Going Live with Siebel Business Applications

Version 8.0

December 2006

ORACLE

Contents

Index

1 What's New in This Release

What's New in Going Live with Siebel Business Applications, Version 8.0

Table 1 lists changes described in this version of the documentation to support release 8.0 of Oracle's Siebel software.

Table 1. What's New in Going Live with Siebel Business Applications, Version 8.0

Topic	Description
Chapter 3, "Migrating Customizations Between Applications"	Removed this chapter on migrating customizations using Application Deployment Manager (ADM). This feature is substantially enhanced for 8.0 and is now documented in the *Siebel Application Deployment Manager Guide*.
"Migrating a Repository Using the Database Configuration Wizard" on page 12	Customized database triggers are not migrated, After migration, they will have to be recreated. Updated Windows and UNIX procedures for repository migration.
"Importing and Exporting Repositories Using the repimexp Utility" on page 19	Updated flag for the export option.
"Migrating Web Templates and Related Files" on page 21	Added tip that these files can also be migrated using ADM.
"Running the Siebel Packager Utility" on page 33	Updated optional steps in this topic's task.

2 Migrating Repositories Between Databases

Migrating repositories between databases is a common requirement when going live to a new development, testing, or production environment. This chapter defines and describes the process for migrating repositories.

Part of the process of going live to a new environment may also include a migration of new run-time data customizations. If necessary, perform this task after migrating repository changes. See *Siebel Application Deployment Manager Guide* for information on this process.

This chapter includes the following topics:

About Migrating Repositories Between Databases

Migrate the repository and application customizations between databases so the database schema for the user data, the business objects, and the user interface remain synchronized. Populate the test database, and when sufficient testing has taken place, migrate the repository and update the target database schema. For information on migrating application customizations, see *Siebel Application Deployment Manager Guide*.

For background information on Siebel repositories and schemas, see *Using Siebel Tools*.

For information about setting up your system and database environment, see the *Siebel Installation Guide* for the operating system you are using.

Note the following restrictions and recommendations when migrating repositories between databases:

- Isolate development and test environments from the production environment.
- Do not migrate repository customizations to your target environment until you have completed testing to verify that the customizations work correctly and meet your business requirements.

- Do not migrate repositories between different versions of Siebel applications, as this action leads to an inconsistent environment.

Use the Database Configuration Wizard, as described in *Using Siebel Tools*, to replace the object definitions. Then distribute a new Siebel repository file (SRF) to the servers and any remote clients. For customizations involving schema changes, you also need to use the Database Configuration Wizard, which:

- Updates the data in the target Siebel Server to the new schema.
- Updates the repository object definitions.

All mobile users need to synchronize prior to the schema update and (if you are not using Siebel Anywhere) reextract following the upgrade.

If you are using Siebel Anywhere, mobile users need to synchronize the next time they log on to their local database after the migration has occurred. Synchronizing will download new schema changes from the Siebel Server to the mobile user's local database. If they do not synchronize, there will be a mismatch between the local database and the server database. For further information on Siebel Anywhere, see *Siebel Anywhere Administration Guide*.

NOTE: Importing a repository and then synchronizing the schema definition in a target system is equivalent to migrating the repository. However, after importing the repository and before synchronizing the schema definition, it is necessary to rename the older repository to a temporary name and give the imported repository the correct name. For further information on repository importing and exporting, see *Using Siebel Tools*.

After the repository migration, perform any post-repository migration tasks, if necessary. See "Post-Repository Migration Tasks" on page 21 for information on these tasks.

Process of Migrating Repositories

This topic lists the ordered tasks of the process of migrating repositories between databases.

NOTE: A typical migration session requires the deployment of the new repository and the new non-repository customizations. To deploy (migrate) the repository, use the following process. To migrate non-repository customizations such as views, responsibilities, and lists of values, see "Post-Repository Migration Tasks" on page 21.

To migrate repositories between databases, perform the following tasks:

1. Check in all projects—in both the source and target databases. For information on projects and checking in projects, see *Using Siebel Tools*.

   If you migrate a database schema with some projects still checked out, the migration will work but the project state will be not locked in the target database.

2. Prepare the target database for the new repository. See "Preparing the Target Database for the New Repository" on page 9 for information on this task.

3. Prepare to run the Database Configuration Wizard. See "Preparing to Migrate the Repository" on page 10 for information on this task.

4 Run the Database Configuration Wizard. See "Migrating a Repository Using the Database Configuration Wizard" on page 12 for information on this task.

5 Upgrade mobile databases that are dependent on the target database. See "Upgrading Mobile Databases" on page 20 for information on this task.

Preparing the Target Database for the New Repository

This topic describes the task of preparing the target database for the new repository.

Complete the following actions before you migrate the repository to the target database:

- Make sure that all mobile users perform a full synchronization to avoid any unexpected issues in the target environment as a result of database schema changes made to the new repository.
- The target enterprise can be online for a portion of the repository migration procedure. (For more information on this feature, see "Preparing for Target Environment Status Selection" on page 11.) However, if you do not select this option, stop all Siebel Server tasks and disconnect all database access until migration has been successfully executed.

  NOTE: All connected users (including the database administrator) must disconnect at one point during the running of the Database Configuration Wizard.
- Do a full backup of the target database after all mobile user transactions have been merged.
- Make sure that the target database configuration meets the database requirements outlined in the *Siebel Installation Guide* for the operating system you are using.
- Do not change ODBC parameters or settings for datasources created by the Siebel Server installation.
- Verify the names of all repositories in the target database.

  You will later choose a new name that the repository being migrated will have in the target database. It is recommended that you keep the name of your target repository constant. Accordingly, rename the existing target repository to show that it has been superseded. You will also later import a repository, to which you should give the standard name for your target repository. For further information on repository naming guidelines, see *Using Siebel Tools*.
- Update database statistics if warranted for your database.
- Delete older repositories from the target database. This procedure may run for some time. For further information on this task, see *Using Siebel Tools*.

NOTE: The repository migration process references two ODBC data sources. One is the source database and the other is the target database. When running the migration task from the source system, it is necessary to create an ODBC data source for the target database manually. This applies to all server and database platforms. For further information on verifying the ODBC data source, see the *Siebel Installation Guide* for the operating system you are using.

Preparing to Migrate the Repository

The repository migration process differs based on several selections made during the running of the Database Configuration Wizard. Depending on your selections, some preparatory tasks may be required before running this utility.

The three repository migration selections of note are:

- **Source Repository Selection.** For information on this selection, see "Preparing for Source Repository Selection" on page 11.
- **Target Environment Status.** For information on this selection, see "Preparing for Target Environment Status Selection" on page 11.
- **Schema Changes Selection.** For information on this selection, see "Preparing for Schema Changes Selection" on page 11.

Figure 1 provides a high-level view of the Database Configuration Wizard screen flow based on the different source repository selections.

