
# Siebel Remote and Replication Manager Administration Guide

Version 7.7

March 2004

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404

# Contents

## Chapter 3: Implementing Siebel Remote Server



## Chapter 4: Setting Up Mobile Web Clients

## Chapter 6: Configuring and Using Mobile Web Client



## Chapter 7: Administering Siebel Remote

## Chapter 11: Administering Siebel Replication Manager



## Chapter 12: Upgrading Regional Nodes



## Chapter 13: Troubleshooting Remote and Replication Manager

# 1 What's New in This Release

## What's New in Siebel Remote and Replication Manager Administration Guide, Version 7.7

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

Table 1. New Product Features in Siebel Remote and Replication Manager Administration Guide, Version 7.7

| Topic | Description |
|---|---|
| Security Adapter Support for Synchronization Authentication<br><br>See "Providing Security Adapter Authentication for Synchronizing the Local Database" on page 42. | Security adapter authentication is supported for Mobile Web Client users to synchronize their local databases with the server database. |
| Enhanced Conflict Resolution Options<br><br>See "MRG:Inter Table Conflict Res" on page 52. | This enhancement lets Siebel administrators choose between available preference values to determine how each Siebel implementation will handle insert conflicts on intersection tables. |
| New Time Filtering Feature<br><br>See "Preventing Extraction and Synchronization of Older Data" on page 55. | This new feature allows administrators to prevent selected kinds of older data from being sent to all Mobile Web Clients during database extraction or synchronization. By reducing the amount of data to be sent to Mobile Web Clients, the use of time filtering can reduce the time required for database extraction and synchronization, and may also enhance Mobile Web Client response time. |
| Resetting the Password for the Local DBA Account at GenNewDb<br><br>See "Changing the Local Database Administrator Password" on page 66. | Whenever you run the Generate New Database component to create a database template, you can use a runtime parameter, NewDbaPwd, to specify the DBA account password for the local databases that will use the new database template. A local DBA password that is specified as a runtime parameter is used *only* for the current run of Generate New Database. If no password is specified at runtime, the local DBA password is set to the value of the parameter as defined in an administrative view. If no administrator has set the value of the parameter in an administrative view, a default value based on the Enterprise name is used. |

Table 1. New Product Features in Siebel Remote and Replication Manager Administration Guide, Version 7.7

| Topic | Description |
|---|---|
| Standard Encryption of the Local Database<br><br>See "Encrypting the Local Database" on page 67. | This new feature provides the option to encrypt the local database template to provide a layer of security against unauthorized access to the local database. |
| Enhanced TrickleSync Administration<br><br>See "Setting Up TrickleSync for Users as an Administrator" on page 79. | This enhancement allows a Siebel administrator to set up the TrickleSync feature (formerly called Autosynchronization) for one or more users. |
| Separate Credentials for Synchronization and Local Database Access<br><br>See "Providing Credentials to Synchronize the Local Database" on page 81 and "Changing the Local Database Password" on page 85. | The credentials for access to the local database and for synchronization can be managed independently. The Mobile Web Client user can change the local database password. New options are provided for providing the synchronization credentials without having to enter them manually. |
| Strong Encryption of Local Database Password<br><br>See "Configuring the Mobile Web Client for Local Database Password Encryption" on page 83 and "Configuring the Remote Server for Local Database Password Encryption" on page 105. | This new feature allows RSA SHA-1 encryption to be applied to the local database password before it is passed to the local database for authentication. |
| Routing Model Enhancements<br><br>See "Routing Model Enhancements in Siebel Release 7.7" on page 90. | Routing model definitions can now exclude information (prevent it from being routed) at the dock object level. The Enterprise dock object is one of the dock objects that routing models can exclude. |
| New Handheld User Routing Model<br><br>See "Handheld User Routing Model" on page 93. | This new routing model is designed for mobile users who synchronize Siebel Handheld applications with Siebel Mobile Web Clients, and who use Siebel Remote to synchronize Siebel Mobile Web Clients with the Enterprise database. |
| New Life Science User Routing Model<br><br>See "Life Science User Routing Model" on page 93. | This new routing model is similar to the Mobile Client - Standard model, but it contains extra rules required by the Siebel Life Sciences Mobile Web application. |

Table 1. New Product Features in Siebel Remote and Replication Manager Administration Guide, Version 7.7

| Topic | Description |
|---|---|
| New Selective Retrieval Routing Model<br><br>See "Selective Retrieval Routing Model" on page 94. | This new routing model supports the use of the new Selective Retrieval feature, which allows users to specify particular records to be included in Mobile Web Client synchronizations. |
| New Selective Retrieval Feature<br><br>"Using the Selective Retrieval Routing Model" on page 94. | This new feature allows users to specify particular records to be included in Mobile Web Client synchronizations. This functionality provides an alternative to more traditional Routing Models for specifying the data to be synchronized.<br><br>Selective Retrieval is suitable for Mobile Web Client users who *also* regularly use another type of Siebel client to connect directly to Enterprise or Regional databases. This distinction is necessary because of the following limitations in the Selective Retrieval feature:<br><br>■ To specify the records to be synchronized, Selective Retrieval users need a direct connection to an Enterprise or Regional database.<br><br>■ Certain advanced features, such as forecasting and quotes, are not available to Selective Retrieval users when they are using the Mobile Web Client.<br><br>For appropriate users, the benefits of using Selective Retrieval can include reduced synchronization time and reduced local database size, due to a reduction in the amount of data that is transferred during synchronization. |
| Enhanced Database Extraction to CD-ROM<br><br>See "Performing a Database Extract to a CD Directory" on page 107. | This enhancement uses a new configuration file parameter, DbinitLocalSource, to specify the directory that will be used to store files to download for the initialization of the client's local database. In previous releases, the FileSystem parameter was used for this purpose.<br><br>You may need to change the value of this parameter if you want a Mobile Web Client to initialize a local database from a CD-ROM or a directory other than the default location on the Siebel Remote Server. |
| Enhanced Monitoring<br><br>See "About Automatic Notification of Critical Conditions" on page 129. | This new feature provides automatic email notification when certain critical conditions are reached in the Siebel system. |
| Enhanced Monitoring<br><br>See "Checking Siebel Remote Transaction Backlogs" on page 136. | This new feature allows administrators to check the number of Siebel Remote transactions waiting for handling by the Transaction Router, on a per user, per server, or Enterprise basis. |

This version of the documentation also contains the following general changes:

- Changed the following navigation paths, throughout this guide:
  - Siebel Remote Administration screen changed to Administration - Siebel Remote screen
  - Application Administration screen changed to Administration - Application screen
  - Data Administration screen changed to Administration - Data screen
  - Server Administration screen changed to one of the following, depending on the specific view required:
    - Administration - Server Configuration
    - Administration - Server Management

    In particular, most components are now started from Administration - Server Management > Jobs.
- Removed instructions for navigating to Site Map from the View menu. Site Map can now be accessed from the Navigation menu or by clicking the Site Map icon.
- Changed instructions using the Show drop-down list to instructions using the new link bar navigation element, throughout this guide.
- Changed references to the Autosynchronization feature to the new feature name TrickleSync.
- Added the Configuring and Using Mobile Web Client chapter to this guide. Most of the information in this chapter was formerly located in the discontinued *Siebel Web Client Administration Guide.*

Siebel Release 7.7 also includes improved resiliency for the Replication Agent. No documentation is required, as the enhancement automatically applies to all Replication Manager deployments.

# 2 Siebel Remote Concepts

This chapter provides an overview of Siebel Remote. It describes Siebel Remote architecture, components, software, setup, security, and resolution of data conflicts.



## About Siebel Remote

Siebel Remote allows Mobile Web Clients (typically operating remotely, in disconnected mode on laptops) to connect to a Siebel Server and exchange updated data and files, a process known as synchronization. Siebel Remote supports mobile computing by allowing field personnel to share current information with members of virtual teams of other mobile and connected users across the organization.

As mobile users enter and update information in their local databases, Siebel Remote client software tracks the changes as synchronization transactions. Subsequently, when the user connects to the Siebel Remote server (through a modem, LAN, WAN, or other network to include a VPN), these transactions are uploaded from the Mobile Web Client to the server.

Between synchronization sessions, the Siebel Remote server prepares transactions applied to the database server by other users. Siebel Server components then write these transactions to a separate outbox for each mobile user. The transactions—combined with updated, published, or requested marketing literature, correspondence templates, and other types of file attachments—are downloaded to the Mobile Web Client during the next synchronization session.

**NOTE:** This guide assumes that you have successfully installed your Siebel application and completed the implementation steps described in *Applications Administration Guide*.

# Siebel Remote Architecture

This section describes the architecture for Siebel Remote and illustrates the process flow.

# Elements in Siebel Remote Architecture

Figure 1 illustrates major elements in the Siebel Remote architecture.

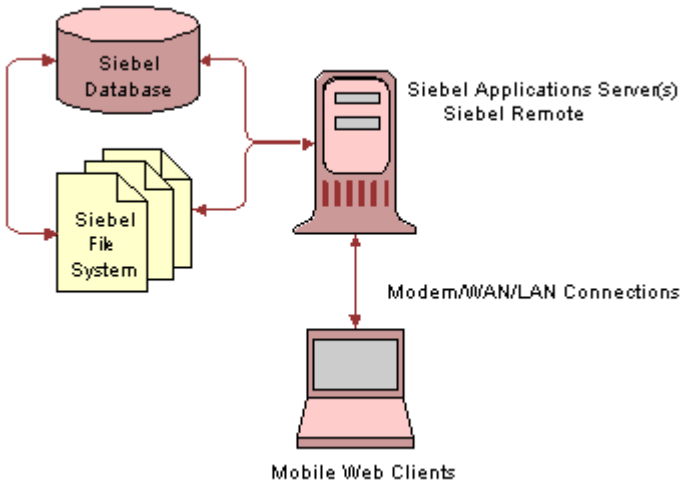


Figure 1. Siebel Remote Hardware Architecture

**Siebel database.** A computer on which the Siebel database resides. Tables in this database store information about your organizational structure, job responsibilities, sales personnel, sales territories, accounts, sales opportunities, and product lines. Other enterprise-wide databases may also reside on this server.

**Siebel File System.** A directory structure that contains the Siebel Encyclopedia items, correspondence templates, file attachments, and other files for client access and download. The Siebel Remote server must have network connectivity to the Siebel File System, which may be located on a Siebel Server, Siebel database server, or other server on the network. For more information on the Siebel File System, see "Siebel File System" on page 25.

**Database access.** The combination of hardware and software that enables a Siebel Remote server to access the Siebel database server. This connectivity is typically established using an ODBC driver and the database vendor's specific connectivity package. Due to the volume of data exchanged between the Siebel Remote server and the database server, this access needs to be provided over a 100 MB or faster network connection, such as Fast Ethernet or FDDI.

**Siebel Remote server.** A Siebel Server that runs the Siebel Remote components and manages synchronization sessions with Mobile Web Clients. Each Siebel Remote server manages a distinct group of mobile users who are assigned to that specific server. Other Siebel Server components, such as Assignment Manager and Enterprise Integration Manager, may run on the same Siebel Server.

To maintain a high level of integrity and availability, the Siebel Remote server provides an interim storage area for data required to synchronize mobile databases with the Siebel database server. To make sure of the integrity and availability of this data, administrators should implement a redundant disk configuration for the Siebel Remote server. Using redundant disk configuration will reduce the likelihood of losing data on the Remote server caused by malfunctioning hardware. However, if it does occur, a reextract of all Mobile Web Clients registered on the affected Remote server will be necessary.

**Siebel Remote Client.** Siebel Remote client software installed on the Siebel Mobile Web Client. During synchronization, it communicates directly with the Synchronization Manager component on the Siebel Server to synchronize the mobile node with its parent.

Components of Siebel Remote client are (a) Local Database and File System, and (b) Local Database Initialization Program.

**Modem, LAN, WAN, or VPN.** The combination of hardware and software that enables a Mobile Web Client to communicate with the Siebel Remote server. Mobile Web Clients must communicate with the Siebel Remote server using the TCP/IP protocol.

**Client.** A computer running the Siebel applications client software. The Mobile Web Client is used for Siebel Remote.

**Siebel Mobile Web Client.** A computer (typically a laptop used by field professionals) that normally runs Siebel applications as a stand-alone unit, using a local database and local file system. The local database and file system contain specific data and files that are initially drawn from the enterprise database and Siebel File System. The Mobile Web Client must connect to a Siebel Remote Server periodically to synchronize both the local database and local file system with their Enterprise counterparts. The connection is made using a modem across public telephone lines, LAN, WAN, or VPN.

The Siebel applications that run on a Mobile Web Client run inside a standard Web browser. However, layers of the Siebel eBusiness Applications architecture, including the local database and file system, reside on the user's personal computer and execute business logic locally. This architectural configuration requires installation of Siebel software on each Mobile Web Client computer.

