished, distribute the new SRF kit to its corresponding Mobile Web Clients.

For links to other steps in this process, see [“Process of Updating a Siebel Database Schema” on page 123](#).

## Deleting Old Upgrade Kits

Sometimes you may want to purge older upgrade kits to free up some space used by these kits. When you purge old kits, you must make sure that users will not encounter errors when they retrieve kits that are expecting to use information from kits that have been deleted.

Two groups of users can be adversely affected when old kits are deleted, new users and infrequent users. For example, assume that your end users belong to the same configuration. Also assume they are currently using version 3 of Client Customer Revisions and version 4 of SRF.

If you were to delete all the previous Client Customer Revisions and SRF kits, a new user might have trouble with Client Customer Revisions if starting from version 0. This situation occurs because Client Customer Revisions kits often depend on the previous versions. If a new user needs version 2 in order to gain access to version 3, and if version 2 is deleted, Siebel Anywhere would not be able to provide the new user with version 3. However, the new user *would* be able to install version 4 of the SRF, because SRF kits do not depend on any version history.

The correct way to handle this situation is to keep an updated installation for new users where each of the kits has been installed. You can do this by creating a package periodically and making the most current package available to new users. This method addresses the problem for new users, because they always install a client with up-to-date configurations.

However, do *not* delete old kits as soon as a new package is created. There may be infrequent users in your organization who have not logged in for months. While the rest of the company is using version 3, the infrequent users may still be on version 1. In this case, it is up to the individual Siebel administrators to assess the situation and determine how old the kit must be before deletion.

Sometimes you have a few kits of the type Customer Revision that were distributed long ago. Use the following procedure to delete them. Use the same logic if you are planning to delete any other kits.

### ***To delete old Customer Revision kits***

- 1** Establish one installation that includes all the Customer Revision kits ever installed.
- 2** Create a package from that installation.

- 3 Set up each new user with the package prepared in [Step 1](#). This provides new clients with the files that were distributed through the old Customer Revision kits, and also provides new clients with the correct version of the `clntrev.cfg`.

## Testing for Upgrade Kit Component Consistency

After you upgrade a Siebel application to a new release, you should check to make sure that version numbering continues across releases for Upgrade Components in each Siebel Anywhere Configuration. Upgrading to a new release of Siebel Business Applications upgrades the Siebel Anywhere table definitions. However, it does not reset the Upgrade Component information for Siebel Anywhere Configurations.

As a result, Siebel Anywhere will resume version checking against the existing component versions in the new release of Siebel Business Applications. If required upgrade kits for Siebel Client Repository File, Siebel Client Customer Revisions, or Siebel CFG File were distributed in the previous release, newly upgraded Production clients may be incorrectly prompted for upgrade kits when they synchronize with the server.

Before upgrading Production users, test the upgrade kit using the Packager Client machine with version checking enabled. Verify that the Packager Client passes version check, so that after the upgrade kit is packaged and distributed, it will function correctly for the remainder of your Production users.

To accomplish this testing, verify that Administrator Client has the latest versions for Siebel Client Repository File\_*[language-code]*, Siebel Client Customer Revisions, and Siebel CFG\_*[language-code]* in advance of running the Siebel Packager utility.

The following procedure describes how to test repository components for consistency.

### *To test repository components for consistency*

- 1 Determine the current version values of your repository components by completing the following substeps:
  - a In your production Siebel environment, navigate to Administration - Siebel Anywhere > Upgrade Component List.
  - b Use standard query techniques to select your server and client repository components.
  - c Determine the current value of your repository components by inspecting their Min Version values. A blank Min Version field indicates a value of zero.
- 2 Use a computer and a login account that have read/write access to the location of the repository file you want to check, and open a command window.
- 3 Make sure that the repository file you want to check is not currently in use.  
You may need to exit from any Siebel application that is using the repository.
- 4 In the command window, navigate to the directory where the `srfstamp` utility is located, as follows:

- If you are using a server computer, navigate to *SIEBEL\_ROOT*\bin, where *SIEBEL\_ROOT* is the root directory for your Siebel Server software.
  - If you are using a client computer, navigate to *SIEBEL\_CLIENT\_ROOT*\bin, where *SIEBEL\_CLIENT\_ROOT* is the root directory for your Siebel Mobile Web Client or Siebel Developer Web Client software.
- 5 Enter the following command, where *repository\_file\_name* is the complete path and file name of your repository file, and *language-code* indicates the language for which this repository is used:

```
srfstamp /r repository_file_name /l language-code
```

After a brief pause, the current version number of the specified repository file is displayed.

- 6 Compare the version displayed by srfstamp to the component version you obtained in [Step 1](#) of this procedure.
- 7 If you want to change the version number of the specified repository file (for example, to use your current repository version number for a new repository file during a major upgrade), enter the following command:

```
srfstamp /w repository_file_name /v new_version_number /l language-code
```

The following procedure describes how to test for Siebel Client Customer Revisions component consistency.

#### **To test Siebel Client Customer Revisions component consistency**

- Copy the clntrev.cfg file from one of your previous release Production Clients to the *SIEBEL\_CLNT\_ROOT*\bin directory of the Administrator Client.

The following procedure describes how to test for Siebel CFG File component consistency.

#### **To test Siebel CFG file component consistency**

- Edit the appropriate CFG file under *SIEBEL\_CLNT\_ROOT*\bin\[*language-code*] and increment the Version parameter in the [Siebel] section to the Maximum Version of the Upgrade Component Siebel Call Center CFG for the previous release.

Test with the Administrator Client until this client passes version check. Run the Siebel Packager Utility to create the custom installer. Distribute the custom installer through Siebel Anywhere (as described in the *Siebel Database Upgrade Guide*), or through a shared network location.

## Creating Kits That Check DLL File Versions

Sometimes you may need to upgrade third-party software based upon the version of a particular DLL file. An example would be an ODBC upgrade on a client, based upon the version of the file *odbcint.dll*.

The following procedure describes how to create a custom component and an upgrade kit that check the version of a particular DLL file.

### To create an upgrade component and kit for DLL version check

- 1 From the application-level menu, select **Navigate > Site Map > Administration - Siebel Anywhere**.
- 2 From the link bar, select **Upgrade Component List**.  
The Upgrade Components list appears.
- 3 Create an Upgrade Component and specify the field values shown in the following table.