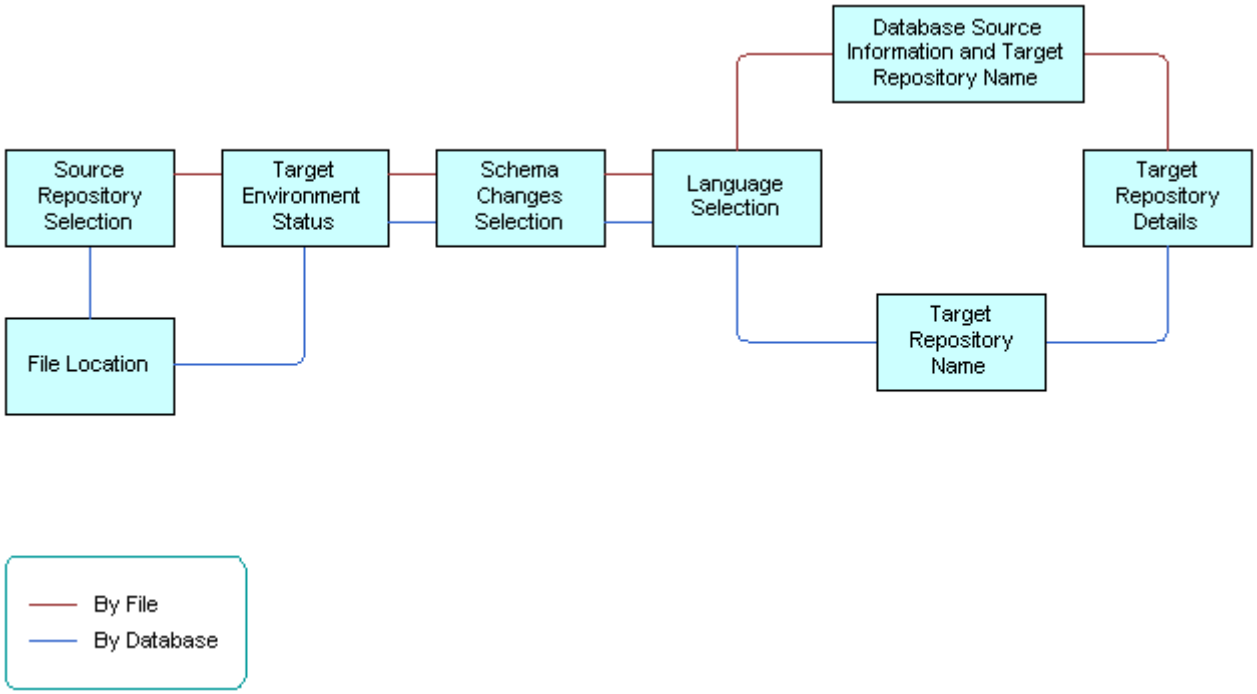

Figure 1. Overview of Repository Migration by Source Repository Selection

Preparing for Source Repository Selection

The source repository selection allows the administrator to migrate the repository:

- By creating a source repository file (**recommended**)

  If using this source repository migration selection, create a source repository migration file before running the repository migration utility. For information on this task, see "Creating a Source Repository Migration File" on page 12.

- Directly from the source database

  If using this source repository migration selection, no preparations are required.

Preparing for Target Environment Status Selection

The target environment status selection allows the administrator the option to import the repository while the target Siebel application is online, which lessens the downtime of the target environment.

To use this option, select the following at the Target Environment Status screen when running the Database Configuration Wizard:

- The Target Enterprise will be online when the migration starts.

Selecting this option allows the administrator to run a repository migration process—a portion of which does not require the target environment to be offline—until it is absolutely necessary that the target environment be shut down. At that point during the repository migration process, a dialog box appears informing the administrator to shut down the target environment for the remainder of the migration.

NOTE: If you are planning to run the repository migration unattended, for example by script, do not select the online option.

Preparing for Schema Changes Selection

The schema changes selection allows the administrator the option not to run the schema migration portion of the repository migration process, which lessens the amount of time required for the migration process.

To use this option, select the following option at the Schema Changes screen when running the Database Configuration Wizard:

- There are no new schema changes.

Selecting this option allows the administrator to skip the schema upgrade portion of the repository migration process if there are no changes to apply to the target's physical or logical schema. Examples of schema updates include the addition of an extension column to an existing table or the addition of a new extension table.

If in doubt, select the option:

- There are new schema changes to be applied.

Creating a Source Repository Migration File

Use the Database Configuration Wizard to create a source repository migration file by clicking the Import/Export Repository selection at the appropriate screen. For complete details on this task in Windows and UNIX environments, see *Using Siebel Tools*.

For further information on other source repository migration preparations, see "Preparing to Migrate the Repository" on page 10.

Migrating a Repository Using the Database Configuration Wizard

Running the Database Configuration Wizard achieves the following results:

- Exports the designated repository from the source database or reads the source repository from file.
- Imports the designated repository into the target database
- Exports the logical schema definition from the specified repository to a control file
- Synchronizes the physical schema of the target database with this logical schema definition

NOTE: The Database Configuration Wizard skips the previous two schema-related steps if you select no schema changes while running the utility.

Take note of the following items before running a repository migration:

- When migrating repositories over a wide area network (WAN) and running the Database Configuration Wizard from the target environment, only the process of exporting the source repository to a flat file takes place on the WAN. All other processing takes place on the local area network (LAN) of the target environment. Consider the export processing times when processing the export over a WAN. The most efficient option is to export the file on the source then transport the exported file to the target environment.
- For multilingual installations where the source repository contains additional language strings, all languages are copied with the repository to the target environment.
- Siebel Business Applications version 8.0 support customized database triggers, but these triggers are not migrated. If you have created customized triggers on your Siebel base tables, you must disable them before migrating the schema. You will then need to recreate the triggers after the migration is finished.

NOTE: If you have a custom table space defined on DB2 UDB for z/OS, the Database Configuration Wizard used in the migration process is tablespace-aware. For further information on z/OS, see *Implementing Siebel Business Applications on DB2 UDB for z/OS*.

To migrate a repository in a Windows environment

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

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

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

The first screen of the Database Configuration Wizard appears.

3 Enter the information you are prompted for in each screen, and click Next to continue.

4 Select Migrate Repository when you are prompted for a database operation.

5 In the Source Repository Selection screen, select to read the source repository from the database or from a previously exported file.

- If you select to read the source repository from file, the Repository File Selection screen appears.

  Enter the path and filename of the source repository file and click Next. The Target Environment Status screen appears.

  For background information on creating a source repository file, see "Creating a Source Repository Migration File" on page 12.

- If you select to read the source repository directly from the database, the Target Environment Status screen appears.

6 In the Target Environment Status screen, select whether the target Siebel Enterprise will be online or offline, and click Next. For further information on this selection, see "Preparing for Target Environment Status Selection" on page 11.

7 In the Schema Changes screen, select whether there are new schema changes to be applied during the repository migration, and click Next. For further information on this selection, see "Preparing for Schema Changes Selection" on page 11.

8 In the Language Selection screen, select the appropriate language, and then click Next.

NOTE: The repository migration process migrates all language strings within the repository. Select the base language installation that applies for this screen.

9 The next succession of screens is dependent on the repository source selection:

- If you chose to read the source repository from file, the Target Repository Name screen appears. Enter the target repository name, and click next.

- If you chose to read the source directly from database, the following succession of screens appears to define the source database.

Screen	Description
ODBC Data Source Name	If you chose to read the source repository from the database, enter the ODBC data source name.
Database User Name	If you chose to read the source repository from the database, enter the source database user name and source database password.
Database Table Owner	If you chose to read the source repository from the database, enter the source database table owner and source table owner password.
Source Database Repository Name	If you chose to read the source repository from the database, enter the database repository name and the target database repository name.

10 Progress by completing the information in the following screens, and then clicking Next.

Screen	Description
Target RDBMS Platform	Select the target RDBMS platform.
Target Database Encoding	Select whether the target database is Unicode. Consult with your DBA on the setup of the target database.
Target Database ODBC Datasource	Enter the target database ODBC datasource.
Target Database User Name	Enter the target database user name and password.
Target Database Table Owner	Enter the target database table owner and table owner password.
Index Table Space Name (DB2-specific)	If you choose IBM DB2 for the target database platform, you get this screen. Enter the index table space name and 4-KB Table Space Name.