The software that is installed for a Mobile Web Client is the same as the software that is installed for a Dedicated Web Client. However, the installation program requires more input for installation of a Dedicated Web Client than a Mobile Web Client. A Dedicated Web Client is capable of functioning as a Mobile Web Client (using a local database and local Siebel File System) if all setup steps for Mobile Web Clients are followed, including registering the user as a Mobile Web Client and extracting a local database. A Mobile Web Client can function as a Dedicated Web Client (using the enterprise database and Siebel File System) if appropriate modifications are made to its configuration (CFG) file.

Siebel Mobile Web Clients include a lightweight HTTP listener that listens on a dynamic port for HTTP requests from the local machine. Any requests from other machines on the network are ignored. Therefore, minimal security risk exists when the application is used on the network.

# Siebel Remote Flow Diagram

Figure 2 illustrates the Siebel Remote flow process. Numbers in the diagram relate to explanations in Table 2 on page 19. The purpose of this diagram and table is to provide a general overview of the process.

After Table 2 on page 19, there are smaller diagrams for the two parts of the Remote flow process—data downflow (Figure 3 on page 23) and data upflow (Figure 4 on page 24).

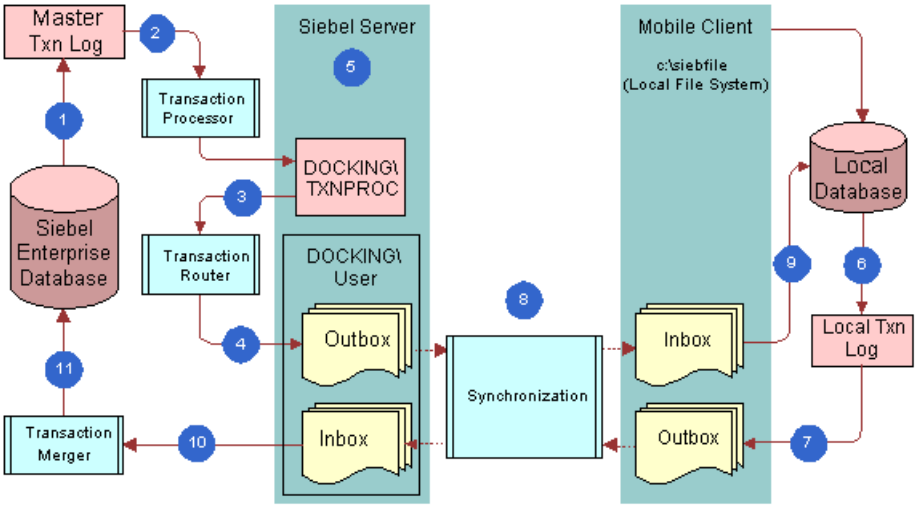


Figure 2. Siebel Remote Flow Diagram

Table 2. Steps in the Siebel Remote Flow Diagram

| Step | Explanation of the Diagram |
|---|---|
| 1 | Every action in the Siebel database is considered a transaction.<br><br>These include adds, deletes, updates, merges, and so on.<br><br>A copy of each transaction is stored in the Master Transaction Log (S_DOCK_TXN_LOG), provided that the transaction is accomplished using Siebel software. Direct SQL modification of Siebel tables is not supported. SQL changes that are executed outside the Siebel application are not recorded in the Master Transaction Log (S_DOCK_TXN_LOG) and therefore will not be routed to mobile clients.<br><br>When using EIM to import records to the database, transactions are logged in the File System to improve performance. If Mobile Web Clients will have read/write access to the records imported during a particular EIM session, it is strongly recommended that the administrator use the row-by-row logging method. For additional information about this topic, see *Siebel Enterprise Integration Manager Administration Guide*.<br><br>Transactions are stored at the field level to minimize the size of S_DOCK_TXN_LOG. When Transaction Logging is turned on and an action occurs, only changes to the fields are captured as transactions. This helps to optimize the synchronization process. |
| 2 | The Transaction Processor picks up the transactions stored in S_DOCK_TXN_LOG and copies them to the Applications Server directory DOCKING\TXNPROC.<br><br>The Transaction Processor also picks up some transactions from the file system and copies them to DOCKING\TXNPROC. These file system transactions are of type External File.<br><br>After the transactions are copied to Applications Servers, TXNPROC clears the S_DOCK_TXN_LOG of the transactions. |
| 3 | Transaction Router picks up the transactions from DOCKING\TXNPROC and determines which mobile users should receive them. |
| 4 | When a Remote administrator first creates a database extract for a remote user, this action triggers the creation of a docking directory on the remote server (a Siebel Server) for that user.<br><br>It creates an inbox and an outbox.<br><br>The outbox stores any future extracts as well as DX files from Transaction Router.<br><br>It then sends these DX files to each Remote client's outbox on the server in the docking directory based upon visibility and routing rules.<br><br>After Transaction Router completes its task, it instructs Transaction Processor to purge the DX files in the DOCKING\TXNPROC directory base on the Transaction Processor's parameter: Clean .dx files iterations. |

Table 2. Steps in the Siebel Remote Flow Diagram

| Step | Explanation of the Diagram |
|---|---|
| 5 | During implementation of Siebel eBusiness Applications, at least one Siebel Server is designated as a Remote server. It will host all or some of the Remote users.<br><br>This server contains the Docking Directories for remote users to transfer the files involved with the synchronization process. These files include:<br><br>New database templates (from the Generate New Database task)<br><br>Database extracts (used with the templates to initialize the local DB)<br><br>DX files (.dx)<br><br>TOC files (.toc)<br><br>Visibility data for the Remote clients (visdata.dbf and dobjinst.dbf) |
| 6 | Remote clients process transactions in their local database while in disconnected mode.<br><br>A copy of each transaction is stored in the Local Transaction Log (similar to the Master Transaction Log on the server side). |
| 7 | When the user starts the synchronization process, the Remote Client creates DX files from the log and moves these DX files to the Remote client's outbox on the local machine. |
| 8 | The synchronization process starts when the Remote client initiates a synchronization session.<br><br>Synchronization Manager must be running and will authenticate the Remote client, based upon the type of authentication in the Component Parameters.<br><br>The synchronization process includes handling communication between the Mobile Web Clients and the file system.<br><br>The process moves the DX files from the docking directory outbox to the Remote client's inbox. It will move files from the Remote client's outbox to the docking directory's inbox.<br><br>Any attachments, correspondence, or templates that the Remote client creates are copied to the Siebel File System. |
| 9 | Changes do *not* occur in the server database until the synchronization session finishes and the Mobile Web Client disconnects.<br><br>Depending on the user synchronization options or command-line options chosen, the Remote client either begins applying the DX files to the local database as soon as the first transaction file is received, or after it completes the file exchange with the server. |
| 10 | Transaction Merger, a component on the server side, pulls the DX files from the inbox within the Docking Directory. |

Table 2. Steps in the Siebel Remote Flow Diagram

| Step | Explanation of the Diagram |
|---|---|
| 11 | Transaction Merger applies the changes to the server.<br><br>It also identifies conflicts.<br><br>A setting for System Preference, MRG: System Conflict Resolution, specifies whether the server or client wins during conflict resolution. The value can be Server Wins or Client Wins.<br><br>Conflicts will be communicated to the Remote user during the next synchronization.<br><br>Transaction Merger deletes the DX files from the inbox within the Docking Directory. |

Table 3 lists notes on the Siebel Remote flow process (Figure 2 on page 18).

Table 3. Siebel Remote Flow Diagram Notes

| Topic | Comments |
|---|---|
| Remote Configuration File | DockConnString is the name of the computer where the Siebel server software is installed and where the mobile client was extracted. DockConnString should be populated when the client is first initialized.<br><br>An important configuration parameter, AutoStopDB, is part of the Local configuration. Its default is FALSE, which means the local database engine will keep running after Siebel eBusiness Applications shut down. This will help shorten the startup time when the user restarts Siebel eBusiness Applications. If AutoStopDB is set to TRUE, the local database engine will shut down automatically when the Siebel eBusiness Applications shut down.<br><br>In the local database connecting string, -q means the local database is started in quiet mode. This prevents a SQL Anywhere window from showing up. However, a SQL Anywhere icon will appear on the Windows task bar, whether or not the -q parameter is used. This provides the user with a way to manually stop the database engine, if the user leaves the engine running after Siebel eBusiness Applications shut down.<br><br>On the local database connect string, the - m means truncate transaction log after checkpoint; -x NONE means do not load any network drivers; -gp 4096 tells the engine that the database page size is 4,096 bytes;<br>-c40m -ch60m sets the initial cache size to 40 MB, with a maximum of 60 MB. The cache sizes are suggested values that the user can adjust. However, the SQL Anywhere engine will determine the actual cache size within the given range of values.<br><br>Users can specify the sorting order on Local and Sample databases. To enable this feature, users modify their CFG file (for example, siebel.cfg, uagent.cfg). The SortCollation parameter in the CFG file determines the sorting order of the SQL Anywhere database. For best performance, it is recommended that SortCollation be set to Binary. For more details about this topic, see *Siebel System Administration Guide*. |

Table 3. Siebel Remote Flow Diagram Notes

| Topic | Comments |
|---|---|
| Authentication method | A Synchronization Manager component parameter that identifies the type of authentication that is used during synchronization. Values include:<br><br>**None.** Does not validate the synchronization authentication credentials. This is the default.<br><br>**Database.** Validates the synchronization authentication credentials against the server database user ID and password.<br><br>**Siebel.** Validates the synchronization authentication credentials against the Mobile Client name and Sync Password that is stored in the server database and is maintained for each Mobile Web Client in the Administration - Siebel Remote > Mobile Clients view under the appropriate parent server.<br><br>**AppServer.** Validates the synchronization authentication credentials against the user ID and its password on the Siebel Server operating system.<br><br>**SecurityAdapter.** Validates the synchronization authentication credentials against a third-party authentication system. |
| Synchronization Frequency | Siebel Systems recommends that mobile users synchronize against the server at least daily. There is an automatic mechanism called TrickleSync to help enforce frequent synchronization. Users can invoke TrickleSync on their laptops or administrators can set up the feature for their mobile users. For more detail about this topic, see "Using TrickleSync" on page 76. |

Figure 3 illustrates the Remote downflow of data. See Table 2 on page 19 for explanation of the numbered steps.

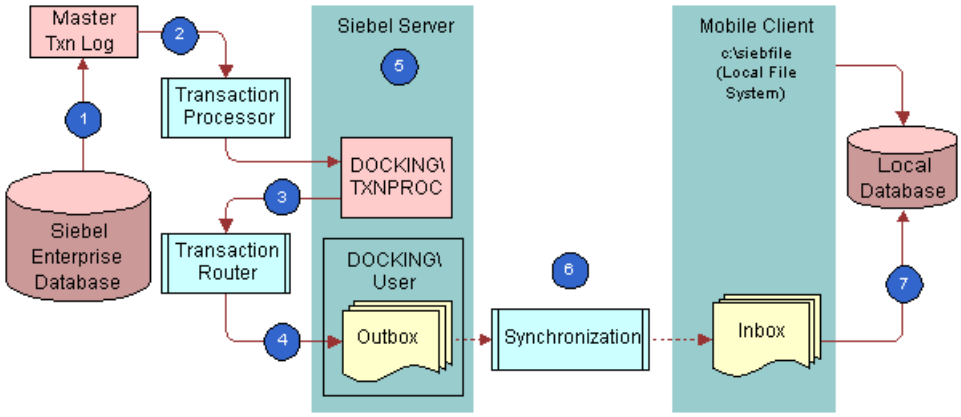


Figure 3. Remote Down Flow Diagram

Figure 4 illustrates the Remote upflow of data transactions. See Table 2 on page 19 for explanation of the numbered steps.

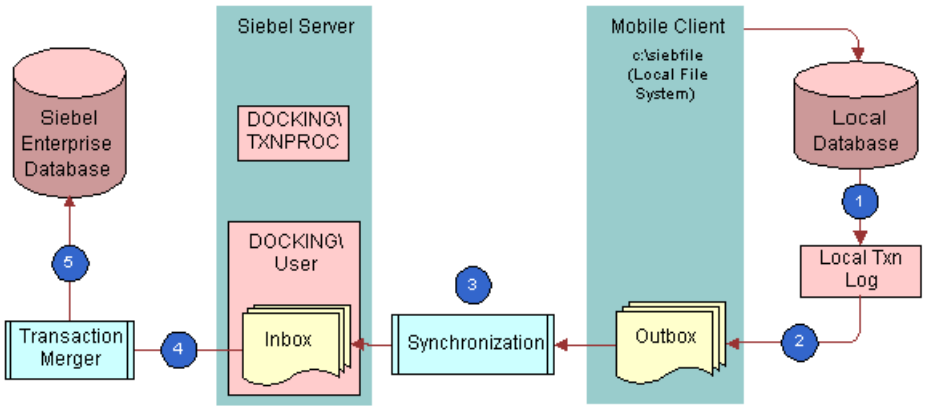


Figure 4. Remote Up Flow Diagram

# Siebel Remote Data Store Components

This section describes the various components that store data used by Siebel Remote:

- "Siebel Database Server"
- "Siebel File System" on page 25
- "Local Database and File System" on page 25

## Siebel Database Server

The database server stores data for users, with both stationary and Mobile Web Clients.