Field	Comment
Name	Upgrade Component Name
Component Type	Third-Party Software
Locate Method	Registry
Locate Information	<i>hkey, subkey_name, value_name</i>
Version Method	FileVersion

In this table, the value for **Locate Information** points to the registry hkey, subkey and a specific value that points to the DLL file you wish to version check.

For example:

```
HKEY_LOCAL_MACHINE, SOFTWARE\ODBCFI I eDSN\Default Icon, Driver
```

If the registry value is `DefaultIcon`, then the *value\_name* must be omitted. This value in the registry will then point to the specific DLL file.

For example:

```
C:\WINNT\System32\odbcint.dll
```

- 4 After creating the custom component, as described in the previous steps of this procedure, define the upgrade kit as described in [Chapter 4, "Defining Upgrade Kits."](#)

When defining the Upgrade Kit, remember to select the name of the custom upgrade component (created above) when the Upgrade Kit Wizard page appears.

## Example of Constructing a Third-Party Upgrade Kit

This section provides an example of how to construct a third-party upgrade kit, using Third-Party Win-Zip as the example software for which the kit is created. The process shown in this section can be used to construct upgrade kits for other third-party software applications, provided that the software is related to the use of Siebel Business Applications.

There are two procedures to this process:

- [To create a new upgrade component Third Party - WinZip](#)
- [To define a Third Party - WinZip upgrade kit on page 135](#)

**To create a new upgrade component Third Party - WinZip**

- 1 From the application-level menu, select Navigate > Site Map > Administration - Siebel Anywhere.
- 2 From the link bar, select Upgrade Component List.  
The Upgrade Components list appears.
- 3 On the Upgrade Components list, click New.  
A blank record appears.  
Enter the necessary information.  
As an alternative, copy the record of an existing Third Party component and then edit the fields as required.
- 4 Enter the values shown in the following table.

Field	Comment
Component Name	Enter Third Party - WinZip.
Component Type	From the pick list, choose Third Party Software.
Locate Method	From the pick list, choose Registry.
Locate Information	Enter HKEY_CLASSES_ROOT, WinZip\shell\open\command Locate Info has the following structure when Locate Method is set to Registry: <i>HKEY,Subkey,Name</i> . If <i>Name</i> is equal to the default, that is, (Default) as seen in the Registry Editor, then leave <i>Name</i> as blank in Locate info.
Version method	From the pick list, choose FileVersion.

- 5 From the link bar, select Upgrade Configurations.  
The Upgrade Configurations list appears, along with the Upgrade Components list.
- 6 On the Upgrade Configurations list, select the desired clients such as Siebel Sales Client.
- 7 On the Upgrade Components list, click New.  
The Upgrade Components pick list appears.
- 8 On the Upgrade Components Selection dialog box, select Third Party - WinZip.  
The Upgrade Configurations list appears, along with the Upgrade Components list.  
The new record (Third Party - WinZip upgrade component) is highlighted in the Upgrade Components list, which indicates that it is selected.  
This step completes the procedure to create a new upgrade component, Third Party - WinZip.

**To define a Third Party - WinZip upgrade kit**

- 1 From the application-level menu, select Navigate > Site Map > Administration - Siebel Anywhere.

- 2 From the link bar, select Upgrade Kits.  
The Upgrade Kits view appears.
- 3 Click Auto Create to invoke the Upgrade Kit Wizard.  
The Upgrade Kit Wizard - Upgrade Components page appears.
- 4 Choose Third Party - WinZip in the list and click Next.  
The Upgrade Kit Wizard - Add Files page appears.
- 5 Input the UNC path of the WinZip installer.
- 6 Click Next.  
The Upgrade Kit Wizard - Select Execute File page appears.
- 7 Input the executable filename such as Instal I .exe and the command-line argument if any.
- 8 Follow the prompts to perform the following actions:
  - a Review and verify the information.
  - b Specify the new version number.
  - c (Optional) Change the title, if you want to do so.
- 9 When you have finished entering upgrade kit information, click Finish.

Follow the procedures in [Chapter 5, "Activating, Applying, and Distributing Upgrade Kits."](#) to activate, apply, and distribute the kit.



# A

## Troubleshooting for Siebel Anywhere

This appendix includes tips to help administrators resolve some problems that can arise while working with Siebel Anywhere. It includes the following topics:

- [“Upgrade Kit Wizard Error at Finish” on page 137](#)
- [New Upgrade Kit Status “Request Submitted” on page 137](#)
- [New Upgrade Kit Status “In Progress” on page 139](#)
- [Unable to Download Upgrade Kit on page 139](#)
- [Required Kit Cannot Be Installed on page 141](#)
- [Failed Upgrade Kit Locks Administrator’s Account on page 141](#)
- [Converting a Required Kit to an Optional Kit on page 142](#)
- [Changing Version Settings After Distribution on page 143](#)

**NOTE:** For any troubleshooting that involves Mobile Web Clients, make sure that Transaction Processor and Transaction Router are up and running on the Siebel Server with which the mobile users synchronize. Verify the Status of the process. For more information see *Siebel Remote and Replication Manager Administration Guide*.

### Upgrade Kit Wizard Error at Finish

If you receive an error message such as “Unable to find definition for component UpgKitBldr” when you click Finish in the Upgrade Kit Wizard, you may need to synchronize server components. For more information, see [“Synchronizing Components” on page 59](#).

### New Upgrade Kit Status “Request Submitted”

From time to time, you may notice that the status of a new upgrade kit remains Request Submitted after several minutes of waiting. A status of Request Submitted means that the Upgrade Kit Builder has not yet picked up the server request to build the kit. There are a number of possible causes for this delay:

- The upgrade kit may have been created when the Siebel Anywhere component was not enabled. The status will not change until the component is enabled. After the component is enabled, the request should be processed automatically.
- The server component Upgrade Kit Builder (UpgKitBldr) may be busy building a previously submitted kit. Most kits take only a few minutes to build. However, database schema kits can take as long as two hours or more to build, depending on the database type, size, and the extent of the schema changes.

- The server component Upgrade Kit Builder (UpgKitBldr) may not be running.
- The server component Server Request Broker (SRBroker) may not be running correctly.
- The server component Server Request Processor (SRProc) may not be running correctly.

The following paragraphs provide information about troubleshooting the latter problems.

## Checking for Current Upgrade Kit Builder Activity

The following two procedures provide ways to check whether the Upgrade Kit Builder is busy building another kit:

### *To check Upgrade Kit Builder activity in a Siebel application*

- 1 In your Siebel application, navigate to Administration - Server Management > Components.
- 2 In the Components list, select the Upgrade Kit Builder component.
- 3 Click the Tasks view tab, and inspect the status of any tasks submitted for the Upgrade Kit Builder component.