Screen	Description
16-KB Table Space Name (DB2-specific)	If you choose IBM DB2 for the target database platform, you get this screen. Enter the 16-KB table space name and 32-KB table space name.
Index Table Space Name (Oracle-specific)	If you choose Oracle as the target database platform, you have only two questions about index and table space. There are no 4-KB, 16-KB, and 32-KB tablespaces in Oracle. For Microsoft SQL server, there are no screens about tablespaces.

11 When the Configuration is Complete screen appears, select one of the following options, and click Next:

- **Yes apply configuration changes now.** The configuration information you entered is saved and you can choose to launch the Siebel Upgrade Wizard in Step 14.
- **No I will apply configuration changes later.** The configuration information is saved but you can *not* choose to launch the Siebel Upgrade Wizard in Step 14.

12 On the Configuration Parameter Review screen, review the configuration values you entered on the previous screens. To change any of the values, click Back to return to the screen with the parameter you need to change. If the values are correct, click Next to continue.

13 You are prompted as to whether you want to execute the configuration:

- Click No if you decide you do not want to continue with the upgrade process.The configuration information you have entered is *not* saved. You must enter the database configuration parameters again.
- Click Yes to continue. The configuration information you have entered *is* saved.

14 Depending on the option you selected in Step 11, do one of the following:

- If you selected the **No I will apply configuration changes later** option, click OK to finish. The configuration information is saved in a master file located in *SIEBEL_ROOT*\bin but the Upgrade Wizard is not launched. You can restart the configuration and run the Upgrade Wizard later. For more information on the Upgrade Wizard, see the *Siebel Database Upgrade Guide*.
- If you selected the **Yes apply configuration changes now** option in Step 11, the configuration information you entered is saved. Click OK and the Siebel Upgrade Wizard is launched; it calls the SQL generator to create or populate SQL scripts.

To migrate the repository in a UNIX environment

1 Verify that the Siebel Server is stopped.

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

2 Make `$SIEBEL_ROOT` the current directory.

3 Source environment variables:

- Korn: . siebenv.sh
- C shell: `source siebenv.csh`

4 Review the values of the following environment variables and confirm the settings are correct:

- `SIEBEL_ROOT`. This path must end in `siebsrvr`, for example `/usr/siebel/siebsrvr`.
- `LANGUAGE`. This is the language in which the Database Configuration Wizard runs. The value of this variable is a language identifier string. For example, enu is the identifier string for English.

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

5 Start the Database Configuration Wizard by running the following command:

```
$SIEBEL_ROOT/bin/ssincfgw -args MODEL_FILE=$SIEBEL_ROOT/admin/dbsrvr.scm
MODE=LIVE
```

The first Database Configuration Wizard screen appears. Enter the information you are prompted for in this screen, and click Next to continue.

6 Enter the information you are prompted for in all subsequent screens. Use the Next and Back button to navigate between screens.

7 Select Migrate Repository when prompted for a database operation.

8 From the Source Repository Selection menu, select to read the source repository from the database or from a previously exported file.

- If you select to read the source repository from file, the Repository File Name menu appears.

  Enter the path and filename of the source repository file and click Next. The Target Environment Status menu appears.

  For background information on creating a source repository file, see "Creating a Source Repository Migration File" on page 12.

- If you select to read the source directly from the database, the Target Environment Status menu appears.
- In the Target Environment Status menu, select whether the target environment will be online or offline, and click Next. For further information on this selection, see "Preparing for Target Environment Status Selection" on page 11.

9 In the Target Environment Status screen, select whether the target environment will be online or offline, and click Next. For further information on this selection, see "Preparing for Target Environment Status Selection" on page 11.

10 In the Schema Changes menu, select whether there are new schema changes to be applied during the repository migration, and click Next. For further information on this selection, see "Preparing for Schema Changes Selection" on page 11.

11 In the Language Selection screen, select the appropriate language, and then click Next.

NOTE: The repository migration process migrates all language strings within the repository. Select the base language installation that applies for this screen.

12 The next succession of screens is dependent on the repository source selection:

- If you chose to read the source repository from file, the Target Repository Name screen appears. Enter the target repository name, and click next.
- If you chose to read the source directly from database, the following succession of screens appears to define the source database.

Screen	Description
ODBC Data Source Name	If you chose to read the source repository from the database, enter the ODBC data source name.
Database User Name	If you chose to read the source repository from the database, enter the source database user name and source database password.
Database Table Owner	If you chose to read the source repository from the database, enter the source database table owner and source table owner password.
Source Database Repository Name	If you chose to read the source repository from the database, enter the database repository name and the target database repository name.

13 Progress by completing the information in each screen.

Screen	Description
RDBMS Platform	Select an RDBMS platform.
Target Database Encoding	Select whether or not the target database is Unicode.
Target Database Repository Name	Enter the target database repository name.
Target RDBMS Platform	Select the target RDBMS platform.
Target Database ODBC Datasource	Enter the target database ODBC datasource.
Target Database User Name	Enter the target database user name and password used by the server components.

Screen	Description
Target Database Table Owner	Enter the target database table owner and table owner password.
Index Table Space Name (DB2-specific)	If you choose IBM DB2 for the target database platform, this screen appears. Enter the index table space name and 4-KB table space name.
16K Table Space Name (DB2-specific)	If you choose IBM DB2 for the target database platform, this screen appears. Enter the 16-KB table space name and 32-KB table space name.
Index Table Space Name (Oracle-specific)	If you choose Oracle as the target database platform, you have only two questions about index and table space. There are no 4-KB, 16-KB, and 32-KB tablespaces in Oracle. For Microsoft SQL server, there are no screens about tablespaces.

14 After you have entered all the requested information, the wizard displays the following message:

```
Configuration is complete: configuration parameters will be saved to
<$Masterfile> file when the wizard completes. Please run the following command
line after you exit from this configuration wizard. This command will deploy the
process you configured to the database.

$SIEBEL_ROOT/siebsrvr/bin/srvrupgwiz /m $SIEBEL_ROOT/siebsrvr/bin/<$Masterfile>
```

15 Click Next to continue. The utility displays the Parameter Review screen listing all the values you have entered.

16 To amend any of the configuration values, click Back to return to the appropriate screen and make changes. Otherwise, click Next.

17 You are prompted as to whether or not you want to execute the configuration:

- Click Yes, and the configuration information is saved in a master file located in `$SIEBEL_ROOT/bin` but the Upgrade Wizard is not launched. For more information on starting the Upgrade Wizard, see the *Siebel Database Upgrade Guide*.
- Click No, and the configuration information you entered is not saved.

NOTE: The database administrator must update statistics after any repository migration, upgrade, or installation finishes.

Importing and Exporting Repositories Using the repimexp Utility

Import and export repositories using the repimexp.exe program only if your migration requires special parameters settings that are inaccessible through the batch files or if you need to perform a file dump. In all other cases, use the Database Configuration Wizard.

For information on importing and exporting repositories using the Database Configuration Wizard, see *Using Siebel Tools*.

The repimexp.exe program imports, exports, or creates a file dump of a repository. It can also import an INTL table. INTL tables contain language-specific information and are a part of the repository.

To import a repository using repimexp

- In the command line, type the following:

  ```
  repimexp /A I /G language_codes
  ```

  where:

  /A I = import switch.

  language_codes = a list such as ENU, FRA, JPN. Use ALL for all languages.

  NOTE: If you want to import your repository with locale objects, you must specify at least one language code. Otherwise, no locale objects will be imported, and the repository will not have any text in the user interface when you compile the imported repository.