The database server contains:

- Siebel applications metadata
- Siebel applications tables that store user data
- A master transaction log table that stores changes made since the last database extraction, provided that the changes are accomplished using Siebel software. Direct SQL modification of Siebel tables is not supported. SQL changes that are executed outside the Siebel application are not recorded in the Master Transaction Log and therefore will not be routed to mobile clients.

The Transaction Processor and Transaction Router components of Siebel Remote, which run on the Siebel Remote server, route the transactions from the master transaction log to the outbox directories of Mobile Web Clients. The Transaction Processor purges rows from the log after transactions from S_DOCK_TXN_LOG have been written into DX files in the *txnproc* directory. Do *not* directly modify the contents or structure of the master transaction log under any conditions.

**CAUTION:** In a Microsoft SQL Server or DB2 environment, truncation of the S_DOCK_TXN_LOG table in a Server Database causes the txn_id values assigned to new transactions to be reset. However, the corresponding txn_id values in the S_DOCK_STATUS table are not reset. Consequently, new transactions inserted into the S_DOCK_TXN_LOG table will have a txn_id mismatch as compared to the values in the S_DOCK_STATUS table. The Transaction Processor will not process these transactions and will purge them from the S_DOCK_TXN_LOG table.

If the S_DOCK_TXN_LOG table is truncated, then all Mobile Web Clients must be reextracted and all existing transactions and local databases must be deleted before the these clients can be reinitialized. The re-extraction process resets the txn_id values in the S_DOCK_STATUS table, so that the values correspond with txn_id values in the S_DOCK_TXN_LOG table. The deletion of transactions and local databases is necessary to prevent mismatch errors between new S_DOC_STATUS txn_id values and older txn_id values in the local databases.

**CAUTION:** After the installation of the Siebel database server, if your Siebel implementation uses both Siebel Remote and Oracle RAC, then you must disable the cache for sequence S_DCK_TXN_LOG_S for each instance of Oracle RAC. If this instruction is not followed, transaction ID gaps can result, causing Siebel Remote failures. For more information about this requirement, see *Siebel Installation Guide* for the operating system you are using.

## Siebel File System

The Siebel File System stores attachments, correspondence, templates, and other types of unstructured data for Siebel users through the File System Manager (FSM). Connected users access files from the Siebel File System. The Siebel Remote server can transfer files between mobile users and the File System during synchronization.

The File System Manager (FSM) server component manages the Siebel File System and handles interplatform security. FSM handles most of the interaction with the Siebel File System within the Siebel applications. However, synchronization manager also interacts with the Siebel File System during synchronization sessions. Web Clients, Dedicated Web Clients, and server components requiring operations on files make requests, through the Server Request Broker, to the FSM.

For more information see *Siebel System Administration Guide*.

## Local Database and File System

Mobile Web Clients use a local database to store data for user access. The local database contains Siebel application tables that store user data.

The local database also contains a local transaction log to store transactions created by the mobile user. Siebel Remote forwards these transactions to the Siebel Remote server when the client synchronizes. Do not directly modify the local transaction log under any circumstances. The Siebel Remote synchronization client automatically purges the local transaction log table when appropriate.

**NOTE:** Users should run only one instance of a local database at any given time. In addition, users should defragment their hard drives regularly to optimize performance. For further instructions on the defragmentation process, follow your local policies.

Also, the local database is designed for only one user and does not support multiple logins to a single remote database.

Mobile Web Clients also use a local Siebel File System to store files from the Siebel File System. Mobile users can request specific files to download to their local File System during synchronization. Also, the Siebel administrator can specify files that should be published or distributed to mobile users. The Siebel Remote server retrieves the files from the Siebel File System and sends them to the local Siebel File System. These files are available when the mobile user disconnects from the Siebel Server.

Files that mobile users add to their local File Systems while disconnected are uploaded to the Siebel Remote server during synchronization. Then the Siebel Remote Server sends these files to the Siebel File System.

# Siebel Remote Client Software

The Siebel Remote client software runs on the Siebel Mobile Web Client and manages the synchronization process between the client and the Siebel Remote server.

Mobile users can start the Siebel Remote client in two ways:

- Background synchronization

  While the Siebel client is running, mobile users can choose File > Synchronize Database from the application-level menu. This launches the Siebel Remote client as a background process so that work can continue within Siebel applications or in other Windows applications during the synchronization process.

- Stand-alone synchronization

  Mobile users can launch Siebel Remote in stand-alone synchronization mode using the Siebel Remote icon or a third-party scheduling program. This allows users to synchronize without starting the Siebel application. The TrickleSync functionality also uses this method to perform synchronization when it starts a synchronization session.

The Siebel Remote client uses the TCP/IP networking protocol to communicate with the Siebel Remote server using a local area network (LAN), wide area network (WAN), virtual private network (VPN), or a modem using dial-up networking.

The Siebel Remote client connects to the Siebel Remote server; the Siebel Remote server then starts a synchronization session for the Mobile Web Client. Transactions (DX files) and file attachments are sent down to the client machine from the server and up from the client machine to the server.

This synchronization process includes the following stages:

1. Connects to the Siebel Remote Server, which starts a Synchronization Manager session for the Mobile Web Client.
2. The Synchronization Manager notifies the mobile user if the Siebel administrator has performed a database extract for the Mobile Web Client or if the local database does not exist.
3. Extracts transactions from the local database.
4. Sends and receives transaction files to and from Synchronization Manager.
5. Sends and retrieves file attachments.
6. Applies transaction files from the Siebel Remote Server to the local database.
7. Applies transaction files from the local database to the server database.

For more detailed information about how information is handled during a synchronization, see "Siebel Remote Flow Diagram" on page 18.

**CAUTION:** In order to operate correctly, each Siebel Remote Client installation must be used with exactly one local database. Do not rename directories that contain Siebel Remote Client software, or attempt to use additional local databases. Do not rename or delete DX files.

**NOTE:** Transactions are not lost after a client reextract. For information about the Save Client Transactions functionality, see "Database Extraction for a Mobile Web Client" on page 103.

# Siebel Remote Server Components

This section discusses the Siebel Remote server components that operate on the Siebel Server and provides an overview of the administration tasks you need to perform for each component.

# Creating Siebel Server Directories for Mobile Web Clients

Each registered Mobile Web Client requires a separate directory on the Siebel Remote server. The Database Extract program creates the appropriate directory and its subdirectories for each Mobile Web Client.

**NOTE:** The installation program also places a directory called txnproc in the docking subdirectory within the Siebel server root directory. Do *not* modify the contents of this directory under any circumstances.

The following example shows a portion of the server directory tree after you run Database Extract for Mobile Web Clients named Adams and Scott:

```
siebel
   docking
      adams
         inbox
         outbox
      scott
         inbox
         outbox
      txnproc
```

# Generate New Database

The Generate New Database component creates the local database template for a given database schema version. The component reads the database schema definition from the Siebel repository, then creates Siebel tables and indexes in a database template file stored on the Siebel Remote server in the dbtempl subdirectory.

The Local Database Initialization program uses the local database template when initializing a new database on the Mobile Web Client.

**NOTE:** Dicdata.dat and diccache.dat are the same dictionary file. Dicdata.dat is named differently for Mobile Web Clients. These files store metadata about schema definitions, vis-rules, and other items.

The Generate New Database component copies diccache.dat to the dbtempl\\dicdata.dat file. The dicdata.dat file is downloaded to Mobile Web Clients and used by the synchronization process whenever transactions are applied to the local database.

If you need to interchange the dicdata.dat file between Mobile Web Clients for different languages, contact Siebel Technical Support to determine if the languages are interchangeable.

## Administration of the Generate New Database Component

You must generate a new database template whenever the Siebel database schema changes in cases such as:

- Immediately upon installing the Siebel database server
- Following an upgrade to a new version of Siebel applications
- Extending the database schema using Siebel Tools—except when using Siebel Anywhere to deliver a database schema upgrade kit

If your deployment requires a different collation template not provided by Siebel eBusiness Applications, please contact Siebel Expert Services for assistance in creating a new collation template.

### Optimal Size for Local Databases

The recommended size for the SQL Anywhere local database depends upon several factors. These include the Mobile Web Client user's position and responsibilities. Also, the Data Routing Model assigned to the client impacts the volume of data to be stored in the local database. The local database should not be larger than 700 MB. For further information on this subject, or if a local database will exceed this number, contact Siebel Systems.

## Database Extract

The Database Extract component creates a database snapshot file for a given mobile user. The file contains the data required to initialize the user's local database. Database Extract retrieves data according to routing rules that define the level of access to information for each Mobile Web Client.

## Synchronization Manager

The Siebel Remote server starts a Synchronization Manager task for each incoming synchronization request from a Mobile Web Client. For each request, the Synchronization Manager:

- Verifies the Mobile Web Client status and *credentials* (user ID and password), provided that Siebel Remote authentication is enabled
- Transfers the local database template and local database extract if applicable
- Exchanges transaction files
- Transfers file attachments to and from the Siebel File Server

Each Synchronization Manager task services only one Mobile Web Client at any one time, but many synchronization tasks can be started concurrently. This behavior is configured by a Synchronization Manager parameter called Max Task. The Synchronization Manager component must be enabled for Siebel Remote mobile users to be able to connect to the Siebel Remote server for synchronization. Synchronization Manager tasks start automatically; you do not need to start tasks for this component manually.

**NOTE:** The Siebel File System parameter for Synchronization Manager component determines the file system location for Mobile Web Clients.

# Transaction Processor

The Transaction Processor component scans the master transaction log on the Siebel database server and prepares transactions for visibility checking and routing by a transaction router. Transaction Processor leaves the last log entry on DB2 and leaves 1000 on MS SQL to avoid deadlock problems.

You can run only one transaction processor on each Siebel Remote server. Upon start-up, the transaction processor verifies that another transaction processor is not running on the same Siebel Remote server.

**CAUTION:** VARCHAR data handled by Siebel Remote must not be larger than 16,350 bytes. VARCHAR data that exceeds this size will cause a Transaction Processor error and stop Siebel Remote operations. If this occurs, contact Siebel Technical Support for assistance in resolving the situation. To prevent this from occurring, configure your implementation to enforce the size restriction for LONG data at the business component layer or use file attachments to store the data.

# Transaction Router

The Transaction Router performs visibility checking and routing of transactions from the DX files created in the txnproc directory by the Transaction Processor.

## Dock Objects and Routing

Routing rules determine the subset of Dock Object instances that Siebel Remote replicates to each Mobile Web Client. Dock objects are groupings of tables in the database that logically form Siebel business components. Dock objects are similar to business components.

Each dock object is classified according to a visibility level. The three classes are:

- Enterprise
- Limited
- Private

For an Enterprise visibility dock object, Siebel Remote sends all data in the object to the Mobile Web Client. For a Limited visibility object, Siebel Remote sends the object to a Mobile Web Client only if the instance is visible to the mobile user. Private dock objects are not routed to a Mobile Web Client from a Siebel server.

Siebel applications provide preconfigured user routing rules. As delivered, the Siebel routing rules encompass a combination of implicit security rules (based on responsibility) and assignment rules that determine a user's access to information.

If your deployment contains a large number of high level objects such as Assets, Accounts, or Activities, you may want to contact Siebel Expert Services to help reconfigure certain routing rules for optimal performance.

In Siebel 7.7, some Dock Objects were added, and some were dropped from those included in Siebel 7.5. Also, some visibility levels were changed. See Appendix B, "Docking Object Changes" for a listing of detailed changes in Dock Objects.

### Administration of the Transaction Router

You must run at least one transaction router on each Siebel Remote server. For better performance, you should run multiple transaction routers on the same Siebel Remote server.

**NOTE:** Changing the definition of organizations (positions and divisions) can cause routers to reevaluate visibility for objects related to the objects that have changed. This can affect the performance of the Transaction Router. To alleviate this situation, reextract all your Mobile Web Client databases.

## Transaction Merger

The Transaction Merger component applies transactions to the Siebel database server that were uploaded into the appropriate Application server inbox by a Siebel Remote mobile user. The application of these transactions to the Siebel database takes place *after* the synchronization session finishes and the Mobile Web Client disconnects.

You must run at least one transaction merger on each Siebel Remote server. For better performance, you can run multiple transaction mergers on the same Siebel Remote server.

# Using Siebel Remote

This section provides a high-level overview of the logical steps involved in extracting, initializing, and synchronizing a Siebel Remote mobile user. Procedures for completing each step are covered in other chapters in this guide.