### *To check Upgrade Kit Builder activity using srvrmgr*

- In `srvrmgr`, enter the following command to check on the status of Upgrade Kit Builder tasks:  
`srvrmgr> list task for comp UpgKitBldr`

## Checking the Status of Components Related to Upgrade Kits

There are three ways to check the status of the UpgKitBldr, SRBroker, and SRProc components:

- In your Siebel application, navigate to Administration - Server Management > Components, select each applicable component, and click the Tasks view tab to inspect the status of tasks submitted for that component.
- In `srvrmgr`, use the following command to view the status of a component, replacing *component\_name* with UpgKitBldr, SRBroker, or SRProc:  
`srvrmgr> list comp component_name`
- In the log directory, check the log files as follows:
  - Inspect the *enterprise\_server\_name.siebel\_server\_name.log* to verify that all the processes for UpgKitBldr, SRBroker, and SRProc were created correctly.
  - See whether there are any signs of trouble in the *SRProc\_task#.log* file or the *SRBroker\_task#.log* file.

## Troubleshooting Server Request Processor Problems

If you find evidence that the SRProc component is not running correctly, check the following aspects of your server's configuration:

- Is database connection information correct?
- Is User Name/Password correct?
- Have you done server component synchronization? For information about this procedure, see *Siebel System Administration Guide*.

For more information about troubleshooting server problems, see *System Monitoring and Diagnostics Guide for Siebel Business Applications*.

## New Upgrade Kit Status “In Progress”

From time to time, you may notice that the status of a new upgrade kit remains In Progress for an extended period, indicating that the upgrade kit is taking a long time to build. If this is the case, consider the following information to help troubleshoot the situation.

Some of the reasons for a delay in status change include:

- **The upgrade kit is still being built.** (In particular, database schema kits can take as long as two hours or more to build.) To monitor the progress of the upgrade kit, you have three options to check the latest status:
  - Navigate to Administration - Server Management > Components > Tasks and look for the latest status of the Upgrade Kit Builder component.
  - In the server manager line-mode utility, enter:
 

```
list task for comp UpgKitBldr
```
  - Look for the component log file for Upgrade Kit Builder (UpgKitBldr\_<taskno>.log) in the log directory.
- **Upgrade Kit Builder is not responding.** Sometimes a Siebel File System error or a network connection error interrupts the operation of Upgrade Kit Builder. If you conclude that Upgrade Kit Builder is not responding, take the following actions:
  - Check two other Server Components that support Upgrade Kit Builder—File System Manager (FSM) and Server Request Broker (SRBroker). Be sure these components are running correctly.
  - Edit the record, change the status from In Progress to Error, save the record, and then delete the record before trying to rebuild. (The reason for deleting the failed record is to avoid a naming conflict.) Then, recreate the upgrade kit.

## Unable to Download Upgrade Kit

There are times when the following error messages appear when attempting to download an upgrade kit.

- Unable to find upgrade kit path to upgrade component: *component\_name*. Please contact your Siebel administrator or try again.
- The file '*upgrade\_kit\_file\_name.arc*' could not be found on any specified file system.

- Upgrade wizard did not complete successfully. Restart Siebel application to attempt the upgrade again.
- Unable to open compressed file  
(*Sublevel\_directory\_path\S\_UPG\_KIT\_IARG\_ROW\_ID\_REV\_NO.saf*).
- Unable to download the Upgrade Kit '*upgrade\_kit\_title*' required to upgrade your system.

These errors can be caused by the following conditions:

- **A previously required kit has been deactivated or deleted.** The client is trying to find an active kit with a certain New Version number, but there is no such kit because it was deactivated or deleted.

To solve this problem, have your Mobile Web Client first synchronize using an invalid Configuration (set the Component Name in the CFG file to none). This action will get the recent transactions routed to the Mobile Web Client and pass the transaction regarding the nonfunctioning or inactivated kit.

Then have users change the Component Name back to the original value and synchronize again. You can make this process transparent to your users by sending them a batch file that synchronizes for them from the command line using a different CFG that was sent to them with the batch file. Use `si_ebsync.exe` and specify the CFG with the `/c` option.

- **A nonexecutable file has been specified for execution.** The kit you have created is of type Siebel Client Customer Revision and when you were creating the kit, you selected a nonexecutable file to be executed. You need to deactivate the kit and recreate it with the same version number. Activate the kit, but do not apply or distribute it.
- **A user's connection to the Siebel File System is not set correctly.** Use one of the following techniques, depending on the type of user involved:
  - **For Developer Web Client users.** Check the Developer Web Client users' CFG files and make sure that the File System parameter under Server section is set to the appropriate location and that your user has read/write access to that directory. One test for correct location and read/write access would be to have the user open an attachment. If the attachment opens, then the connection to the Siebel File System is set correctly.
  - **For Mobile Web Client users.** Because Mobile Web Client users are connected to the Siebel File System through the Siebel Server, make sure the Siebel File System parameter of Synchronization Manager is set to the correct location—the same location as the File System parameter in the CFG file (of the user who has created the kit). In other words, make sure that the File System Manager is working properly.
- **There is not enough space on the machine to download the kit.** Free some space and try to retrieve and install the upgrade kit again.
- **A prerequisite upgrade kit is not available.** An upgrade that the System Administrator intends for the end users to retrieve may depend on another kit that has either been deactivated, deleted, or not created yet. The Siebel Client will make sure that all kits are ready before invoking the Upgrade Wizard.

## Required Kit Cannot Be Installed

If you create and distribute a required upgrade kit that, for some reason, cannot be installed on the client or server, you need to define and distribute a repair kit (a new upgrade kit that functions properly).

Use the Upgrade Kit Wizard to define the repair kit. Make sure the version information in the repair kit differs from the version information in the nonfunctioning kit in the following ways:

- If the minimum version is editable, make sure that its value is set to the last known good version for this upgrade component.

For example, if the minimum version of the nonfunctioning kit was 5, the minimum version of the repair kit should be no higher than 5.

- Make sure that the new version of the repair kit is at least one version number higher than the new version of the nonfunctioning kit.

For example, if the new version of the nonfunctioning kit was 6, make the new version of the repair kit 7. Then, Siebel Anywhere can upgrade the user's setup from the last good version, skipping the nonfunctioning version.