To export a repository using repimexp

- In the command line, type the following:

  ```
  repimexp /A E argument_list
  ```

  Export uses the arguments listed in Table 2.

Table 2. Parameter Settings Passed as Export Arguments to repimexp.exe

Parameter	Required	Description
/A E	Yes	Export switch.
/U <userName>	Yes	Siebel administrator user name.
/P <password>	Yes	Siebel password.
/C <ODBC data source>	Yes	ODBC data source. The default is the value in the SIEBEL_DATA_SOURCE environment variable.
/D <table owner>	Yes	Siebel database table owner. The default is the value in the SIEBEL_TABLE_OWNER environment variable.

Table 2. Parameter Settings Passed as Export Arguments to repimexp.exe

Parameter	Required	Description
/G <language code>	No	Exports a repository with a specified language or specify ALL to export all languages in the Siebel repository.
/R <repository>	Yes	Repository name. The default is Siebel Repository.
/F <dataFile>	Yes	Data file, including path, to which to export.
/T <Y|N>	No	Test only, do not export into database.
/V <Y|N>	No	Verify data. The default is N.
/N <0|1|2>	No	Change creation and update information: 0= no change

Upgrading Mobile Databases

This topic describes the task of upgrading mobile databases. This task is a part of the "Process of Migrating Repositories" on page 8. For background information on migrating repositories, see "About Migrating Repositories Between Databases" on page 7.

Follow these steps to upgrade mobile databases:

1. Stop and restart Siebel Remote server components.

   When you have restarted the processes, wait until the Transaction Processor (alias TxnProc) and the Transaction Router (alias TxnRoute) have processed all pending transactions before proceeding with the remaining steps.

2. Regenerate local database templates.

   Use the Siebel Server component Generate New Database (alias GenNewDb) to regenerate the local database template file to update its schema to the same version as the database server. For details on this task, see *Siebel Remote and Replication Manager Administration Guide*.

3. Reextract mobile users.

   If you are not using Siebel Anywhere to upgrade your mobile clients, reextract all mobile users, using the component Database Extract (alias DbXtract).

   If mobile databases are not reextracted, users will still be able to synchronize—no error message will be generated. This functionality allows Siebel Anywhere, which users might use to upgrade mobile databases, to continue working.

   If you are using Siebel Anywhere, refer to *Siebel Anywhere Administration Guide* for instructions on propagating schema extensions.

Post-Repository Migration Tasks

After a successful repository migration, there are other tasks required to migrate or recreate nonrepository configurations from one environment to another. Review the following list to update data and information to the new environment, if required:

- Enterprise customizations data, such as views, responsibilities, assignment rules, and so on. To migrate these customizations, see *Siebel Application Deployment Manager Guide*
- Web Templates and related files. To migrate these files, see "Migrating Web Templates and Related Files" on page 21 and *Siebel Application Deployment Manager Guide*.
- Custom configurations such as parameter values and component definitions. To recreate these configurations, see "Recreating Custom Environment Configurations" on page 22. For specific details on recreating parameter values, see Chapter 4, "Migrating Parameters Between Environments."

After these nonrepository items are migrated, roll out the updates to mobile clients. For more information on this task, see Chapter 5, "Rolling Out Updates to Mobile Clients."

Migrating Web Templates and Related Files

Every Siebel Server installation includes a set of files that define how the application appears in a Web browser. These include:

- Web templates (`.swt` files)
- Images (`.gif` and `.jpg` files)
- Cascading style sheets (`.css` files)

If you make changes to any of these files in your source environment, you must copy the modified files to the target environment.

TIP: You can automate the migration of modified files to a new same-version environment using the Application Deployment Manager (ADM) feature. For information on migrating these files with ADM, see *Siebel Application Deployment Manager Guide*.

For information on specifying the Siebel Enterprise Security Token and how you can explicitly migrate content from the Siebel Server to the Web server, see *Siebel Installation Guide* for the operating system you are using.

To move Web templates and related files

1. Copy any new or modified files of the following types from the source Siebel Server to the target Siebel Server:

Files	File Location
Web Templates	`siebsrvr_root\WEBTEMPL`

Files	File Location
Images	siebsrvr_root\webmaster\images\language_code
Cascading Style Sheets	siebsrvr_root\webmaster\files\language_code

Recreating Custom Environment Configurations

When migrating from one environment to another, recreate any custom environment configurations from the source environment in the target environment, if necessary. Custom environment configurations include data such as custom component definitions. This information is stored in the Siebel Gateway siebns.dat file and is not migrated as part of a repository migration.

For specific details on automatically recreating custom parameter values, see Chapter 4, "Migrating Parameters Between Environments."

Recreate custom environment configurations in one of two ways:

- Use the Server Manager GUI or command-line interface program (srvrmgr) to create these custom configurations manually in the new target environment. For information on creating custom component definitions, see *Siebel System Administration Guide*.
- Use the Server Manager command-line interface program (srvrmgr) with input scripts to automate this task in the new target environment. This is the preferred method because scripts can be saved and reapplied in future environment updates of the same version. For information on using scripts with srvrmgr, see *Siebel System Administration Guide*.

NOTE: If reapplying a srvrmgr script to recreate custom environment configurations, make sure to edit the script if the target environment is a new Siebel version (as in the case of a Siebel upgrade). New parameters or srvrmgr commands may be available with the new version.

3 Migrating Customizations Between Applications

Migrating customizations between applications using the Application Deployment Manager (ADM) feature has been substantially enhanced for release 8.0 of Siebel Business Applications.

Prior ADM functionality documented in this guide was limited to a subset of database data types and deployment was run through the Siebel Business applications GUI or command-line interface.

The 8.0 version of ADM includes new data types available for migration, a new ADM architecture, and a deployment process that includes activation, backups, and detailed logging and status reporting. For complete information on the new ADM feature, see the new *Siebel Application Deployment Manager Guide*.

Migrating customizations outside the new ADM framework, as in past versions, is supported. All documentation on former ADM behavior is also located in the new *Siebel Application Deployment Manager Guide*.

4 Migrating Parameters Between Environments

Migrating Siebel Enterprise or Siebel Server parameter values from one Siebel application environment to another is a common requirement when going live to a new development, test, or production environment. The configuration upgrade utility, cfgmerge, facilitates the automatic migration of these parameters between applications.

This chapter includes the following topics:

About Migrating Parameters Between Environments

Migrating configured Siebel Enterprise and Siebel Server parameter values is a necessary task when updating from one Siebel environment to a newer or alternate Siebel environment.

If undertaken manually, the parameter migration process can incur data entry errors and extend downtime of the target Siebel environment. The configuration upgrade utility, cfgmerge, allows for the automation of the parameter migration process by comparing the source environment with the target environment and creating a parameter migration script, which includes documentation on the parameter differences between the two applications. The migration script documents the differences between environments, including noting parameters that cannot be updated automatically and identifying obsolete or modified parameters in the new environment.

After a review, the migration script can then be applied on the target environment to update the parameters.

Use the automated migration process to:

- Update environments of the same software version (for example, updating a test environment from the development environment).
- Update environments with different software versions (for example, you can migrate parameter values between version 7.7 and 7.8 environments).

NOTE: In all cases, the target environment must be version 7.8 or later.

Migrating parameters between environments can be run in either:

- **Enterprise mode.** This mode migrates enterprise parameters, component definition parameters, and named subsystem parameters between environments.

- **Siebel Server mode.** This mode migrates Siebel Server parameters and Siebel Server component parameters between Siebel Server environments.