- "About Registering a Mobile User"
- "About Generating a Database"
- "About Extracting a Mobile Web Client" on page 32
- "About Initializing the Local Database" on page 32
- "About Synchronizing a Mobile Web Client" on page 33
- "How Changes Are Propagated to and from a Mobile Web Client" on page 36
- "Synchronizing a Local Database with the Enterprise Database" on page 40

## About Registering a Mobile User

Before a mobile user is registered, that person must be in the system as a user. Mobile users typically operate in disconnected mode and synchronize frequently to keep their local databases in balance with the Server database. For information on adding users, see *Applications Administration Guide*.

## About Generating a Database

If you are setting up a Siebel Remote server for the first time, you will also need to generate a new *database template*, that is, a database that contains no end user data, but which has the specific structure required for use with Siebel applications. You can generate a new database template by running the Generate New Database component. On the Application server, this component creates a template and places it in the dbtempl directory on the Application server. For information on generating a new database template, see "Generating a New Database Template" on page 67.

## About Extracting a Mobile Web Client

The first step in creating a local database for a new mobile user is to extract the database on the Siebel Remote server. Extract Mobile Web Clients by running the Database Extract component. This component extracts visible data for the mobile user into a snapshot file in the Mobile Web Client's outbox directory on the Application server.

**NOTE:** Each local database is a different entity. Therefore, a user cannot use multiple machines as the same Remote client.

## About Initializing the Local Database

When users log in and specify the Local database as the data source the first time, Siebel Remote will detect that a local database does not exist. This prompts the user to connect to the Siebel Remote server in order to retrieve it.

This process performs the following tasks:

- **Prompts for Mobile Web Client connection information.** Enter the user ID, the Mobile Web Client name (a given user may have multiple Mobile Web Clients, such as two clients on two separate computers), and a confirmation password. If the database initialization process was started by starting the Siebel Mobile Web Client, then the user ID defaults to the value that the user provided when the Mobile Web Client was started.
- **Connects.** If specified by the user, the synchronization client dials the modem to connect to the application server. The synchronization client connects to the Siebel Server—if the user has access to this server and the Synchronization Manager port is open. Then a new synchronization thread services the Mobile Web Client.

  For example, a VPN setup may require the user to enter a special PIN and security token number such as an RSA.
- **Validates Mobile Web Client.** The Synchronization Manager validates the Mobile Web Client's node name against the list of valid users in the server database. The Synchronization Manager also checks that the Mobile Web Client is connected to the correct Siebel Remote server. Finally, if Siebel Remote authentication is enabled, the Synchronization Manager authenticates the Mobile Web Client's credentials.

- **Checks for database extract.** The Synchronization Manager verifies that a database extract is pending in the application server outbox. During this verification, the Synchronization Manager checks for UAF files (.uaf) and TOC files (.toc) in the mobile user's outbox subfolder. If none are there, the synchronization client asks the user to contact the Siebel administrator to perform a database extract.
- **Downloads snapshot and file attachments.** If the UAF and TOC files are present, the mobile user will be prompted to download a new database during the synchronization session. The synchronization client downloads the extract and file attachments to the Mobile Web Client's inbox directory.
- **Disconnects.** The synchronization client closes the connection with the Synchronization Manager. The mobile user can disconnect from the network at this point.
- **Creates and loads database.** The synchronization client shuts down the Siebel client or Siebel Remote program and starts the Siebel Upgrade Wizard. The Siebel Upgrade Wizard creates a new local database, loads data from the extract file into the local database, and applies file attachments to the local File System.

Figure 5 shows the processes that occur when a local database is initialized.

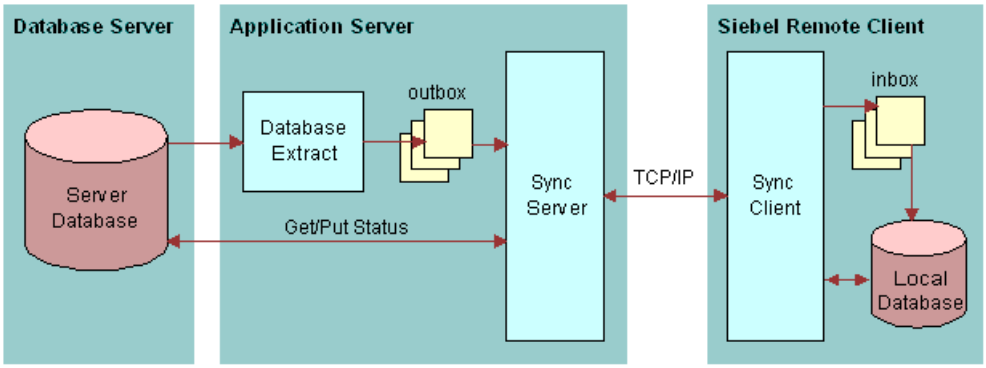


Figure 5. Initialize Local Database

For additional details on initializing the Mobile Web Client database, see "Initializing a Mobile Web Client Database" on page 111.

**NOTE:** In order to load the Mobile Web Client faster, you can enable Siebel QuickStart. Check the box titled Enable Siebel QuickStart on the Siebel Login Screen for the Mobile Web Client. Siebel.exe will be preloaded to the memory at the system login time. Then every subsequent client session will call the existing process instead of creating a new one, until the user disables QuickStart. For more information about QuickStart, see "Siebel Mobile Web Client and Siebel QuickStart" on page 121.

# About Synchronizing a Mobile Web Client

This section describes the processes for synchronizing a Mobile Web Client.

## Routing and Merging

On the Siebel Remote server, the Transaction Router and Transaction Merger components continuously route and apply transactions for Mobile Web Clients. These two tasks process data asynchronously from the synchronization sessions in order to minimize the connection time between the Mobile Web Client and the Siebel Remote server.

- **Transaction router.** One or more transaction router tasks on the application server continuously route outgoing transactions (in the txnproc directory created by the Transaction Processor) to the Mobile Web Client's outbox folders.
- **Transaction merger.** One or more Transaction Merger tasks on the application server continuously merge incoming transactions from the Mobile Web Client's inbox directories to the server database and the Siebel File System.

Figure 6 shows the processes that occur when a Mobile Web Client is synchronized.

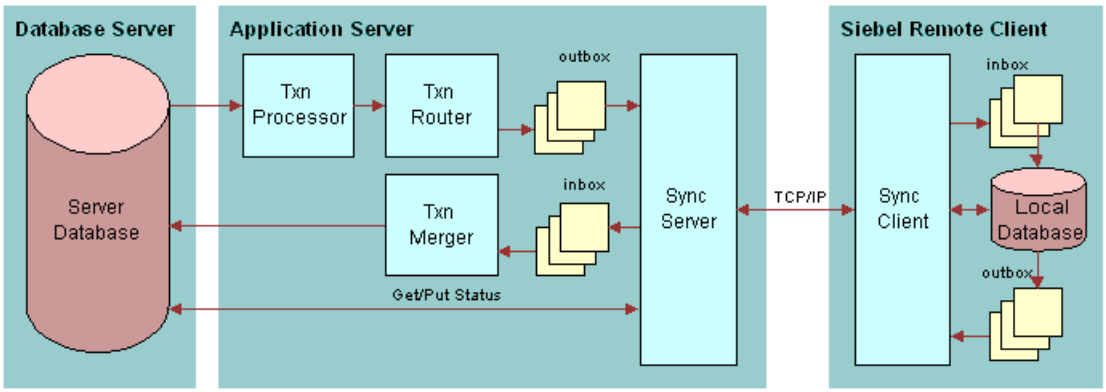


Figure 6. Synchronizing a Mobile Web Client (Routing and Merging)

## Synchronization Session

To synchronize an existing Mobile Web Client, the mobile user launches Siebel Remote, either from within the Siebel Mobile Web Client or in stand-alone mode. Siebel Remote executes the following steps:

- **Connects.** If specified by the user, Siebel Remote dials the modem to connect to the Siebel Remote server, using connection information from the Windows phone book. Once network connectivity is established, the Siebel Remote client connects to the Siebel Server. If the user is already connected through a LAN or WAN connection, then it merely performs a handshake to validate a network connection exists.

- **Validates Mobile Web Client.** The Synchronization Manager validates the Mobile Web Client's name against the list of valid users in the database server.

  The Synchronization Manager also verifies that the Mobile Web Client is connected to the correct Siebel Remote server. If this is not the case, the Synchronization Manager reconnects the Mobile Web Client to the appropriate Siebel Remote server and updates the client's local configuration information.

  Finally, if Siebel Remote authentication is enabled, the Synchronization Manager authenticates the Mobile Web Client's credentials.

- **Check for correct version**. The Synchronization Manager checks, against the server, that the Mobile Web Client is running with the most up-to-date version of the application. If not, it will prompt the user to download a new version of the application.

- **Checks for database extract.** The Synchronization Manager checks whether a database extract is pending for the Mobile Web Client. If not, the synchronization session continues. If there is a database extract pending, the synchronization client reinitializes the Mobile Web Client, using the same process described in "About Initializing the Local Database" on page 32. Then another synchronization session begins.

- **Retrieves transactions and file attachments.** The client retrieves transaction files—created by the Transaction Router—from the Mobile Web Client's outbox directory on the Siebel Remote server and stores the transaction files in the Mobile Web Client's local inbox directory. The client also retrieves requested, published, or broadcasted file attachments from the Siebel File Server.

- **Sends transactions and file attachments.** The Siebel Remote client extracts pending transactions from the local transaction log into transaction files, and sends the transaction files to the user's inbox directory on the Siebel Remote server.

- **Applies changes to the server database.** Transaction Merger applies the incoming transaction files from the Mobile Web Client's inbox directory on the server to the server database and Siebel File Server.

- **Applies changes to the local database.** The Siebel Remote client applies the incoming transaction files from the Mobile Web Client's inbox directory on the Mobile Web Client to the local database and applies retrieved file attachments to the local file system. The mobile user can use the Siebel client while the Siebel Remote client applies the changes to the local database.

  **NOTE:** The timing for applying the incoming transactions depends on the options chosen. By default, the client begins applying transactions as soon as the first transaction file has been downloaded. However, you can use command line options or user synchronization options to direct the client to apply transactions only after downloading is complete, or to postpone applying transactions until another time. For more information about command line options, see "Enabling the Stand-Alone Synchronizer" on page 113. For more information about user synchronization options, see "User Synchronization Options for Mobile Web Client" on page 117.

- **Disconnects.** Siebel Remote then closes the connection with the Siebel Remote server. If the Siebel Remote client created the network connection automatically, it also disconnects the modem; otherwise, the mobile user can manually disconnect from the network at this point.

- **Cleans up.** This refers to those files (.dx) that were successfully applied during the *previous* synchronization session. The Siebel Remote client deletes the transaction files in the Mobile Web Client's local outbox directory that the Transaction Merger successfully applied to the database server during the *previous* synchronization session. On the Siebel Remote server, the Synchronization Manager deletes the transaction files in the Mobile Web Client's outbox directory that were successfully applied to the local database during the *previous* synchronization session.

# How Changes Are Propagated to and from a Mobile Web Client

Between synchronization sessions, the Siebel Remote server prepares transactions applied to the database server by other users—mobile and dedicated. Siebel Server components write the transactions to a separate directory for each mobile user. These transactions, combined with items from the File System, are downloaded to the Mobile Web Client during the next synchronization session. Items from the File System include updated, published, or requested marketing literature, correspondence templates, and other types of file attachments.

A similar process occurs on the client as well, although without the server component.

## Process Flow for Changes by Connected Users

This section describes the process flow for downloading changes on the server database to local databases. The flow takes place from the time a connected user creates a new opportunity until it appears in a local database.

This specific process flow includes an example of a telesales representative in a Call Center. The telesales person talks to potential customers responding to a new advertising campaign.

The telesales person decides to create a new opportunity record for one of the more promising responses. Figure 7 illustrates this flow. The numbers in the diagram correspond to the list immediately following the diagram.