After you have defined your repair kit, activate and test the kit as you would do for any other required kit. Distribute the repair kit to the appropriate Developer or Mobile Web Client users, as described in the following procedure.

### *To distribute a repair kit*

- 1 From the application-level menu, select **Navigate > Site Map > Administration - Siebel Anywhere**.
- 2 From the link bar, select **Upgrade Configurations**.

The Upgrade Configurations list appears.

- 3 Prepare the kit for distribution.
  - a In the Upgrade Configurations list, select the configuration under which you distributed the nonfunctioning kit.
  - b Verify that the components in the repair kit are included in the configuration as related components.
  - c With this configuration selected, click **Distribute**.

## Failed Upgrade Kit Locks Administrator's Account

If the Siebel administrator's account is locked out due to a failed upgrade kit, the following procedure will help you recover.

It is also sometimes possible to log in as a Web Client that does not have version checking or upgrades, and take corrective measures to resolve the issue related to the failed upgrade.

### *To unlock the Siebel administrator*

- 1 Edit the Siebel administrator's CFG file by changing the ComponentName parameter to None.

This change will prevent the Siebel client from performing any version checking. If you dynamically associated the System Administrator account with a configuration, contact Siebel Technical Support for assistance to help resolve this issue.

**CAUTION:** Dynamically associating the System Administrator account with any configuration is *not* recommended, because doing so sometimes prevents the System Administrator from logging in.

- 2 If the upgrade kit failed while the Siebel Upgrade Wizard was running, then delete the file upgwiz.ucf in the \bin directory.

The Siebel client always checks for the existence of the file upgwiz.ucf in the \bin directory to find out if it was in the middle of an upgrade.

- 3 Log in to the system to determine the reason for the failed upgrade, and take corrective measures.

Presumably, the other subscribers to your original configuration are also affected.

- 4 Test your solutions using the Siebel Test Client configuration.

- 5 After testing the upgrade kit thoroughly, distribute the correct upgrade kit to every appropriate configuration.

Remember to change the value of the ComponentName parameter to its original value in the CFG.

## Converting a Required Kit to an Optional Kit

If you distribute a required kit for Developer Web Client users and realize it should have been optional, use the following procedure to convert the kit from required to optional.

**NOTE:** *This procedure only applies to Developer Web Client users.* Kits for Mobile Web Client users cannot be changed from required to optional. If a Mobile Web Client user has difficulty with a required kit, you must deactivate the problematic kit and create a new kit. The new kit should have the same Minimum Old Version and Maximum Old Version settings as the deactivated kit, but it should have a higher New Version number.

### *To convert a required kit to an optional kit (for Developer Web Clients only)*

- 1 From the application-level menu, select Navigate > Site Map > Anywhere Administration.

- 2 From the link bar, select Upgrade Component List.

The Upgrade Components list appears.

- 3 From the Upgrade Components list, select the name of the Upgrade Component for which you distributed the upgrade kit.

- 4 Modify the Min Version field with the minimum value that is acceptable, or change it to null.  
**CAUTION:** Perform this step carefully, because the wrong minimum value could cause some users to skip a related required kit.
- 5 Step off the record, or choose Save Record from the applet menu, to save the record.
- 6 Navigate to the Administration - Siebel Anywhere screen.
- 7 From the link bar, select Upgrade Configurations.  
The Upgrade Configurations and Upgrade Components list applets appear.
- 8 On the Upgrade Configurations list applet, select the configurations you had previously distributed the component information to.
- 9 Click Distribute.

These steps make the upgrade kit optional and allow Oracle's Siebel Developer Client users and Oracle's Siebel Mobile Web Client users to log in to the Component Upgrades view—if the client version they are using is between the minimum and maximum for the upgrade component in the Upgrade Component List view.

## Changing Version Settings After Distribution

If users receive an error message such as “Unable to select required upgrade kits to upgrade your system” when attempting to retrieve an upgrade kit, Minimum Old Version and Maximum Old Version may be set inappropriately.

The following procedure describes how to correct inappropriate values for Minimum Old Version and Maximum Old Version after an upgrade kit has been distributed.

### *To change Minimum Old and Maximum Old versions of a kit after distribution*

- 1 From the application-level menu, select Navigate > Site Map > Administration - Siebel Anywhere.
- 2 From the link bar, select Upgrade Kits.  
The Upgrade Kits list appears.
- 3 From the Upgrade Kits list, select the target upgrade kit.
- 4 If the status is Active, change the status to Pending.
- 5 Change Minimum Old Version and Maximum Old Version values per your requirement.  
If you want the upgrade kit to be available for every client, set both values to blank.
- 6 While the kit is selected, click Activate.

**NOTE:** Do *not* change the status of the kit to Active manually. Manually changing the status in this way does not actually activate the kit.

You do not need to apply or distribute.





# Index

## A

- activating upgrade kits** 93
- Active status, about changing status**
  - manually 143
- adding components to a configuration** 39
- Administration - Siebel Anywhere screen, about using** 11
- administrative tasks, caution running Siebel Smart Web Client** 23
- applying an upgrade kit** 96
- attachments, list of files** 12

## C

- CFG files**
  - caution, about using Customer Revision kit type 84
  - component consistency, testing for 133
  - configuration, overriding 41
  - distributing, different files to different users 100
  - login, checking at 42
  - Siebel Configuration File upgrade kit, defining 64
  - testing distribution to different users 102
  - upgrade kit, about which kit to use 48
- Client Customer Revision, creating for multiple kits** 122
- client upgrade, displaying error** 111
- columns, about downsizing during schema upgrade** 124
- Component Upgrades view, about using** 11
- ComponentName parameter**
  - associating employees with configuration, about 41
  - note, updating name on Siebel clients 38
- components**
  - caution, preserving version numbers for upgrades 31
  - checking status of 138
  - configurations, adding to 39
  - configurations, creating 38
  - configurations, modifying 38
  - configurations, removing from 39
  - defined and example 13
  - deployment recommendations 24
  - existing component versions, displaying information on 31

- new version number, about specifying 14
- synchronizing 59, 137
- upgrade component version information, about distributing 98
- upgrade component version information, distributing 99
- version information, displaying for 31
- computers, identifying to receive upgrades** 28
- configurations**
  - caution, disassociating from incorrect configuration 41
  - creating or modifying 38
  - displaying and related components 28
  - employees, about assigning to 41
  - employees, assigning to 42
  - employees, listing those associated with 29
  - employees, removing from 42
  - languages, about setting up configurations based on 40
  - new configuration, creating 40
  - upgrade components, identifying to deliver 28
- configurations, setting up**
  - components, adding to 39
  - components, removing from 39
  - language, about setting up configuration based on 40
  - new configuration, about creating 39
  - new configuration, creating 40
- custom component upgrade kit, defining** 88
- custom components**
  - creating 46
  - setting up, about 43
  - version numbers, about monitoring and verifying 43
- Customer Revision kits, deleting** 131
- Customer Revisions upgrade kit, about using** 48