For procedures on migrating parameters between environments, see "Process for Migrating Parameters Between Environments" on page 26.

The cfgmerge utility runs on both Windows and UNIX. For further detail on this utility, see "About the cfgmerge Utility" on page 26.

The cfgmerge utility only migrates parameter values between environments. Other environment configurations, such as custom components, are not migrated to the target environment. For example, if you have a custom component in the source environment, you need to create a component definition of the same type, with the same name, in the target environment to migrate the parameter settings for that component. For information on creating component definitions, see *Siebel System Administration Guide*.

About the cfgmerge Utility

The cfgmerge program is a command-line utility that creates a parameter migration script after comparing two different Siebel environments. You run the cfgmerge utility as part of the task "Running Environment Comparison" on page 27. This task is a part of the overall "Process for Migrating Parameters Between Environments" on page 26.

The cfgmerge utility resides in the bin subdirectory of the Siebel Server root directory as the executable program cfgmerge.exe on Microsoft Windows or cfgmerge on UNIX.

The parameter migration script created after a cfgmerge utility execution contains parameter listings, analysis, and recommendations for migrating the parameters. For background information on the parameter migration script, see "About Parameter Migration Scripts" on page 29

Process for Migrating Parameters Between Environments

Perform the following tasks to migrate parameters between applications:

1. Run a comparison analysis between the two applications of interest using the cfgmerge utility. For information on this task, see "Running Environment Comparison" on page 27.
2. Review and edit the migration script, which results from running the application comparison. For information on this task, see "Reviewing and Editing a Parameter Migration Script" on page 28.
3. Run the parameter migration script on the target application to migrate the parameters. For information on this task, see "Running a Parameter Migration Script" on page 30.

Running Environment Comparison

This task uses the cfgmerge command-line utility to run a comparison between two applications and creates a parameter migration script, which documents the parameter differences between the two applications. In all cases, run the cfgmerge utility in the target environment, which must be version 7.8 or later.

To run the environment comparison

1. Before running the cfgmerge utility, make sure all component groups of interest are enabled on both the source and target application.

   For example, if you want to migrate component parameters for a server component in the Siebel Remote component group (alias Remote), make sure this component group is enabled on both the source and target environment.

2. Make a backup copy of the target application's `siebns.dat` file. For information on this GUI or command-line interface procedure, see *Siebel System Administration Guide*. Rename the backup copy to a unique value, for example, `target_siebns.dat`.

   The `siebns.dat` file is available in the Administration folder of the Siebel Gateway Name Server root directory.

3. Make a backup copy of the source application's siebns.dat file. Rename the backup copy to a unique value, for example, source_siebns.dat.

4. Move the `source_siebns.dat` file and the `target_siebns.dat` file copies to the bin subdirectory of the Siebel Server root directory, which contains the cfgmerge utility.

   NOTE: The cfgmerge utility does not require the Siebel application to be up or down when running.

5. Run the cfgmerge utility using the following commands to execute the application comparison. The cfgmerge utility can run a comparison in Enterprise mode or Siebel Server mode:

   - Running Enterprise-mode comparison:

     ```
     cfgmerge -l language_code -i source_siebns.dat,target_siebns.dat -e
     source_enterprise_name,target_enterprise_name -o output_file.cmd
     ```

   - Running Siebel Server-mode comparison:

     ```
     cfgmerge -l language_code -i source_siebns.dat,target_siebns.dat -e
     source_enterprise_name,target_enterprise_name -s
     source_server_name,target_server_name -o output_file.cmd
     ```

   NOTE: Do not include a space between the target and source parameter pairs.

   For details on the cfgmerge utility's flags and arguments, see Table 3.

A successful execution results in the creation of a parameter migration script, which is saved by specifying an output file during command execution. Review this file to note the difference between applications. For further information on this task, see "Reviewing and Editing a Parameter Migration Script" on page 28.

Table 3. Siebel cfgmerge Utility Flags

Flag	Arguments	Description
-e	*source_enterprise_name*, *target_enterprise_name*	Use this flag and two arguments to specify the source and target Siebel Enterprise Server names used for the application comparison. Make sure to include the comma after the first argument, but do not insert a space after the comma.
-i	*source_siebns.dat*, *target_siebns.dat*	Use this flag and two arguments to specify the source and target siebns.dat files. The siebens.dat file, stored in the Siebel Gateway Name Server, defines the configurations of an individual application. Make sure to include the comma after the first argument, but do not insert a space after the comma.
-l	*language_code*	Use this flag to set the language in the script file. The default language code is ENU. Make sure the appropriate language pack is installed before using this flag.
-o	*output_file*.cmd	Use this flag to specify a file name and path for the migration script, which is created as a result of execution of the cfgmerge utility and documents the application differences.
-s	*source_server_name*, *target_server_name*	Use this flag and two arguments to specify the source and target Siebel Server names used for the application comparison. Make sure to include the comma after the first argument, but do not insert a space after the comma.

Reviewing and Editing a Parameter Migration Script

This task provides information on reviewing and editing a parameter migration script, which is created as a result of an execution of the cfgmerge utility. This utility compares the differences between two environments.

To review and edit a parameter migration script

1. Locate the parameter migration script specified by a cfgmerge utility execution.

   The migration script has an extension of CMD, and the default location of the script—that is, if no folder path is specified—is the same folder as the cfgmerge utility.

2. Open the parameter migration script with a text editor.

3. Review the results of the comparison analysis, and make edits to the migration script as appropriate by deleting or adding preceding semi-colons, which activate and deactivate commands respectively.

About Parameter Migration Scripts

A parameter migration script results after an execution of the cfgmerge utility, which compares parameter differences between applications. Migration scripts have the extension CMD and are named as part of the command to run the cfgmerge utility, see "Running Environment Comparison" on page 27 for further details on this task.

The resulting parameter migration script is composed of the following information:

- List of the source and target parameter values if they are different.
- Messages and recommend actions.
- Server Manager command-line interface (srvrmgr) commands to change the target environment's parameter values to match the source environment's value.
- Commented out srvrmgr commands (with a preceding semi-colon) if the utility recommends not to synchronize the values with the source environment.

You must open, review, and edit the migration script prior to running the script. For information on this task, see "Reviewing and Editing a Parameter Migration Script" on page 28.

Parameter migration scripts can act as a documentation record for an environment's configurations. The migration script files can be reused or reviewed for historical comparisons at a later date.

An example portion of a parameter migration script follows:

```
;  Component definition SCCObjMgr_enu
;
;  Parameter Actuate Server Report Server Host (ActuateReportServerHost)
;  Value on source system: acttest_winxp
;  Value on target system: actprod_hpux
;  Recommended action: retain target value
;  To apply value from source configuration, enable the next line
;  change param ActuateReportServerHost=actprod_winxp for compdef SCCObjMgr_enu
;
;  Parameter DB Multiplex - Min Number of Shared DB Connections (MinSharedDbConns)
;  Value on source system: 10
;  No value set on target system:
;  Recommended action: apply value from source
;  To keep the target configuration unchanged, comment out the next line
change param MinSharedDbConns=10 for compdef SCCObjMgr_enu
```

;

Running a Parameter Migration Script

This task provides information on running a parameter migration script created as a result of an execution of the cfgmerge utility. Run the parameter migration script using the following procedure and the Server Manager command-line interface program. For background information on the Server Manager command-line interface, see *Siebel System Administration Guide*.

Make sure to review and edit the migration script before running. For information on this task, see "Reviewing and Editing a Parameter Migration Script" on page 28.

NOTE: As a best practice, run the parameter migration script at times of low usage of the Siebel application.