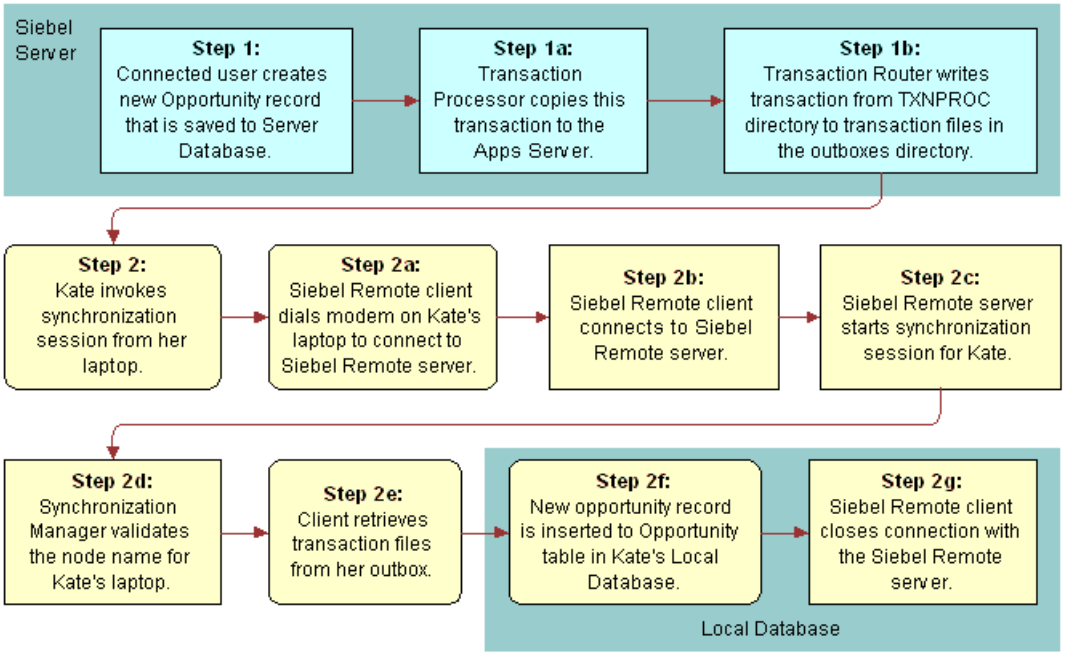


Figure 7. Process Flow for Changes by Connected Users

The process flow for the example above includes the following steps:

1. The telesales person creates a new opportunity record—a transaction saved in the opportunities table on the server database. A copy is saved to the master transaction log.
   a. Transaction Processor copies this transaction, and others, to the Apps Server.
   b. A Transaction Router task writes each transaction from the TXNPROC directory to transaction files in separate outbox directories for each Mobile Web Client.

      The Mobile Web Client outbox directories are stored on the Siebel Remote server.
2. A mobile user invokes a synchronization session from the laptop.

   During the synchronization session, the following steps occur to download the new Opportunity record to the mobile user's local database:
   a. The Siebel Remote client on the laptop dials the modem to connect to the Siebel Remote server.

      This can be handled by an existing modem or LAN connection.
   b. The Siebel Remote client connects to the Siebel Remote server.
   c. The Siebel Remote server starts the synchronization session for the Mobile Web Client.

**d** The Synchronization Manager validates the Mobile Web Client's node name against the list of valid mobile users in the server database.

The Siebel Remote client receives transaction files going to and from the Synchronization server, and retrieves file attachments.

**e** The client retrieves transaction (.dx) files from the user's outbox directory on the Siebel Remote server and stores them in the mobile user's local inbox directory. The client also retrieves the user's requested, published, or broadcasted file attachments from the Siebel File Server.

**f** The new opportunity record is inserted into the Opportunity table in the user's local database.

**g** The Siebel Remote client closes the connection with the Siebel Remote server.

The mobile user can use the Siebel client while the Siebel Remote client applies the changes to the local database.

**CAUTION:** Users should never directly modify the local transaction log. The Siebel Remote synchronization client automatically purges the local transaction log table.

## Process Flow for Changes Made by Mobile Users

Mobile Web Clients use a local database to store data for user access. The local database contains Siebel eBusiness Applications tables that store user data. The local database also contains a local transaction log to store transactions created by the mobile user. Siebel Remote forwards these transactions to the Siebel Remote server when the client synchronizes.

This section provides a description of each phase of the process flow, from the time when the mobile user modifies the new opportunity until the time when the modifications appear in the Server database. For this example, assume that a mobile user is meeting with a potential new client, represented by the opportunity record entered by the telesales representative in the previous section.

Figure 8 illustrates this flow. The numbers in the diagram correspond to the list following the diagram.

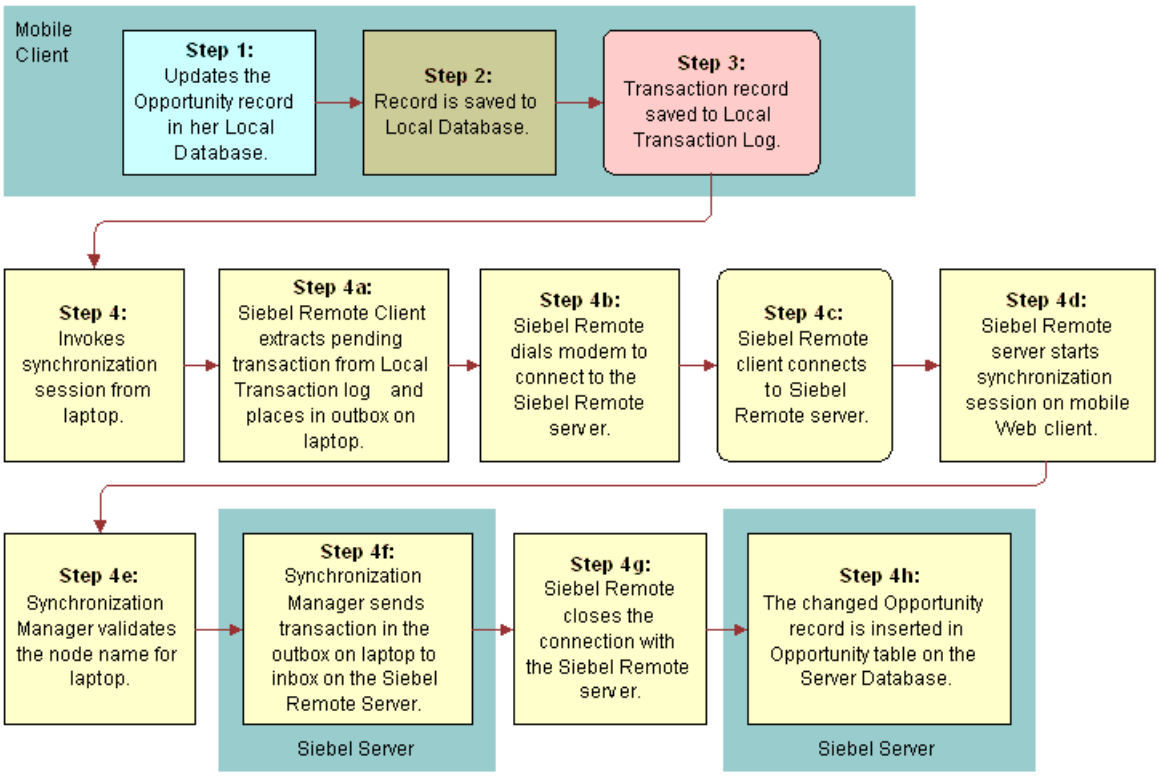


Figure 8. Process Flow for Changes by Mobile Users

1. As a result of the meeting, the mobile user makes changes to the Opportunity record in the local database on the laptop. The user enters these changes immediately after the meeting while working offline.
2. The modified opportunity record is saved to the Opportunities table in the local database. A transaction record is saved to the Local Transaction log.
3. The Siebel Remote client extracts pending transactions from the Local Transaction log into transaction files (.dx). The client then places these DX files in the outbox directory on the laptop.
4. The mobile user synchronizes the laptop.

   During the synchronization session, the following steps occur to record the changes to the opportunity record:

   a. Siebel Remote dials the modem to connect to the Siebel Remote server.

      If the user is in the office, a LAN connection will also work.

   b. Siebel Remote client connects to the Siebel Remote server.

   c. Siebel Remote server starts the synchronization session for the mobile user.

**d** Synchronization Manager validates the Mobile Web Client's node name against the list of valid Mobile Web Clients in the server database.

**e** Synchronization Manager sends the transaction files in the outbox directory on the laptop to the mobile user's inbox directory on the Siebel Remote server.

**f** Siebel Remote closes the connection with the Siebel Remote server.

**g** The changed Opportunity record is inserted into the Opportunity table on the Server database.

## File Attachments and Siebel Remote

This section provides additional information about how Siebel Remote handles file attachments.

When a Siebel Remote client attaches a file to a record in the local Siebel database, the metadata concerning the file is stored in the local Siebel database, and the file is stored in the local Siebel file system. When the client synchronizes with the Siebel Remote Server, normal synchronization procedures result in the file being copied to the server's Siebel file system, and metadata concerning the file being copied to the server's Siebel database.

If the record to which the file is attached is accessible to another Siebel Remote client, such as another member of a sales team, then ordinarily file metadata is transferred to that team member's local Siebel database when that team member next synchronizes. However, the file attachment, itself, usually is not transferred to the local Siebel file system until it is specifically requested. (The exception is that certain types of file attachments, such as Literature items, have a Distribution Method setting. If Distribution Method is set to Publish, the file is automatically sent to Siebel Remote users, without needing to be requested.)

A Remote client requests a file attachment by clicking its link in a Siebel application, or by selecting its Request File check box. When this is done, a request for the file is queued. In order to actually receive the file, the client must synchronize again, which will result in the file being copied to the local Siebel file system. Once the synchronization is complete, the client must click the link again to view the file.

# Synchronizing a Local Database with the Enterprise Database

Mobile users must synchronize frequently to obtain and view possible updates in the server database. Also, there may be updates to the store of documentation, marketing literature, and sales brochures in the file system.

The following procedure describes how to synchronize while using a Siebel application. Synchronization can also be done in stand-alone mode, using the Siebel Remote icon.

### *To synchronize a local database with the enterprise database*

1. Start your Siebel application, such as Siebel Sales, in the Siebel program group on the Mobile Web Client machine.
2. From the application-level menu, select File > Synchronize Database.

3 Choose the synchronization options.

For detailed information on synchronization options, see "User Synchronization Options for Mobile Web Client" on page 117.

4 Click Synchronize.

A progress indicator appears. In any synchronization that includes both upload/download and apply operations, the indicator initially shows the progress of the upload/download operation. When uploading and downloading is complete, the indicator shows the progress of the apply operation.

# Security and Authentication

This section discusses security and authentication for mobile users.



For general information about security issues, see *Security Guide for Siebel eBusiness Applications*.

## Authentication for Synchronizing the Local Database

The Siebel Remote Synchronization Manager authenticates incoming Mobile Web Client requests to make sure that a Mobile Web Client is valid.

The Synchronization Manager validates the following information when a Mobile Web Client synchronizes with the server database:

**Mobile Web Client name.** Synchronization Manager validates the Mobile Web Client's name against the list of valid Mobile Web Clients in the server database and validates that the effective end date is valid or NULL.

**Application server.** The Mobile Web Client attempts to synchronize on the server defined by the DockConnString parameter for the local database in the application's .cfg file on the Mobile Web Client. If DockConnString differs from the server defined by the App Server Name parameter defined in the Administration - Siebel Remote > Mobile Clients view, the Synchronization Manager will reroute the connection to the server defined by App Server Name, and Synchronization Manager will edit the application's .cfg file on the Mobile Web Client by resetting the DockConnString parameter to the App Server Name value.

**Credentials.** Synchronization Manager validates the Mobile Web Client's synchronization authentication credentials, provided that the Authentication Method parameter for Synchronization Manager is *not* set to None. The nature of the credentials varies depending on whether or not the local database has been initialized:

- Prior to local database initialization, the user ID and password that the user provided at login are used as the synchronization authentication credentials.

- Following local database initialization, the user ID and password from the User Preferences > DB Synchronization view on the Mobile Web Client are used as the synchronization authentication credentials.

Synchronization Manager uses one of the following methods to authenticate the synchronization credentials. The Siebel administrator specifies the method by entering one of the following method names as the Authentication Method parameter value for the Synchronization Manager server component.

- **None.** Does not validate the synchronization authentication credentials. This is the default.
- **Database.** Validates the synchronization authentication credentials against the server database user ID and password.
- **Siebel.** Validates the synchronization authentication credentials against the Mobile Client name and Sync Password that is stored in the server database and is maintained for each Mobile Web Client in the Administration - Siebel Remote > Mobile Clients view under the appropriate parent server.
- **AppServer.** Validates the synchronization authentication credentials against the user ID and its password on the Siebel Server operating system.
- **SecurityAdapter.** Validates the synchronization authentication credentials against a third-party authentication system.

## Providing Security Adapter Authentication for Synchronizing the Local Database

An authentication system is a data store outside the Siebel Database in which the information that is required to allow users to connect to a data source or to a server component is maintained. This information typically consists of database accounts and Siebel user IDs.

A security adapter is a plug-in to the authentication manager. The security adapter uses the credentials entered by a user or supplied by an authentication service to retrieve the Siebel user ID, a database account, and, optionally, a set of roles from the authentication system.

Security adapters can be used in authentication of Mobile Web Client synchronization sessions. For information about other authentication methods for synchronization, see "Authentication for Synchronizing the Local Database" on page 41.

For context-independent guidelines and instructions for setting up security adapter authentication, see *Security Guide for Siebel eBusiness Applications*.