## D

- database schema**
  - See Siebel database schema
- DB2, before install database schema upgrade kit** 128
- deactivating upgrade kit**
  - about and guidelines 95

- caution, deactivating kit and mobile users 95
  - deleting**
    - components from a configuration 39
    - Customer Revision kits 131
    - upgrade kits 131
  - delivery method, evaluating** 27
  - delta install upgrade kit**
    - See Siebel Client Executables upgrade kit
  - delta patch file, creating** 53
  - delta repository file upgrade kit**
    - See Siebel Repository File upgrade kit
  - deployment recommendations** 24
  - Developer Web Clients**
    - caution, testing upgrade kits 113
    - database schema upgrades, about delivering to 114
    - disabling Siebel Anywhere 102
    - language packs, distributing additional 118
    - misunderstanding about retrieving 106
    - optional upgrade kits, retrieving for 108
    - required upgrade kits, about retrieving 109
    - required upgrade kits, retrieving 110
    - Siebel Executables upgrade kit, preparing contents for 49
    - Siebel File System, connecting to 47
    - Siebel patches, caution about downloading 117
    - Siebel Upgrade Wizard upgrade kit, preparing contents for 48
    - troubleshooting, converting required kit to optional kit 142
    - upgrade not required 18
    - version check occurrence 17
  - developmental environment, changing database schema in** 125
  - diffsrf utility** 55, 57
  - disabling Siebel Anywhere** 102
  - distribution**
    - caution, before distributing kits 99
    - CFG files, different files to different users 100
    - CFG files, testing distributing to different users 102
    - limiting distribution to specific subscribers 100
    - upgrade kits, about distributing 98
    - upgrade kits, distributing 99
  - DLL versions, creating upgrade kits that check for** 133
  - download, unable to download kit** 139
- E**
- employees**
    - caution, disassociating from incorrect configuration 41
    - configuration, listing those associated with 29
    - login specified from view, about 42
    - subscriber, assigning to upgrade configurations 22
    - upgrade configurations, about assigning to 41
    - upgrade configurations, assigning to 42
    - upgrade configurations, removing from 42
  - environments, moving between database schema** 125
  - errors**
    - client upgrade error recovery 111
    - in Upgrade Kit Wizard 137
    - Upgrade Wizard error recovery, about 12
- F**
- files**
    - attachments, list of 12
    - identifying those to include 30
- G**
- global deployment example**
    - assumptions 119
    - overview 119
    - records 120
    - requirements 119
  - global deployment, note, support of** 23
- H**
- History-dependent components, about** 20
  - history-independent components**
    - keeping 78
    - Minimum Old Version and Maximum Old Version settings 71
  - HQ Server, distributing upgrade kit for Mobile Web Clients** 129
- I**
- In Progress status remains, troubleshooting** 139
  - installing upgrade kits**
    - order, setting 94
    - process of installing 105
    - Upgrade Wizard, launching 111
- K**
- kit installation, setting order** 94
  - kits**
    - See upgrade kits

**L****languages**

- configurations, about setting up based on language 40
- dependent components, planning for 27
- language packs, distributing additional 118

**Locate Method**

- Siebel Anywhere, how used by 45
- using 44

**lockout, recovering from 141****M****maintenance release**

- about and process flow 117
- files, preparing for including in upgrade kits 50

**Max Version setting, about 15****Maximum Old Version**

- caution, setting correctly 31
- distribution, changing after 143
- guidelines for setting 34
- multiple kits, setting for 122
- replacement kit, same values as deactivated kit 98
- upgrade versions, specifying 14

**Min Version setting, about 15****Minimum Old Version**

- caution, setting correctly 31
- distribution, changing after 143
- guidelines for setting 34
- multiple kits, setting for 122
- replacement kit, same values as deactivated kit 98
- upgrade versions, specifying 14

**Mobile Web Clients**

- caution, testing upgrade kits 113
- database schema changes, about preparing for 126
- database schema changes, installing on 129
- database schema upgrade, about synchronizing 131
- database schema upgrade, distributing a kit for 129
- dynamic configuration assignment, detecting 42
- language packs, distributing additional 118
- misunderstanding about retrieving 106
- note, about converting required kit to optional kit 142
- optional upgrade kits, retrieving for 106
- required upgrade kits, about retrieving 109
- required upgrade kits, retrieving 110
- Siebel Anywhere, disabling 102

- Siebel Executables upgrade kit, preparing contents for 49

- Siebel File System, connecting to 47

- Siebel patches, caution about downloading 117

- Siebel Upgrade Wizard upgrade kit, preparing contents for 48

- troubleshooting, about 137

- upgrade not required 18

- version check occurrence 17

**multiple kits of the same type, creating example 122****N****New Version, same values as deactivated kit 98****O****optional upgrade kits**

- about and example 19
- caution, using as testing method 20
- Developer Web Clients, retrieving for 108
- Mobile Web Clients, retrieving for 106
- required kit, converting to 98
- retrieving, about 106
- troubleshooting, converting from a required kit 142
- upgrade requirements, choosing type 30

**P****Packager, about *not* using for maintenance release 117****patches**

- about and process flow 117
- files, preparing for including in upgrade kits 50

**process overview, steps 10****properties**

- entire kits and components, viewing for 89
- Status column, about and values 90
- upgrade kit items and parameters, viewing 91

**R****Regional Node Servers**

- database schema changes, about installing 126
- database schema changes, installing on 129
- database schema changes, process of preparing for 127
- parameters, checking before a database schema update 127