Running the parameter migration script

1. Copy and save the reviewed and edited migration script in an accessible location for the Server Manager command-line interface program (srvrmgr) accessing the target application; that is, the application receiving the parameter update.
2. Log in to the srvrmgr program and set the program at either the enterprise or Siebel Server level depending on whether the migration script updates enterprise or Siebel Server parameters. For details on these srvrmgr commands, see *Siebel System Administration Guide*.

   NOTE: Make sure server components planned for parameter updates are enabled on the appropriate Siebel Server.
3. Run the migration script using the read command at the srvrmgr command prompt, which inputs commands from the script to the srvrmgr program.

   For example:

   ```
   srvrmgr>read Migration_Parameter_Script.cmd
   ```

   (You can also run the migration script when logging into the srvrmgr program by specifying the `i` switch and file name with the other login parameters.)
4. Check to make sure the parameters have successfully updated in the target application.

5 Rolling Out Updates to Mobile Clients

Rolling out application configuration changes or updates to mobile clients is the final step in going live to a new application environment. Use the Siebel Packager utility for this purpose.

This chapter includes the following topics:

About Rolling Out Updates to Mobile Clients

The Siebel Packager utility allows the Siebel administrator to roll out Siebel application configurations and customizations to Siebel Mobile Web Clients. Siebel Packager assembles the Siebel client executable program and other installed files, including your custom configuration, into a customized installation package.

Siebel administrators can use these installation packages when installing Siebel Business Applications for the first time or when upgrading from previous versions. Optionally, you can package the Siebel client installation as a single, self-extracting archive file.

CAUTION: Do not use Packager to distribute Siebel patch 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.

For details on the process of rolling out customizations using the Siebel Packager utility, see "Process of Rolling Out Updates to Mobile Clients" on page 32.

Distributing Siebel Client Packages

After the Siebel client installation has been packaged, the package can be distributed to your users in several ways. For details, see "Making Your Customized Installer Available to End Users" on page 38.

- **Siebel Anywhere.** Distribute and execute the package automatically as a Siebel Anywhere kit. For further information on Siebel Anywhere, see *Siebel Anywhere Administration Guide*.
- **CD-ROM or DVD.** Distribute the package to end users on CD-ROMs or DVDs.

- **LAN, WAN, VPN, or modem.** Distribute the package across a LAN, WAN, VPN, or modem. Do this directly or using third-party software.
- **Other methods.** Distribute the package by email or FTP to end users.

General Process of Creating a Package

The Packager utility creates a standard installation package in two steps:

1. Gathers the Siebel components and files you specify, copies the standard InstallShield components into the client installer directory, and creates a packing list used by InstallShield.
2. (Optional.) Packages the Siebel client installer (prepared in the previous step) into a self-extracting archive, which, when executed, automatically decompresses and starts the Siebel client installer.

Process of Rolling Out Updates to Mobile Clients

To roll out customizations using Siebel Packager:

1. Make preparations before running the Siebel Packager utility. See "Preparing to Use the Siebel Packager Utility" on page 32 for information on this task.
2. Run the Siebel Packager utility. See "Running the Siebel Packager Utility" on page 33 for information on this task.
3. Distribute the update to mobile clients. See "Making Your Customized Installer Available to End Users" on page 38 for information on this task.

Preparing to Use the Siebel Packager Utility

Before using the Siebel Packager utility, follow the steps in this topic.

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

NOTE: This chapter refers to the Siebel client root directory, such as `C:\program files\Siebel\8.0\Web client`, as *SIEBEL_CLIENT_ROOT*.

To prepare to use the Packager utility

1 Perform a Siebel client installation on the computer on which you will run the Packager utility.

The Packager uses the files from this model client installation (or another client installation, as specified when running the Packager) in creating the installation package.

During Siebel client installation, select *Typical*, or select *Custom* and make sure to select the Packager Utility option.

NOTE: If two Siebel clients are installed on the same computer, where one is used as a Mobile Web Client and the other is a master installation for Packager, make sure to exit the SQL Anywhere engine before running Packager. An installation that has an initialized local database should never be used as a master installation for creating packages.

2 Customize the model Siebel client installation so that it is identical to how you intend to package it. When you create the custom installer, the Packager utility reproduces this model installation.

If you have custom Siebel repository file (SRF) or configuration files (CFG), report files, Web templates, or other changes or additions, copy them to the appropriate subdirectories under *SIEBEL_CLIENT_ROOT*, or under the root directory of another installation that you will use to create the custom installation package. Make sure your custom files are the latest version so they pass version check. For more information please see FAQ 1361 on Siebel SupportWeb.

NOTE: The Packager utility can package only those files that reside in the *SIEBEL_CLIENT_ROOT* directory.

3 Make sure that you have sufficient free disk space on the computer on which you are installing the Packager utility and will create packages.

During the packaging process, the Packager utility temporarily requires at least:

- Three times the amount of disk space required by the Siebel Mobile Web Client software you are packaging, and
- Twice the disk space required by the third-party software (provided with the Siebel Business Application) that you are packaging.

Running the Siebel Packager Utility

This topic describes how to run the Siebel Packager utility. The Siebel Packager utility wizard guides you in creating the custom Siebel client installer.

NOTE: You must run the Packager once for the base Siebel client modules (BASE option), and once for each language pack you want to include. To include elements in the same package, you must specify the same package name each time.

To run the Siebel Packager utility

1 From the Windows Start menu, select Programs > Siebel Web Client 8.0 > Siebel Packager.

The Siebel Client Packager wizard launches and the Choose Setup Language window appears.

2 Choose the language in which to conduct the rest of the Siebel Packager procedure and click OK. The Directory Definition window appears.

3 Specify the following values:

- **Package.** The name of the package. This is used as the name of the self-extracting archive file (if you create one) and as the name of the subdirectory under *SIEBEL_CLIENT_ROOT*\packager\temp in which the custom installer is created.
- **Siebel Client.** The root-level directory of the Siebel client installation that is included in the custom installation. Accept the default, represented in this book as *SIEBEL_CLIENT_ROOT*.
- **Language Packs.** Specify BASE or specify an installed language pack (for example, ENU for U.S. English). If you want to include language packs in the customized installer, select a language pack.

  NOTE: Siebel Packager does not work with custom languages.

4 At the bottom-right of the window, choose Full Install or Patch Install, based on your desired goal:

- **Full Install.** Intended for full installations of Siebel Business Applications. This performs an entire installation, using the parameters in the siebel.ini file. For more information on the siebel.ini file, see Siebel SupportWeb.
- **Patch Install.** Copies only the packaged files, preserving the same directory structure as the source. Typically, this is used with an existing installation not requiring further customizing. When you run a patch installer, it prompts only for the existing installation directory.

5 Click Next and the Module Definition window appears.

6 In the Module Definition window:

a Choose the Siebel modules to be included in the custom installation package.

A list of possible modules appears in the Modules list on the left. For explanations of some of these modules, see Table 4 on page 36.

b If you want to include or exclude a template, select an item in the Modules list. Notice that *.* appears in the Include Templates box on the right.

The Include Templates and Exclude Templates boxes allow you to set the filters used to include or exclude files for each selected component. The default Include filter is *.*, which includes all files.

Include Templates also has an Include Subdirectories check box to indicate whether files in subdirectories for these components are included.

NOTE: You do not need to modify Include Templates and Exclude Templates for a typical installer.

- If you want to prevent a directory from being created for a particular module, select the module and click Remove.

- If you want to add modules that are located in the *SIEBEL_CLIENT_ROOT* directory but do not appear in the Modules list, click Add and specify the path of the module.