To implement security adapter authentication for synchronizing the local database, be aware of the following considerations:

- The Security Adapter Mode and Security Adapter Name parameters specify the type of security adapter authentication and the name of the security adapter profile (named subsystem), respectively. These parameters can be set for the enterprise, for a server, or for a component. Settings that are made at lower levels, such as the component level, override settings made at higher levels, such as the enterprise level. If you set these parameters at the component level for use with synchronization, the component to set is Synchronization Manager.

- The Synchronization Manager component accepts requests for synchronization and manages the synchronization session. As such, it is the component that delivers credentials to the security adapter and receives the user's database credentials from the security adapter.
- Web Single Sign On is not supported for synchronization.
- The same directory or different directories can be used for synchronization authentication as for authentication against other data sources or components.

## Authentication Against the Local Database

A Mobile Web Client user is typically authenticated before connecting to the local database.

The Mobile Web Client only supports database authentication through the database security adapter on the Mobile client.

**CAUTION:** The Mobile Web Client does not support third-party or customer proprietary authentication methods.

The password in the local database is set in one of the following ways:

- During client initialization, the password assigned to the local database is the same as the password assigned by an administrator at the server level for synchronizing the local database.
- The user may assign a different password for local database access.

  For information about changing the local database password, see "Configuring the Mobile Web Client for Local Database Password Encryption" on page 83.

An administrator can opt for the local database password to be encrypted before being sent to the database for authentication. For information about RSA SHA-1 encryption of the local database password, see "Configuring the Mobile Web Client for Local Database Password Encryption" on page 83 and "Configuring the Remote Server for Local Database Password Encryption" on page 105.

# Locking and Concurrency

The following are concurrency rules and behaviors of the Siebel Remote components:

- A transaction processor can run while any other Siebel Remote server component is running.
- There can be only one transaction processor for each Applications server.
- Only one instance of the Transaction Router, Transaction Merger, Synchronization Manager, or Database Extract can process a given Mobile Web Client at any one time. Two transaction routers, therefore, *cannot* route transactions to the same Mobile Web Client at the same time, nor can two transaction mergers merge transactions from the same Mobile Web Client at the same time. Multiple instances of these components can be running on one Applications server for better performance.
- The Transaction Router, Transaction Merger, or Synchronization Manager cannot process a Mobile Web Client while a database extract is in progress for that Mobile Web Client.

- The Transaction Router, Transaction Merger, and Synchronization Manager, on the same server, can process the same Mobile Web Client at the same time.
- Multiple instances of Database Extract running at the same time allow the extraction of multiple lists of mobile users, thus reducing the overall extract duration.

# Conflict Detection and Resolution

Siebel Remote supports team selling and field service by allowing different mobile users to access data. This creates the possibility that two users may make conflicting changes to the same data when they are disconnected from the server. Siebel Remote automatically detects update conflicts by comparing transactions at the field level. To accomplish this, Siebel Remote uses specific rules to resolve conflicts for three types of database data changes:

- Updating values in an existing row
- Deleting an existing row
- Adding a new row

More complex conflicts involve deleting and adding database rows. One user may change a value in a database row and another may delete the entire row. One user may add a database row to a local database, but the user primary key of the row may be in use by an existing row in the server's database or in another Mobile Web Client's database. Over time, if such changes go undetected or unresolved, the databases on various machines become less synchronized, a phenomenon called *data divergence.* Siebel Remote incorporates conflict detection and resolution logic to prevent data divergence between server and Mobile Web Client databases.

For more information about specific types of data conflicts, see the following topics:


- "Update Conflicts" on page 44
- "Insert Conflicts" on page 45
- "Delete Conflicts" on page 46
- "Merge Conflicts" on page 46


## Update Conflicts

An update conflict occurs if, for example, one user changes a contact's area code to 415 and another user changes it to 408. Siebel Remote detects and consistently resolves conflicts by using one of two rules: *Client Wins* or *Server Wins.* The Client Wins rule states that the Mobile Web Client database value overrides the server database value. The Server Wins rule states that the server database value overrides the Mobile Web Client database value. The default setting, Server Wins, is strongly recommended to resolve such conflicts because data changes will converge more quickly. The Client Wins rule will work as well but will take longer for data to converge.

**CAUTION:** To maintain integrity across databases, the same conflict resolution rule must be in effect for client and server databases. You should specify the rule as part of the initial Siebel Remote implementation on your database server, and before running a database extract for any client so that the rule is replicated consistently to the clients.

Siebel Remote applies the same logic to databases to make sure that they remain synchronized. During the initial Siebel Remote implementation, the Siebel administrator sets a system preference to reflect which rule is to be in effect.

**NOTE:** Mobile Web Client transactions are processed in the order in which the users synchronized. Any successful database update sent from a Mobile Web Client to the server becomes a server transaction for the purpose of conflict resolution.

If Siebel Remote rejects an update from a Mobile Web Client, the mobile user can determine whether an update conflict has occurred. If the result of Siebel's conflict resolution is inappropriate, the user can manually reapply the change to the local database. It will be sent to the server again during the next synchronization session when the Mobile Web Client sends database changes to the server. If other changes have not been made to the data value since the last synchronization session, the change will not conflict and will succeed on the server.

## Insert Conflicts

Although a user may successfully add a database row to a client database (or the server database), the added (or inserted) transaction may duplicate a new entry in another database residing elsewhere and not yet processed by the server. Siebel Remote determines that an insert or duplicate conflict exists when a new row's user primary key matches that of an existing row.

Because Siebel Remote cannot determine whether the transaction is a true duplicate or simply an erroneous use of the same identifier for two different data entities, Siebel Remote cannot ignore the duplicate transaction. Instead, Siebel Remote adds the new row and sets the column CONFLICT_ID, which is normally Zero, to the record's ROW_ID. This both indicates that the row is a duplicate and assures a unique value for the _U1 index. The mobile user can determine whether an insert conflict (also sometimes called a duplicate conflict) has occurred by consulting the Remote Status view and by watching for duplicates in regular data views, such as Accounts or Contacts.

For example, the user might change the user primary key and resubmit the update. As the user resolves the conflict, Siebel Remote captures the database update for transmission to, and resynchronization with, the server during the next synchronization session.

**NOTE:** The local database treats null as a unique value and consequently allows duplicates if you leave a key field null for two or more records.

The system administrator must monitor and resolve conflicts created on the server. The conflicts are visible as duplicate records in regular data views such as Accounts or Contacts. To resolve an insert conflict, you can perform a merge on the duplicate records using the user interface Merge Record feature (Edit > Merge Records). The Merge Records option is enabled only after you select more than two records in the applet.

You can also change the user keys of one of the duplicate records to resolve the conflict.

Until these conflicts are resolved, EIM cannot be used to merge the records, since the conflict flag is not reflected in the interface table columns.

## Delete Conflicts

A potential delete conflict arises whenever one of the database changes is to delete a row. A single rule applies to delete operations and overrides other transactions. Whenever Siebel Remote encounters a delete transaction, *delete always wins* whether or not the transaction is in conflict with another update. If one transaction updates a database row and another deletes the row, Siebel Remote ignores the update and deletes the row.

**NOTE:** This rule supersedes the System Preferences Conflict Resolution rule set during initial implementation.

When detecting delete conflicts, deleted records appear in the User Preferences screen > Remote Status view > Session Actions list after the user synchronizes. However, the deleted records appear only if the system preference MRG: User Friendly Notification is set to TRUE. See "Setting Siebel Remote System Preferences" on page 48 for more information.

## Merge Conflicts

A potential merge conflict arises when records are merged separately on both the Mobile Web Client and the database server. The following example illustrates the problem:

1. On the Mobile Web Client, data from account A may be merged with account B.
2. On the server database, data from account B may be merged with account A.
3. Since delete transactions have the highest priority in Siebel Remote, this may lead to two delete transactions—one from the Mobile Web Client and one from the database server.
4. In this case, data from both accounts will be deleted. To avoid this potential problem, do not merge data on Mobile Web Clients.

# 3 Implementing Siebel Remote Server

This chapter describes actions that are necessary or strongly recommended to prepare the Remote server for use in the production environment. Table 4 includes these tasks.

Table 4. Tasks for Initial Siebel Remote Implementation

| Task | Where Performed |
|---|---|
| About Virtual Memory or Swap Size for a Siebel Remote Server on page 47 | Siebel Remote server |
| Setting Siebel Remote System Preferences on page 48 | Siebel application connected to server |
| Choosing Assignment Manager Settings for Siebel Remote on page 54 | Siebel Server running Assignment Manager |
| Disabling Local Access to All Views on page 55 | Siebel application connected to server |
| Preventing Extraction and Synchronization of Older Data on page 55 | Siebel application connected to server |
| Starting Siebel Remote Server Components on page 58 | Siebel Server Manager |
| Changing the Local Database Administrator Password on page 66 | Siebel Remote server |
| Generating a New Database Template on page 67 | Siebel Server Manager |

## About Virtual Memory or Swap Size for a Siebel Remote Server

Performance can be affected by the size you set for virtual memory (Windows) or swap space (UNIX) on your Siebel Remote server. For best results, it is recommended that you set the size of virtual memory or swap space to 1.5 times the size of the server's physical memory (RAM). Use standard operating system techniques to accomplish this. For example,

- In Windows 2000, the virtual memory setting is reached by navigating to My Computer > Properties > Advanced > Performance Options > Virtual Memory > Change.
- In UNIX, a typical command to set swap space size is *swap*. Consult the documentation for your server's operating system for confirmation that this is the correct command and for command syntax details.

# Setting Siebel Remote System Preferences

Several enterprise-wide preferences affect the way Siebel Remote manages database changes. During the initial implementation, you should use the System Preferences Administration view to set the system preferences.

**NOTE:** It is important to remember that Mobile Web Clients need to be reextracted whenever there is a change to any of the system preferences settings. To determine which changes will be routed to the Mobile Web Clients, check the Dock Object Visibility Rules under the SystemPref Dock Object in Siebel Tools. For more information on Dock Objects and Visibility Rules, see *Configuring Siebel eBusiness Applications*.

### *To set Siebel Remote system preferences*

1. From the application-level menu, choose Navigate > Site Map > Administration - Application > System Preferences.
2. In the System Preferences list, select the desired record and enter the appropriate value in the System Preference Value field.

Table 5 lists the Siebel Remote system preferences that you may want to set. For more information about each system preference, see the related section in the following pages.

Table 5. Siebel Remote System Preferences

| System Preference | Default Value | Other Values |
|---|---|---|
| CSM Logging | FALSE | TRUE |
| DBX: Vis Rules Per Statement 1 | 20 for DB2<br>20 for Microsoft SQL Server<br>20 for Oracle | |
| DBX: Vis Rules Per Statement N | 1 for DB2<br>1 for Microsoft SQL Server<br>1 for Oracle | |
| Docking: Transaction Logging | TRUE | FALSE |

Table 5. Siebel Remote System Preferences

| System Preference | Default Value | Other Values |
|---|---|---|
| Enterprise DB Server Code Page | Value is set by the database installer. | ■ utf-8<br>■ utf-16 (includes MS SQL and DB2 only)<br>■ cp1252<br>■ cp932<br>The following are for development and migration only:<br>■ cp847<br>■ cp936<br>■ cp949<br>■ cp950<br>■ cp1250<br>■ cp1251<br>■ cp1253<br>■ cp1254<br>■ cp1255<br>■ cp1256<br>■ cp1257<br>■ cp1258 |
| LOGMGR: Vis Rules Per Statement | 50 | 20 for DB2<br>20 for Microsoft SQL Server<br>20 for Oracle |
| MRG: Docking Timestamp Source | Client Transaction Time | Server Database Merge Time |
| MRG:Inter Table Conflict Res | Merge | First In<br>Last In |
| MRG:Inter Table Merge Rule | First In | Last In |
| MRG: System Conflict Resolution | Server Wins | Client Wins |

Table 5. Siebel Remote System Preferences

| System Preference | Default Value | Other Values |
|---|---|---|
| MRG: Txns Per Commit | 50 | 10 for DB2<br>10 for Oracle<br>1 for Microsoft SQL Server |
| MRG: User Friendly Notification | Conflicts | TRUE<br>FALSE |

## CSM Logging

Turns on logging of merge transactions. These are transactions created when connected users combine or merge two records of the same component, such as two opportunities into one. Information from this log can be used to relink orphaned records created when the merge process is crossed with an update process during a synchronization. This is only applicable if the installation uses Siebel Remote or Replication Manager.

## DBX: Vis Rules Per Statement 1

Sets the number of visibility or routing rules checked per SQL statement processed by the Database Extract component—for the first SQL statement with header information. For default values see Table 5 on page 48.

**NOTE:** Do not change this value unless instructed by Siebel Technical Services.

## DBX: Vis Rules Per Statement N

Sets the number of visibility or routing rules checked per SQL statement processed by the Database Extract component for other SQL statements. For default values see Table 5 on page 48.

**NOTE:** Do not change this value unless instructed by Siebel Technical Services.

## Docking: Transaction Logging

This preference enables or disables docking transaction logging. The default value is TRUE. This value is case sensitive.