- parameters, setting before distributing 128
  - Siebel Server upgrade kits, setting
    - configuration 111
    - upgrade not required 18
    - version check occurrence 17
  - Related At Startup field, about** 29
  - removing employees from a configuration** 42
  - repair kit**
    - distributing 141
    - troubleshooting, cannot be installed 141
  - repository file upgrade kit**
    - See Siebel Repository File upgrade kit
  - Request Submitted status remains, about troubleshooting** 137
  - required upgrade kits**
    - about 19
    - Developer Web Clients, retrieving 110
    - Mobile Web Clients, retrieving 110
    - optional kit, converting to required kit 98
    - retrieving, about 109
    - troubleshooting, cannot install 141
    - troubleshooting, converting to an optional kit 142
    - upgrade requirements, choosing type 30
  - retrieving Siebel Server upgrade kits**
    - Regional Node Servers, setting
      - configuration 111
    - Upgrade Wizard on UNIX, failure to invoke 113
    - Upgrade Wizard on UNIX, manually invoking 113
    - Upgrade Wizard on Windows, manually invoking 112
  - retrieving upgrade kits**
    - client upgrade error recovery, about 111
    - common misunderstandings 106
    - optional upgrade kits for Developer Web Clients 108
    - optional upgrade kits for Mobile Web Clients 106
    - process of retrieving 105
    - required upgrade kits for Developer Web Clients 110
    - required upgrade kits for Mobile Web Clients 110
    - required upgrade kits, about retrieving 109
    - retrieving, about and types of 106
    - Upgrade Wizard, invoking process by subscriber 106
- S**
- SAF files, about** 13
  - Server Manager, using to check and modify server parameters** 128
  - Server Request processor problems, troubleshooting** 138
  - servers**
    - See Regional Node Servers
  - setup process, optimizing for new users** 113
  - Siebel administrator**
    - account unlocking 141
    - Administration - Siebel Anywhere screen, about accessing 11
    - user account, about running Siebel Smart Web Client 41
  - Siebel Anywhere**
    - benefits of 9
    - concepts for understanding 9
    - Developer Web clients, disabling 102
    - Mobile Web Clients, disabling for 102
    - troubleshooting, about 137
    - wizards and utilities 11
  - Siebel Anywhere component group, verifying availability** 47
  - Siebel Anywhere configuration**
    - about 28
    - displaying related components 28
    - employees, listing those associated with 29
  - Siebel Anywhere subscribers**
    - about and types of 23
    - defined and associated with components 13
  - Siebel Anywhere upgrade components, setting up custom**
    - about 43
    - caution, copying existing components and renaming 46
    - custom component, creating 46
    - version numbers, about monitoring and verifying 43
  - Siebel Anywhere Upgrade Kits**
    - activating 93
    - applying 96
    - caution, about dependent on database schema 126
    - caution, before distributing kits 99
    - caution, deactivating kit and mobile users 95
    - CFG files, distributing different files to different users 100
    - component consistency, testing for 132
    - component deployment
      - recommendations 24
    - Component Upgrades view, about using 11
    - components, predefined list 21
    - contents, preparing 48
    - creating process, steps for completing 93
    - creating, process of planning and

- preparing 25
  - Customer Revision kits, deleting 131
  - deactivating, about and guidelines 95
  - defined and contents of 19
  - deleting old kits 131
  - distributing 99
  - distributing, about 98
  - distribution, limited to specific subscribers 100
  - DLL versions, creating kits that check for 133
  - files, identifying those to include 30
  - kit installation, setting order 94
  - optional kit, converting to a required kit 98
  - optional kits, about and example 19
  - progress, ways to monitor 139
  - properties, viewing entire kits and components 89
  - properties, viewing upgrade kit items and parameters 91
  - Repository file or third-party upgrade kits, preparing contents for 52
  - Siebel Anywhere component group, verifying availability of 47
  - Siebel Anywhere subscribers, about and types of 23
  - Siebel File System, verifying connections to 47
  - Status column, about and values 90
  - testing CFG file distribution 102
  - Third Party-WinZip, using to create third-party upgrade kit 135
  - Third Party-WinZip, using to define third-party upgrade kit 135
  - third-party upgrade kit, example of creating 134
  - upgrade components, about and categories of 20
  - upgrade configurations, about 22
  - upgrade kit sequences, determining 37
  - upgrade requirements, determining 26
  - upgrade test details, planning 37
  - upgrade, specifying version information 15
  - version settings, planning for 32
  - versions, specifying 14
- Siebel Anywhere Upgrade Kits, defining**
- custom component upgrade kit 88
  - download operations, avoiding unnecessary 68
  - Siebel Client Executables upgrade kit 70
  - Siebel Configuration File upgrade kit 64
  - Siebel Configuration File upgrade kit, sequence of elements 64
  - Siebel Customer Revisions upgrade kit 84
  - Siebel Database Schema upgrade kit 66
  - Siebel Database Schema upgrade kit, sequence of elements 68
  - Siebel Repository File upgrade kit 76
  - Siebel Upgrade Wizard upgrade kit 62
  - Third-Party Software upgrade kit 80
  - Upgrade Kit Wizard, running 62
- Siebel Anywhere, global deployment example**
- assumptions 119
  - Mobile Web Clients, preparing for database changes 126
  - overview 119
  - records 120
  - requirements 119
- Siebel CFG file**
- component consistency, testing for 133
  - Customer Revision kit type, caution about using 84
- Siebel Client Customer Revisions, testing for component consistency 133**
- Siebel Client Executables upgrade kit**
- defining 70
  - delta install type, about 51
  - delta install type, process of creating 52
  - delta patch file, creating 53
  - elements, sequence of 72
  - history-independent components, recommendation for keeping 73
  - multiple kits, about deactivating 97
  - reducing size of, about 51
  - standard install type, about 51
  - upgrade sequence, determining 37
- Siebel Client Repository File upgrade kit, preliminary tasks 36**
- Siebel Configuration File upgrade kit, defining 64**
- Siebel Customer Revisions upgrade kit**
- defining 84
  - element, sequencing 85
  - preliminary tasks, list of 35
  - tasks, preliminary 37
- Siebel database schema**
- development environment, changing in 125
  - environments, moving database schemas between 125
  - Mobile Web Clients, about creating for 126
  - Mobile Web Clients, installing on 129
  - Regional Node Server parameters, checking before a database schema update 127
  - Regional Node Server parameters, setting 128
  - Regional Node Servers, process of