To create a required directory without any files, select that module from the Modules list and, under Include Templates, click Remove.

NOTE: When preparing a full installation, do not remove any module or components unless you know they will not be needed. See module descriptions in Table 4 on page 36.

c. Click Next and the Packaging screen appears.

7. To create the custom installer, click Start.

   The Packager utility displays progress information while the Packager executes and creates the package.

8. (Optional.) After this process is finished, you can further customize the behavior of the packaged installer by editing the siebel.ini file. To do so, click the button labeled Edit siebel.ini. For more information on this procedure, see "Customizing siebel.ini Files" on page 39 and Siebel SupportWeb.

9. (Optional.) In order to bypass the installer screen labeled Choose Setup Language, edit the setup.ini file and the siebel.ini file by changing the value of the parameter EnableLangDlg from Y to N.

10. (Optional.) If you want to package the custom installer into a self-extracting archive, click Next.

    NOTE: If you do not want to perform this step at this time, you can do it later by running the `selfex.bat` file in the directory *SIEBEL_CLIENT_ROOT*\Packager\Temp*package_name*.

11. (Optional.) In the Self-extracting Archive window, if you are producing an installer for BASE, click Start to package the self-extracting archive.

    The Packager creates the archive as a single executable file called setupex.exe in the directory *SIEBEL_CLIENT_ROOT*\Packager\Temp*package_name*\selfex. This step may take some time to complete, depending on the processing speed of the computer you are using. Verify the location of the executable file after the process has completed.

About Siebel Modules for Packaging

The Siebel Mobile Web Client consists of the installable modules described in Table 4. These modules correspond to the subdirectories under the *SIEBEL_CLIENT_ROOT* directory. Additional subdirectories, other than those shown here, may apply for your client installations.

NOTE: When you create a package, you can decide which modules or files to include or exclude in the package for distribution to end users. Do not remove any modules or component subdirectories or files unless you know they will not be needed. Verify that your Siebel directory structure contains the files you require, including any modified files or other configuration changes.

Table 4. Siebel Business Applications Modules for Packaging

Component	Description
ACTUATE	Actuate-related files for Reports, located in the *SIEBEL_CLIENT_ROOT*\actuate directory.
BIN	Siebel executable files (binaries) located in the *SIEBEL_CLIENT_ROOT*\bin directory, including the required DLL files, configuration files, and executable files such as siebel.exe. If you have customized the configuration files, replace the default configuration files in this directory with your own before you start the Packager utility. **NOTE:** When you create a package, include this module and all components in the bin directory, except for the user preferences file, *user_ID&SiebelAppname*.spf, and the session file, siebel.ses.
CHARTS	Siebel Charts components for generating charts, located in the *SIEBEL_CLIENT_ROOT*\charts directory.
FONTS	Barcode font files, located in the *SIEBEL_CLIENT_ROOT*\fonts directory.
ISSTEMPL	Template files for the SIS (Siebel Interactive Selling) CDA application, located in the *SIEBEL_CLIENT_ROOT*\isstempl directory.
LOCAL	Location of the local database, local Siebel File System, and docking files for Siebel Remote, located in the *SIEBEL_CLIENT_ROOT*\local directory. **NOTE:** Include this module in order to create the local directory when you create a package for a full installation. However, you should not initialize a local database before you create the package. Each local database is unique to an individual user and should not be packaged.
LOCALE	Language-specific files, located in the *SIEBEL_CLIENT_ROOT*\locale directory. **NOTE:** Include this module and all components in the locale directory when you create a package for a full installation.

Table 4. Siebel Business Applications Modules for Packaging

Component	Description
LOG	Log files from Siebel client operations (such as synchronization), located in the *SIEBEL_CLIENT_ROOT*\log directory. **NOTE:** Include this module in order to create the log directory when you create a package for a full installation.
MSGTEMPL	Message template files used by the Siebel client, located in the *SIEBEL_CLIENT_ROOT*\msgtempl directory. **NOTE:** Include this module and all components in the msgtempl directory when you create a package for a full installation.
OBJECTS	Object configuration template files (configured objects), located in the *SIEBEL_CLIENT_ROOT*\objects directory—the precompiled SRF file to distribute to end users. The objects directory must contain at least one SRF file before you start the Packager utility. **NOTE:** Include this module and all components in the objects directory when you create a package for a full installation.
PACKAGER	**NOTE:** Do not include the Packager utility itself when you create a package for distribution to end users.
PATCH_BACK	Removes modules that were applied as part of a patch. **NOTE:** Do *not* include this module when you create a package for distribution to end users.
PUBLIC	HTML, help, JavaScript, image, and other files for the Siebel client, located in the *SIEBEL_CLIENT_ROOT*\public directory. **NOTE:** Include this module and all components in the public directory when you create a package for a full installation.
REPORTS	Report template files located in the *SIEBEL_CLIENT_ROOT*\reports directory. If you have created your own reports, replace the standard report templates in this directory with your own, or add your own, before you start the Packager utility. **NOTE:** Include this module and all components in the reports directory when you create a package for a full installation.

Table 4. Siebel Business Applications Modules for Packaging

Component	Description
SAMPLE	Location of the Sample Database and sample Siebel File System, located in the *SIEBEL_CLIENT_ROOT*\sample directory—if you have installed the Sample Database. **NOTE:** Generally, you would not distribute the Sample Database to your end users, who will access a local database (Mobile Web Client). You may instead decide to distribute a separate client installer package that includes the Sample Database. If you do this, include the Sample Database, sample configuration files that refer to the Sample Database (located in *SIEBEL_CLIENT_ROOT*\bin*LANGUAGE*), and sample SRF files (located in *SIEBEL_CLIENT_ROOT*\objects).
SQLTEMPL	SQL template files, located in the *SIEBEL_CLIENT_ROOT*\sqltempl directory.
TEMP	Working report files, located in the *SIEBEL_CLIENT_ROOT*\temp directory. **NOTE:** Include this module in order to create the temp directory when you create a package for a full installation.
UPGRADE	Siebel Anywhere upgrade files retrieved by the user, located in the *SIEBEL_CLIENT_ROOT*\upgrade directory. **NOTE:** This module is applicable only to upgrades, not to new installations.
WEBTEMPL	Siebel Web templates, located in the *SIEBEL_CLIENT_ROOT*\webtempl directory. **NOTE:** Include this module and all components in the webtempl directory when you create a package for a full installation.

Making Your Customized Installer Available to End Users

After you have tested your customizations and are satisfied with the client installer you have created, make your customized installer available to end users.

You can distribute your customized installer to end users using one of the methods described in the following topics:

- "Deploying the Installer Using Siebel Anywhere" on page 39
- "Deploying the Customized Siebel Client Installer" on page 39

The suitability of each method will depend on many factors, including available network bandwidth.

Deploying the Installer Using Siebel Anywhere

You can distribute and execute the customized Siebel client installer automatically as a Siebel Anywhere kit.

User access to a Siebel Anywhere kit requires an existing installation of the Siebel client and administrative rights on the user's machine. Therefore, you can use Siebel Anywhere for upgrades, but not for an initial deployment or new installations.

For more information on using Oracle's Siebel Anywhere, see *Siebel Anywhere Administration Guide*.

Deploying the Customized Siebel Client Installer

For new installations or for upgrades, you can distribute the customized Siebel client installer to end users on CD-ROM or local area network (LAN). You can also distribute the customized Siebel client installer across a WAN or VPN, using a modem, by email, or by FTP.

The customized Siebel client installer may optionally be in the form of a self-extracting archive file. Packaging the installer as a self-extracting archive is generally best for distribution methods such as using a modem, email, or FTP.