You must set this value to TRUE when performing a database extract. Transaction logging must be enabled during extraction to log changes to the local database. The following procedures provide detailed instructions for disabling and enabling transaction logging.

**CAUTION:** Disabling Transaction Logging will prevent Siebel Remote from working.

### *To disable transaction logging*

1. From the application-level menu, choose Navigate > Site Map > Administration - Application > System Preferences.
2. In the System Preferences list, select the Docking: Transaction Logging record and set System Preference Value to False.
3. Restart the Siebel server.

### *To enable transaction logging*

1. From the application-level menu, choose Navigate > Site Map > Administration - Application > System Preferences.
2. In the System Preferences list, select the Docking: Transaction Logging record and set System Preference Value to TRUE.

   The following steps must also be completed whenever the value of Docking: Transaction Logging is changed from FALSE to TRUE:

3. Restart the Transaction Processor with TS DB Recreate set to True.

   This restores transaction logging and rebuilds the dobjinst.dbf file to make sure that correct, up-to-date information about the data in the database is cached. For information about starting the Transaction Processor, see "Starting Siebel Remote Server Components" on page 58.

   **CAUTION:** If the dobjinst.dbf file is not rebuilt when Transaction Logging is enabled, Mobile Web Clients may experience data discrepancies or visibility problems.

4. For each active Mobile Web Client in your system (that is, any Mobile Web Client who has no End Date set), reextract and reinitialize that client's local database to make sure that it contains up-to-date data.

   This is necessary because any transactions that occur while Transaction Logging is set to False are not logged, and thus are not routed to the Mobile Web Clients that would otherwise receive them, even after Transaction Logging is set to True again.

When using the Assignment Manager or EIM and Docking: Transaction Logging is set to TRUE (default), all the changes will be logged and the appropriate ones will be propagated to the Remote clients. The changes are all logged into the S_DOCK_TXN_LOG table.

With a large volume of data, it may take quite a long time for the Transaction Processor and Router tasks to process the changes for each of the Remote clients. It may take so long that it would be faster to reextract the Mobile Web Client and apply the extract remotely.

Therefore, it is recommended that you turn off transaction logging when loading data using EIM to avoid the rapid expansion of the S_DOCK_TXN_LOG table. After, the data has been loaded and assigned, turn on transaction logging and reextract the mobile clients. This will also prevent the risk of overloading of the Transaction Processor and Router processes.

In general, bigger batches tend to enhance performance for EIM, but cause problems for Siebel Remote. If you are using both EIM and Siebel Remote, it is recommended that you limit batch sizes to 1000 or fewer records.

## Enterprise DB Server Code Page

Beginning with Siebel 7.5, non-Unicode code pages are supported for the server database, while the local database only supports Unicode. This parameter is used by the Mobile Web Client to prevent mobile users from entering noncompatible-character-set data into the local database. The value of this parameter should be set to match the character set of the server database. Values for this parameter include: utf-8, utf-16, cp1252, and cp932. Utf-8 and utf-16 are for Unicode, cp1252 for English and most western European languages, and cp932 for Japanese. For default values, see Table 5 on page 48.

## LOGMGR: Vis Rules Per Statement

Sets the number of visibility or routing rules to be checked by the Transaction Router server component per SQL statement. For default values see Table 5 on page 48.

**NOTE:** Do not change this value unless instructed by Siebel Technical Services.

## MRG: Docking Timestamp Source

Siebel database table rows include the date a row was last changed. For rows updated by Mobile Web Clients, the date can be stamped with the time when the change was made on the client (client timestamp) or the time when the update was applied to the server (server timestamp):

- Using the client timestamp promotes consistency across client and server databases, but can be misleading if a client's clock is set incorrectly or if the client resides in a different time zone.
- Using the server timestamp makes sure that timestamps are accurate, but causes the timestamps to differ for the same row on the client and the server.
- Setting this preference does not affect Siebel Remote conflict resolution logic or priorities. Updated timestamps are stored for purely informational purposes.
- To select Docking Timestamp Source, choose either Client Transaction Time (the default) or Server Database Merge Time.

## MRG:Inter Table Conflict Res

In releases prior to Siebel Release 7.7, insert conflicts on intersection tables were handled by creating a duplicate record with a non-zero value in the Conflict ID field. Beginning with Release 7.7, you can choose between available values for this preference to determine how your Siebel Implementation will handle insert conflicts on intersection tables in Enterprise, Regional, and local Siebel databases.

When a record that is involved in an insert conflict is associated with extension records and child records, the extension records and child records undergo the same conflict resolution process as the parent record.

The following paragraphs describe the available values for the MRG:Inter Table Conflict Res system preference:

- **First In.** Keeps the existing record, along with its associated extension records and child records. Discards all values from the record for which an insertion was attempted, along with that record's associated extension records and child records.
- **Last In.** Replaces the existing record with the record being inserted. Replaces the associated extension records and child records of the existing record with those of the record being inserted.
- **Merge.** (Default value) Combines values from both records, along with their extension records and child records. Exact behavior depends on which fields are populated with non-NULL values and on the value of the system preference MRG:Inter Table Merge Rule. For more information, see "MRG:Inter Table Merge Rule" on page 53.

**CAUTION:** This system preference should be set before any Mobile Web Clients are extracted. If the preference is changed after extraction, Mobile Web Clients should be re-extracted, otherwise server data and client data may diverge.

Mobile Web Client users can view the results of conflict resolution operations in their local databases by navigating to User Preferences > Remote Status.

## MRG:Inter Table Merge Rule

If the system preference MRG:Inter Table Conflict Res has a value of Merge, then the system preference MRG:Inter Table Merge Rule determines which field values survive when two records are merged to resolve an insert conflict on an intersection table. Available values for this preference are as follows:

- **First In.** (Default value) If MRG:Inter Table Merge Rule value is First In, the following applies to fields in parent records, child records, and associated extension records:
  - New non-NULL values always replace existing NULL values.
  - Existing non-NULL values always are preserved.
- **Last In.** If MRG:Inter Table Merge Rule value is Last In, the following applies to fields in parent records, child records, and associated extension records:
  - New non-NULL values always replace existing NULL values.
  - New non-NULL values replace existing non-NULL values.
  - Existing non-NULL values are preserved if the corresponding new value is NULL.

If the System Preference called MRG:Inter Table Conflict Res has a value *other than* Merge, then the System Preference called MRG:Inter Table Merge Rule is ignored.

**CAUTION:** This system preference should be set before any Mobile Web Clients are extracted. If the preference is changed after extraction, Mobile Web Clients should be re-extracted, otherwise server data and client data may diverge.

For more information about resolving insert conflicts, see "MRG:Inter Table Conflict Res" on page 52.

## MRG: System Conflict Resolution

Siebel Remote uses one of two rules to resolve conflicting database updates:

- **Client Wins.** Updates from a Mobile Web Client will take precedence and overwrite those already on the server.
- **Server Wins.** Updates from a server will take precedence and overwrite those already on the Mobile Web Client. This is the default value.

These values are case sensitive. You should use the default setting.

## MRG: Txns Per Commit

The value for this preference determines how many database transactions the Transaction Merger processes before it issues a database commit:

- Setting a low value (1) reduces the frequency with which users encounter a locked database row and reduces the risk of deadlock.
- Setting a high value (10) minimizes processing overhead.
- For default values see Table 5 on page 48.

## MRG: User Friendly Notification

This value determines whether Siebel Remote writes information about database updates to the Siebel Remote Status view on the Mobile Web Client:

- The values are case sensitive.
- If the value is TRUE, Siebel Remote writes information about database updates.
- If the value is CONFLICTS, Siebel Remote writes information about database updates that caused conflicts.
- If the value is FALSE, Siebel Remote does not write any information about database updates.

# Choosing Assignment Manager Settings for Siebel Remote

Assignment Manager settings can affect which transactions are sent to a Siebel Remote client. In normal operation, Assignment Manager frequently updates the timestamps for large numbers of records, even if there are no other changes to many of those records. To avoid sending large numbers of transactions to a Siebel Remote client when the only change is to the timestamp, it is recommended that you set the LogTxnChgOnly parameter for the Assignment Manager component to TRUE at the component level.

When the LogTxnChgOnly parameter is set to TRUE, Assignment Manager transactions are logged only when there is a net change in assignments, such as a change in the membership of a team or a change in the primary for a team. This parameter only affects the Assignment Manager transactions, so record changes made by other means are logged and sent to Siebel Remote clients normally.

For information about configuring Assignment Manager component parameters, see *Siebel Assignment Manager Administration Guide*.

# Disabling Local Access to All Views

If the tables associated with the business objects have limited visibility, you should not allow mobile users to use any of the All views when connected to a Siebel Remote local database. The Siebel client attempts to fix foreign key relationships when displaying data. Siebel Remote sends this change to the server database and other Mobile Web Clients. This will corrupt the integrity of these databases.

An example using one of the All views illustrates this behavior, which is inherent in any of the All views.

The All Opportunity List view resets the value of an opportunity's primary account to NULL if the Siebel client cannot locate the account in the local database. Siebel Remote may replicate an opportunity to a local database because the opportunity is referenced by an activity owned by the mobile user. Siebel Remote replicates the opportunity so that the Siebel client can display the opportunity's name with the activity in the Activity List view. However, Siebel Remote does not replicate the opportunity's primary account if the mobile user is not a member of the opportunity's sales team. Thus, if the mobile user employs the All Opportunity List view to display the opportunity, and the Siebel client does not find the opportunity's primary account in the local database, Siebel Remote resets the opportunity's primary account value to NULL and replicates this change to other databases.

### *To disable local access to the All Opportunity List view*

1. From the application-level menu, choose Navigate > Site Map > Administration - Application > Views.
2. In the Views list, select the appropriate view such as the All Opportunity List view and clear the Local Access check box.

Because this is an example that applies to All views, repeat this procedure for all other views that are assigned to the responsibilities of mobile users.

# Preventing Extraction and Synchronization of Older Data

The Time Filters feature allows administrators to prevent selected kinds of older data from being sent to all Mobile Web Clients during database extraction or synchronization. By reducing the amount of data to be sent, time filtering can reduce the amount of time required for database extraction and synchronization. Time filtering may also enhance Mobile Web Client response time.

Time filtering is deployed on a per dock object basis. For each supported dock object that will use time filtering, the administrator chooses a cut-off date. Data that was last modified prior to the cut-off date is not included in extractions or synchronizations.

**NOTE:** The cut-off date specified for each dock object is a fixed date that is not dynamically adjusted. If the cut-off date is changed, Mobile Web Clients should be reextracted. If a Mobile Web Client is not reextracted after changing to a later cut-off date, that client would retain unnecessary older data in the local database. Storage of this unneeded data could affect performance.

**CAUTION:** Before deploying time filtering in your Production environment, be sure to test it thoroughly. Make sure that your chosen cut-off dates for time filtering allow all necessary data to reach test Mobile Web Clients. Deployment of an inappropriate cut-off date can prevent stable but necessary data (such as price lists or rate lists) from reaching Mobile Web Clients. If this occurs, you must choose an earlier cut-off date and reextract the Mobile Web Clients.

By default, time filtering is supported for the following dock objects:

- Activity
- Expense
- Expense Report
- Invoice
- Opportunity
- Price List
- Project
- Project Item
- Quote
- Service Request

It is possible to configure additional dock objects for time filtering. However, for many Siebel implementations, most of the benefits that time filtering provides will come from using the dock objects that are supported by default. For information about making time filtering available for additional dock objects, see "Configuring New Dock Objects for Time Filtering" on page 57.

The following procedure describes how to set up time filtering for a supported dock object.

### *To prevent synchronization of older data for a supported dock object*

1. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote > Time Filters.
2. In the Time Filters list, click New to display a blank record.
3. In the Dock Object field, select the dock object for which you want to set up time filtering.

4 In the Cutoff Time field, specify a date and time to limit the data to be extracted and synchronized for the selected dock object.

**NOTE:** Time Filters records can be created while the Transaction Router is running. The time filtering you specify will affect any database extracts that are started after the record is saved. However, to affect synchronizations, you must also restart the Transaction Router component after saving the record.

If you use time filtering but find, by experience, that your business still requires synchronization of older data, the following procedure describes how you can stop using time filtering for any supported dock object.

### *To allow synchronization of older data for a supported dock object*

1 From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote > Time Filters.

2 In the Time Filters list, select the record that lists the dock object for which you need older data to be synchronized.

3 Click the Menu button and select Delete Record.

Repeat Step 2 and Step 3 for any other dock objects that no longer need time filtering.

4 Restart the Transaction Router component.

For information on starting Siebel Remote components, see "Starting Siebel Remote Server Components" on page 58.

5 Re-extract local databases for any Mobile Web Clients that need access to the older data.

For information on extracting local databases, see Chapter 5, "Extracting Databases for Mobile Web Clients."