- preparing 127
  - Regional Node, installing on 129
  - server parameters, checking and modifying 128
  - updating process 123
  - updating, about 123
  - upgrade kits, about creating 125
  - upgrade, what happens during 124
- Siebel Database Schema upgrade kit**
  - defining 66
  - download operations, avoiding unnecessary 68
  - elements, sequence of 68
  - multiple kits, about deactivating 97
  - preliminary tasks, list of 35
- Siebel database schema upgrades, about delivering to Developer Web Client user** 114
- Siebel Developer Web Client, caution running with administrator account** 41
- Siebel Executables Upgrade Kit**
  - Siebel maintenance release files, preparing for including in 50
- Siebel Executables upgrade kit**
  - using, about 49
- Siebel File System, verifying connections to** 47
- Siebel maintenance release**
  - about and process flow 117
  - files, preparing for including in upgrade kits 50
- Siebel Repository File upgrade kit**
  - defining 76
  - delta repository file, preparing 57
  - delta, process of creating 55
  - downloaded size differing 106
  - elements, sequence of 77
  - history-independent components, keeping 78
  - preliminary tasks, list of 35
  - preparing contents for 52
  - SRF, compiling for delta upgrade kit 56
- Siebel Server upgrade kits, retrieving**
  - Regional Node Servers, setting configuration 111
  - Regional Node Servers, setting configuration using command line 112
  - Upgrade Wizard on UNIX, failure to invoke 113
  - Upgrade Wizard on UNIX, manually invoking 113
  - Upgrade Wizard on Windows, manually
    - invoking 112
- Siebel Server, caution enabling Siebel Anywhere** 24
- Siebel Smart Web Client**
  - caution, running 23
  - Siebel Anywhere subscribers, and are not 106
  - user accounts, about running 41
- Siebel SRF file, caution about using Customer Revision kit type** 84
- Siebel SRF kits, about deactivating multiple kits** 97
- Siebel Test Client, using** 114
- Siebel Upgrade Kit Wizard, about using** 11
- Siebel Upgrade Wizard**
  - about 12
  - error recovery, about 12
  - launching 111
  - retrieving upgrade kits, invoking process by subscriber 106
  - UNIX, failure to invoke 113
  - UNIX, manually invoking on 113
  - Upgrade Kit Wizard, difference from 11
  - Windows, manually invoking on 112
- Siebel Upgrade Wizard upgrade kit**
  - defining 62
  - elements, sequence of 63
  - preparing contents for 48
- siebpatch utility** 52, 53
- software**
  - new software, about distribution of 27
  - upgrading existing, about 26
- SRBroker component, checking status of** 138
- .SRF files**
  - Customer Revision kit type, caution about using 84
  - upgrade kit, about kit type to use 48
- SRF upgrade kits, about deactivating multiple kits** 97
- SRProc component**
  - checking status of 138
  - processing problems, troubleshooting 138
- srvrupgwiz1.log, Upgrade Wizard logfile** 12
- standard install upgrade kit**
  - See Siebel Client Executables upgrade kit
- Status column, about and values** 90
- storing, about version information** 16
- subscribers**
  - defined and associated with components 13
  - types of 23
- system administrative functions, caution about performing** 11

## T

**terminology** 13

**testing CFG file distribution to different users** 102

**testing upgrade kits**

- caution, using optional kit as testing method 20
- new users, optimizing setup process 113
- optional kit, converting to required kit 98
- process of testing 105
- Siebel Test Client, using 114
- test plans, planning 37
- testing recommended method 113

**Third-Party Software upgrade kit**

- defining 80
- elements, sequence of 81
- preliminary tasks, list of 35

**third-party upgrade kits**

- contents, preparing for 52
- tasks, preliminary 37
- Third Party-WinZip, using to create upgrade kit 135
- Third Party-WinZip, using to define upgrade kit 135

**troubleshooting**

- components, checking status of 138
- download, unable to 139
- In Progress status remains, possible causes 139
- lockout, recovering from 141
- Request Submitted status remains 137
- required upgrade kits, cannot be installed 141
- required upgrade kits, converting to an optional kit 142
- Server Request process problems, troubleshooting 138
- Upgrade Kit Builder Activity, checking for 138
- version settings, changing after distribution 143

## U

**UNIX**

- svrupgwiz1.log, Upgrade Wizard logfile 12
- Upgrade Wizard, failure to invoke 113
- Upgrade Wizard, manually invoking 113

**UpgKitBlDr component, checking status of** 138

**UpgKitBlDr\_task\_id.log, Upgrade Kit Builder logfile** 12

**upgrade components**

- about and categories of 20

- caution, using Upgrade Components list 32
- predefined components, list 21

**Upgrade Components list, fields** 29

**upgrade components, setting up custom**

- about 43
- caution, copying existing components and renaming 46
- custom component, creating 46
- version numbers, about monitoring and verifying 43

**upgrade configurations**

- about 22
- subscribers, assigning to 22

**Upgrade Kit Builder**

- about 12
- not responding, actions 139

**Upgrade Kit Builder Activity, checking for** 138

**upgrade kit files, file attachment** 13

**upgrade kit item files, file attachment** 12

**Upgrade Kit Item Parameters list, note about modifying** 91

**Upgrade Kit Items list, note about modifying** 91

**Upgrade Kit Wizard**

- about using 11
- custom component upgrade kit, defining 88
- download operations, about avoiding unnecessary 68
- errors while using 137
- running 62
- Siebel Client Executables upgrade kit, defining 70
- Siebel Client Executables upgrade kit, sequence of elements 72
- Siebel Configuration File upgrade kit, defining 64
- Siebel Configuration File upgrade kits, sequence of elements 64
- Siebel Customer Revisions upgrade kit, defining 84
- Siebel Database Schema upgrade kit, defining 66
- Siebel Database Schema upgrade kit, sequence of elements 68
- Siebel Repository file upgrade kit, defining 76
- Siebel Repository File upgrade kit, sequence of elements 77
- Siebel Upgrade Wizard upgrade kit, defining 62
- Siebel Upgrade Wizard upgrade kit, sequence of elements 63
- Third-Party Software upgrade kit,