To distribute a customized Siebel client installer on a CD-ROM or network

1. Perform the following step, as applicable:
   - *CD-ROM:* Place the self-extracting archive file (setupex.exe) or installer package directory onto a CD-ROM, then distribute the CD-ROM to your users.
   - *Network:* Place the self-extracting archive file (setupex.exe) or installer package directory in a network-accessible directory, such as on a LAN. Make sure that all users have access to this directory.
2. Notify your users how to access the installer package and initiate the installation process. Optionally, users may need to copy files to their local machine, such as if you packaged a self-extracting archive file.

   The procedure users will follow varies according to how you distributed the package, how you created the package, and whether the package is a self-extracting archive file. Users can install a self-extracting archive package by running the archive file.

Customizing siebel.ini Files

The siebel.ini file controls the behavior of the Siebel client installation. Separate versions of this file are created for packages for a base installation and those for language packs. The siebel.ini file is located in the following directories, where *package_name* is the name of your package:

- *SIEBEL_CLIENT_ROOT*\packager\temp*package_name* (for BASE)

- *SIEBEL_CLIENT_ROOT*\packager\temp*package_name**LANGUAGE* (where LANGUAGE is the subdirectory for the applicable language pack, such as ENU for U.S. English)

NOTE: The value of the FolderName parameter, described in "Key Parameters in the siebel.ini File" on page 40, must be the same in all siebel.ini files for the same package.

Review all applicable siebel.ini files and customize the files as necessary to make sure the client installation uses the correct settings for your specific environment.

NOTE: If you customize the siebel.ini file, do so at Step 8 in "Running the Siebel Packager Utility" on page 33. This step occurs after you create the package itself, and before you (optionally) create a self-extracting archive file. If you customize the siebel.ini file before you create the package, those changes will not become part of the customized installer.

The siebel.ini file determines all of the parameters used by the client installer, including the following:

- Third-party software programs and versions that are required on the client computer
- System settings that improve performance
- Configuration of data sources
- Which installation screens end users see when they run the Siebel client installer
- Which shortcuts (icons) are created upon installation

Instructions for modifying the siebel.ini file can be found in the file itself. Use a standard text editor to review and edit siebel.ini.

Key Parameters in the siebel.ini File

The following parameters, located in different sections in the siebel.ini file, are generally already set appropriately, based on performing the master installation. Review these settings and modify those that require it.

- Set the FolderName parameter in the [Defaults] section for each language pack to the same value as the FolderName parameter in the [Defaults] section for the base installation. If you do so, the relevant shortcuts (icons) will be delivered to the same location under the Windows Start menu.
- Set the RootDirectory parameter in the [Defaults] section to the installation location on the target system—for a full, language pack, or patch (maintenance release) installation. By default, the parameter value is populated by the Packager utility to correspond to the staging location where you built the package—for example, C:\sea752\client. You can leave this value as is, or set it to another location on the target system.
- Add and set the parameter NoFolder to No in the [Startup] section to suppress the creation of the startup folder at install. Set this parameter in both the base and language siebel.ini file.
- Do not enclose the DockConnString parameter value in double quotes.
- Do not modify the parameter AppServer in the [Startup] section.

- If the SystemDSN parameter is set to no, the installer creates single-user data sources. This type of data source is visible only to the user who installed it.

  If SystemDSN parameter is set to yes, the installer creates system data sources, which are shared by all users who log into that particular machine. You must have administrator privileges to create a system installation.

- You can use the addLanguages parameter to enable users to add language packs to an existing Siebel client installation.

  If addLanguages (in the [Dialog] section) is defined and set to yes, when the user installs, a single dialog box appears, listing all installed instances of the Siebel client. The user can choose a Siebel client instance to add a language pack to, or click Next to install a new Siebel client.

  If addLanguages is set to no, then an installation directory must be specified using the RootDirectory parameter.

  For a package installing a patch, do not use the addLanguages parameter.

NOTE: Do not enable any dialog boxes in a packaged installation, except the one described above for adding languages. Enabling any other dialog boxes in the Packager is redundant, because the necessary user input has already been captured by the initial installation. (If other dialog boxes are enabled, user input is ignored.)

Major Sections of the siebel.ini File

The major sections of the siebel.ini file are as follows:

[Startup]—This section defines values needed for setup initialization. Examples include version number, patch install, application name, and so forth.

[AppCollision]—This section defines the file which defines a product. This file is then used to validate that no installations try to overwrite one another.

[StartupFiles]—This section defines third-party startup (.ini) files that may need to be updated or expanded during installation.

[DeleteFiles]—Defines any files that need to be deleted prior to file delivery.

[Module]—Defines component descriptions for use in the user interface.

[Module.Configuration]—This section defines what components or features to assign to a given setup type.

[Module.Destination]—This section defines where a component or feature should be installed on the end user's system.

[Regsvr32]—This section defines the files to register on Windows when using the regsvr32 utility.

[Prerequisites]—This section defines prerequisites for installation to proceed.

[Dialog]—This section defines what dialog boxes should be enabled or disabled in the installer at run time.

[Defaults]—This section defines default values to be used in the user interface of the installer, or the prompt to use if a dialog box is turned off.

[Behavior]—This section defines general installer behavior, such as whether to abort or continue on a failed condition.

[RunAfter]—This section defines the programs or functions to run or call after the installation is complete.

[CustomUninstall]—This section defines programs or what functions should be run or called during uninstallation.

[Icons]—This section defines what shortcuts (icons) to create on the end user's system.

About siebel.ini File Hierarchy and Organization

This section describes the hierarchy and organization of the siebel.ini file.

Sections Containing Child Sections

In sections like the following, a child subsection provides additional information about the parent section. Parent and child sections take the following format:

```
[Parent_Section_Name]
Child_Section_Name = Value
```

where *Child_Section_Name* is a key in the parent section.

```
[Child_Section_Name]
Key = Value
Key = Value
```

The key in the parent section tells the installer whether that element is enabled. If it is enabled, the installer looks to the child section whose name is the key from the parent.

In the following example, the [AppCollision] section is traversed. The installer finds a key (GtwySrvr) and determines if that check should be enabled. In this case, the check would be enabled if a Siebel Gateway was selected during installation. If so, the installer looks for the definition. The previous key (GtwySrvr) is redefined as a section, which then defines the behavior.

```
[AppCollision]
GtwySrvr = $(Gateway Selected)=yes

[GtwySrvr]
Description = Siebel Gateway
File = gtwysrvr\bin\namesrvr.exe
```

Sections Without Child Sections

In sections like the following, all necessary information for the key is contained in the value:

```
[Section_Name]
Key = Value
```

In the following example, the installer displays a welcome dialog box. All necessary information for the key is contained in the value; no child section is required.

```
[Dialog]
Welcome = yes
```

Testing an Installer After Customizing the siebel.ini File

After you customize the siebel.ini file, you should test the installer.

To test an installer

1. Finish modifying the siebel.ini file.
2. Run install.exe from the *SIEBEL_CLIENT_ROOT*\packager\temp*package_name* directory on the network installation server.
3. Repeat Step 1 and Step 2 until the siebel.ini file is configured to achieve the desired installation.
4. Test any self-extracting archive file you create.

NOTE: If you test a self-extracting archive file on the same machine on which you generated the package, the installer will overwrite the Siebel client directory. To install to a different location, specify an installation directory using the RootDirectory parameter in the siebel.ini file. Or, run the archive file on a different machine. (Run test installs on clean machines to avoid registry problems.)

Index

U

W