- defining 80
  - Third-Party Software upgrade kit, sequence of elements 81
  - version settings, planning for 32
- upgrade kits**
  - activating 93
  - applying 96
  - caution, about dependent on database schema 126
  - caution, before distributing kits 99
  - caution, deactivating kit and mobile users 95
  - caution, Minimum and Maximum version settings 31
  - component consistency, testing for 132
  - component deployment
    - recommendations 24
  - Component Upgrades view, about using 11
  - components, predefined list 21
  - contents, preparing 48
  - creating multiple kits of same type, example 122
  - creating, process of planning and preparing 25
  - creating, required infrastructure for 38
  - creation process, steps for completing 93
  - Customer Revision kits, deleting 131
  - database schema changes, creating kits for 125
  - deactivating, about and guidelines 95
  - defined and contents of 19
  - distribution, about distributing 98
  - distribution, different CFG files to different users 100
  - distribution, distributing 99
  - distribution, limited to specific subscribers 100
  - DLL versions, creating kits that check for 133
  - download, unable to 139
  - files, identifying those to include 30
  - kit installation, setting order 94
  - old kits, deleting 131
  - optional kit, converting to a required kit 98
  - optional kits, about and example 19
  - preliminary upgrade tasks 35
  - progress, ways to monitor 139
  - properties, about viewing 88
  - properties, viewing entire kits and components 89
  - properties, viewing upgrade kit items and parameters 91
  - Repository file upgrade kits, preparing contents for 52
  - required or optional upgrade types,
    - choosing 30
  - required upgrade kits, about 19
  - Siebel Anywhere component group, verifying availability 47
  - Siebel Anywhere subscribers, about and types of 23
  - Siebel Customer Revisions upgrade kit, preliminary tasks 37
  - Siebel Database Schema upgrade kits, preliminary tasks 35
  - Siebel File System, verifying connections to 47
  - Status column, about and values 90
  - testing CFG file distribution 102
  - Third Party-WinZip, using to create third-party upgrade kit 135
  - Third Party-WinZip, using to define third-party upgrade kit 135
  - third-party upgrade kit, example of creating 134
  - third-party upgrade kits, preliminary tasks 37
  - third-party upgrade kits, preparing contents for 52
  - upgrade components, about and categories of 20
  - upgrade configurations, about 22
  - upgrade kit sequences, determining 37
  - upgrade requirements, determining 26
  - upgrade test details, planning 37
  - upgrade, specifying version information 15
  - version settings, planning for 32
  - versions, specifying 14
- upgrade kits, defining**
  - custom component upgrade kit, defining 88
  - download operations, avoiding unnecessary 68
  - Siebel Client Executables upgrade kit, defining 70
  - Siebel Client Executables upgrade kit, sequence of elements 72
  - Siebel Configuration File upgrade kit, defining 64
  - Siebel Configuration File upgrade kit, sequence of elements 64
  - Siebel Customer Revisions upgrade kit, defining 84
  - Siebel Database Schema upgrade kit, defining 66
  - Siebel Database Schema upgrade kit, sequence of elements 68
  - Siebel Repository file upgrade kit, defining 76
  - Siebel Repository file upgrade kit, sequence of



- elements 77
  - Siebel Upgrade Wizard upgrade kit, defining 62
  - Siebel Upgrade Wizard upgrade kit, sequence of elements 63
  - Third-Party Software upgrade kit, defining 80
  - Third-Party Software upgrade kit, sequence of elements 81
  - Upgrade Kit Wizard, running 62
  - upgrade kits, installing**
    - kit installation order, controlling 94
    - process of installing 105
    - Upgrade Wizard, launching 111
  - upgrade kits, retrieving**
    - client upgrade error recovery, about 111
    - common misunderstandings 106
    - optional upgrade kits for Developer Web Clients 108
    - optional upgrade kits for Mobile Web Clients 106
    - process of retrieving 105
    - required upgrade kits for Developer Web Client, retrieving 110
    - required upgrade kits for Mobile Web Client, retrieving 110
    - required upgrade kits, about retrieving 109
    - retrieving, about and types of 106
    - Upgrade Wizard, invoking process by subscriber 106
  - upgrade kits, testing**
    - new users, optimizing setup process 113
    - process of testing 105
    - recommended method 113
    - Siebel Test Client, using 114
  - upgrade planning, process flow** 48
  - upgrade requirements**
    - caution, using Upgrade Components list 32
    - computers and users, identifying 28
    - configurations, displaying and related component 28
    - configurations, identifying to deliver components 28
    - determining, list of 26
    - employees, listing those associated with 29
    - existing component versions, displaying information on 31
    - existing software, about upgrading 26
    - files, identifying those to include 30
    - language-dependent components, planning for 27
    - new software, about distribution of 27
    - new upgrade kit, planning version settings for 32
    - required or optional upgrade types, choosing 30
    - Siebel Anywhere, evaluating as delivery method 27
    - Siebel Customer Revisions upgrade kit, preliminary tasks 37
    - Siebel Database Schema upgrade kits, preliminary tasks 35
    - Siebel Repository File upgrade kit, preliminary tasks 36
    - third-party upgrade kits, preliminary tasks 37
    - upgrade kit sequences, determining 37
    - upgrade test details, planning 37
    - version setting values, determining and types of 31
  - Upgrade Wizard**
    - about 12
    - error recovery, about 12
    - launching 111
    - retrieving upgrade kits, invoking process by subscriber 106
    - UNIX, failure to invoke 113
    - UNIX, manually invoking on 113
    - Upgrade Kit Wizard, difference from 11
    - upgrade kit, defining 62
    - upgrade kit, preparing contents for 48
    - Windows, manually invoking on 112
  - UpgWiz.log, Upgrade Wizard logfile** 12
  - users**
    - optimizing setup process for new users 113
    - upgrades, identifying to receive 28
- ## V
- version checking**
    - Developer Web Clients, retrieving required upgrade kits 110
    - Mobile Web Clients, retrieving required upgrade kits 110
    - required upgrade kits, about retrieving 109
    - upgrade not required, about 18
    - version check occurrence table 17
    - version check occurrence, response to 17
  - version information, distributing** 98
  - Version Method**
    - Siebel Anywhere, how used by 45
    - using, about 45
  - version settings**
    - new version numbers, about specifying 14
    - storing version information 16
    - troubleshooting, changing after distribution 143
    - upgrade kit, specifying upgrade versions 14

- upgrade not required, about 18
- upgrade, specifying versions information 15
- using, about 13
- version check occurrence table 17
- version check occurrence, response to 17

**versions**

- building path using Locate Method 44
- caution, preserving for major upgrades 31
- caution, using Upgrade Components list 32
- existing component versions, displaying information on 31
- information, displaying for components 31
- Locate and Version methods, how Siebel Anywhere uses 45
- Minimum and Maximum Old Versions, guidelines 34
- new upgrade kit, planning version settings for 32
- note, activating and version information 94

- Siebel Customer Revisions upgrade kit, preliminary tasks 37
- Siebel Database Schema upgrade kits, preliminary tasks 35
- Siebel Repository File upgrade kit, preliminary tasks 36
- third-party upgrade kits, preliminary tasks 37
- version method, using 45
- version numbers, about monitoring and verifying 43
- version setting values, determining and types of 31

**W**

- Windows, manually invoking Upgrade Wizard 112**