**NOTE:** Time Filters records can be deleted while the Transaction Router is running. The deletion of a time filtering record will affect any database extracts that are started after the record is deleted. However, to affect synchronizations, you must also restart the Transaction Router component after deleting the record.

## Configuring New Dock Objects for Time Filtering

Beginning with Siebel release 7.7, the time filtering feature allows you to prevent synchronization of older data for specified dock objects. For information about the dock objects for which time filtering is supported by default, see "Preventing Extraction and Synchronization of Older Data" on page 55.

It is possible to configure additional dock objects to use time filtering. However, for many Siebel implementations, most of the benefits that time filtering provides will come from using the dock objects that are supported by default. The following procedure describes how to make time filtering available for a non-default dock object.

**NOTE:** The following procedure is a particular example of adding a new value to a List of Values (LOV). For general information about working with LOVs, see *Applications Administration Guide*.

***To make time filtering available for a non-default dock object***

1. In Siebel Tools, navigate to the Dock Object Type, find the dock object for which you want to use time filtering, and make a note of its exact name.
2. Log in to the Siebel application using an administrator user ID, and then, from the application-level menu, choose Navigate > Site Map > Administration - Data > List of Values.
3. In the List of Values list, use standard query techniques to display existing records that have the value DOCK_OBJ_TIME_FILTER in the Type field.
4. Select any one of these records, click the Menu button, and select Copy Record.
5. In the Language-Independent Code field of the new record, enter the exact name of the dock object for which you want to use time filtering.
6. In the Display Value field of the new record, enter the name you would like your Siebel applications to display when you set up time filtering for this dock object.
7. Inspect the copied values in the other fields of the new record, make any necessary changes, and save the record.

   Time filtering is now available for the specified dock object. For information about setting up time filtering for this dock object, see "Preventing Extraction and Synchronization of Older Data" on page 55.

# Starting Siebel Remote Server Components

Use Siebel Server Manager to set values for start-up parameters for the following Siebel Remote server components:

- Transaction Processor
- Transaction Router
- Transaction Merger
- Synchronization Manager

The default parameter values for each component are described in the following sections.

The following procedures describe how to configure these server components.

***To start Transaction Processor – srvrmgr command line***

- From the srvrmgr command line, enter the following command all on one line:

  ```
  start task for comp txnproc server <server_name> with <parameter1>= <value1>,
  <parameter2=<value2>,...
  ```

  Values are from "Start-Up Parameters for Transaction Processor" on page 60.

  **NOTE:** When logging into srvrmgr command line, indicate the server name. Otherwise srvrmgr will default to the Enterprise Server. For details regarding the Server Manager command-line interface, see *Siebel System Administration Guide*.

### *To start Transaction Processor – GUI*

- In Siebel 7, the Transaction Processor cannot be started manually through the GUI Server Manager unless the Siebel Server is up and running. The component has been defined to start one task when the Siebel Server is started if the component is enabled. If the component is not started, from the application-level menu, choose Navigate > Site Map > Administration - Server Management > Servers > Component Group, select the Siebel Remote component group, select the Transaction Processor component, and click Startup. A new task should start automatically.

### *To start Transaction Router – srvrmgr command line*

- From the srvrmgr command line, enter the following command all on one line:

  ```
  start task for comp txnroute server <server_name> with
  <parameter>=<value>,<parameter2=<value2>,...
  ```

  Values are from "Start-Up Parameters for the Transaction Router" on page 61.

### *To start Transaction Router – GUI*

- In Siebel 7, the Transaction Router cannot be started manually through the GUI Server Manager unless the Siebel Server is up and running. The component has been defined to start at least one task when the Siebel Server is started if the component is enabled. If the component is not started, from the application-level menu, choose Navigate > Site Map > Administration - Server Management > Servers > Component Group, select the Siebel Remote component group, select the Transaction Router component, and click Startup. A new task should start automatically.

### *To start Transaction Merger – srvrmgr command line*

- From the srvrmgr command line, enter the following command all on one line:

  ```
  start task for comp txnmerge server <server_name> with
  <parameter>=<value>,<parameter2>=<value2>,...
  ```

  Values are from "Start-Up Parameters for Transaction Merger" on page 63.

### *To configure Transaction Merger – GUI*

- In Siebel 7, the Transaction Merger cannot be started manually through the GUI Server Manager unless the Siebel Server is up and running. The component has been defined to start at least one task when the Siebel Server is started if the component is enabled. If the component is not started, from the application-level menu, choose Navigate > Site Map > Administration - Server Management > Servers > Component Group, select the Siebel Remote component group, select the Transaction Merger component, and click Startup. A new task should start automatically.

### *To configure Synchronization Manager - srvrmgr command line*

- Synchronization Manager is started automatically by the Siebel Server using the default configuration. It does not need an explicit configuration. For information on using Siebel Server Manager to manage and administer server components, see *Siebel System Administration Guide*.

# Transaction Processor

Table 6 shows the start-up parameters for the Transaction Processor.

Table 6. Start-Up Parameters for Transaction Processor

| Name | Alias | Required /Optional | Comments and Default Value |
|---|---|---|---|
| Sleep Time | SleepTime | Optional | Time (in seconds) to sleep between iterations. When it wakes up, it will process transactions. The default is 60. |
| Maximum Reads per Iteration | MaxRead | Optional | Maximum number of operations to read per run. The default value is 0, which means it will read all outstanding operations in the Master Transaction Log. |
| Clean txns iterations | CleanTxnsIter | Optional | Sets frequency for Txn Processor to delete txns from server database master txn log table. This parameter is specified in the number of iterations. Default is 10. |
| Clean .dx files iterations | CleanFilesIter | Optional | Sets frequency for Txn Processor to delete DX files from the Siebel server's DOCKING\TXNPROC directory. The default is 1 iteration, default of 60 seconds per iteration as shown in SleepTime parameter. |
| Write compressed .dx files | WriteCompressed | Optional | Writes DX files in compressed format. Default is FALSE and should *not* be changed unless advised by Siebel Technical Support or Siebel Expert Services. |
| TS Block Size | TSBlockSize | Optional | Block size for dobjinst.dbf (visibility database) in bytes. Data is stored blocks. The block size here is different from Operating System block size. It is determined during dbxtract for a Mobile Web Client. The default is 4096 bytes. Please do not modify this parameter without approval from Expert Services or Technical Support. |

Table 6. Start-Up Parameters for Transaction Processor

| Name | Alias | Required /Optional | Comments and Default Value |
|---|---|---|---|
| TS Cache Size | TSCacheSize | Optional | Cache size for dobjinst.dbf (visibility database) in kilobytes. The default is 4096 kilobytes.<br>Please do not modify this parameter without approval from Expert Services or Technical Support. |
| TS DB Recreate | TSDbRecreate | Optional | The default value is FALSE. If the parameter is set to TRUE, Txnproc will recreate its dobjinst.dbf (visibility database). It is necessary to do this because an older dobjinst.dbf database (file system recovery) will mismatch with the server database. The new one resolves the mismatch. |

## Transaction Router

The Transaction Router calculates visibility for transactions and routes data to mobile users. Table 7 shows selected parameters for the Transaction Router.

Table 7. Start-Up Parameters for the Transaction Router

| Name | Alias | Required /Optional | Default Value and Usage Notes |
|---|---|---|---|
| Sleep Time | SleepTime | Optional | Time to sleep between iterations (in seconds). The default is 60. |
| Maximum writes per file | MaxWrite | Optional | Maximum number of operations written per DX file. The default is 5000. |
| Maximum reads per iteration | MaxRead | Optional | Maximum number of operations processed by the Transaction Router for a given mobile user during each run. The default is 10000. |

Table 7. Start-Up Parameters for the Transaction Router

| Name | Alias | Required /Optional | Default Value and Usage Notes |
|---|---|---|---|
| Set Application Server Name | SetAppName | Optional | Upgrades Mobile Web Clients from previous versions of Siebel eBusiness Applications to v4.0 and above. The default is TRUE.<br><br>At start-up, the transaction router searches the server database for Mobile Web Clients whose Siebel Remote server value is not set and that have a docking directory on the local server. This parameter sets the application server name in the server database for each of these Mobile Web Clients. |
| Write compressed .dx files | WriteCompressed | Optional | Write DX files in compressed format. The default is TRUE. |
| TS Cache Size | TSCacheSize | Optional | Cache size for dobjinst.dbf database in kilobytes. The default is 2048 kilobytes. Maximum value should not exceed the number of mobile users on the application server multiplied by the size of the largest dobjinst.dbf file. |
| Read client list iterations | ReadClientsIter | Optional | The number of runs before the Transaction Router refreshes the list of users it should be processing from the database.<br><br>The default is 10 iterations. |
| Id Db Recreate | IdDbRecreate | Optional | Recreates the visibility ID database, visdata.dbf. It is a cache storage for data required for the visibility check. The default is FALSE, which means it does not recreate the database every time Txnroute starts. |
| Id Db Size | IdDbSize | Optional | Size of the visibility ID database, visdata.dbf. This parameter is used only when no visdata.dbf file is present, or when Id Db Recreate is set to True. The default is 10MB, which is optimal in most cases. Maximum value should not exceed 200 MB. |

Table 7. Start-Up Parameters for the Transaction Router

| Name | Alias | Required /Optional | Default Value and Usage Notes |
|---|---|---|---|
| Node Division Factor | NodeDivFactor | Optional | Determines the maximum number of mobile users to process by a Transaction Router instance during every run. The component will process the fractional number of users computed using this parameter. For example, if the value of this parameter is 5 (the default) and there are 100 mobile users on the Application server, the component will process 100/5 or 20 users. |
| Maximum seconds per iteration | MaxSecs | Optional | Determines the longest duration that a Transaction Router instance will work on one mobile user during each run. The default is 300 seconds. |

## Transaction Merger

Table 8 shows selected parameters for the Transaction Merger.

Table 8. Start-Up Parameters for Transaction Merger

| Name | Alias | Required /Optional | Default Value and Usage Notes |
|---|---|---|---|
| Error Mode | ErrorMode | Optional | Determines how database errors are handled during a Transaction Merger process. Default is STOP. Available modes include:<br>■ STOP. Stops Transaction Merger.<br>■ IGNORE. Causes Transaction Merger to sleep if all nodes were ignored in an iteration.<br>■ DISABLE_NODE. Disables all nodes and stops Transaction Merger if an error recurs in the same row. If the mode is set to DISABLE_NODE, Transaction Merger sets the effective end dates of clients with errors to the current time. You then need to reextract the Mobile Web Clients to reactivate them. |
| Sleep Time | SleepTime | Optional | Time to sleep between iterations (in seconds). Default is 60. |

# Synchronization Manager

Table 9 shows the start-up parameters for the Synchronization Manager.

Table 9. Selected Synchronization Manager Start-up Parameters

| Name | Alias | Required/ Optional | Default Value and Usage Notes |
|---|---|---|---|
| Authentication Method | Authentication | Optional | Method that Siebel Remote uses to authenticate Mobile Web Clients. The default is None.<br>■ **None.** Does not validate the synchronization authentication credentials. This is the default.<br>■ **Database.** Validates the synchronization authentication credentials against the server database user ID and password.<br>■ **Siebel.** Validates the synchronization authentication credentials against the Mobile Client name and Sync Password that is stored in the server database and is maintained for each Mobile Web Client in the Administration - Siebel Remote > Mobile Clients view under the appropriate parent server.<br>■ **AppServer.** Validates the synchronization authentication credentials against the user ID and its password on the Siebel Server operating system.<br>■ **SecurityAdapter.** Validates the synchronization authentication credentials against a third-party authentication system. |
| Domain Name | NTDomain | Optional | If you are using AppServer authentication, this parameter specifies the name of the Windows domain for the user name and password.<br>To use Windows password authentication, you must enable the Windows user rights for the Siebel Server. |
| Minimum Number of cached thread contexts | MinCtxCache | Optional | Minimum number of cached thread contexts maintained by a multithreaded server. The default is 2 cached contexts. |

Table 9. Selected Synchronization Manager Start-up Parameters

| Name | Alias | Required/ Optional | Default Value and Usage Notes |
|---|---|---|---|
| Maximum Number of cached thread contexts | MaxCtxCache | Optional | Maximum number of cached thread contexts maintained by a multithreaded server. The default is 10 cached contexts. |
| Maximum Task Number | MaxTasks | Optional | Maximum number of synchronization sessions Synchronization Manager can service simultaneously. Default is 100. |
| Static Port Number | PortNumber | Optional | The TCP/IP port number dedicated to the Synchronization Manager. Default value is 40400. Use the Administration – Server Configuration > Servers view to override the default value with a new value, and then restart the component.<br>You can also specify this port number as a command-line option when starting this server component via Server Manager using the command-line interface. |
| Siebel File System | FileSystem | Optional | Siebel File System location for the Enterprise. |
| Encryption Type | Crypt | Optional | Determines if the traffic of a synchronization session will be encrypted. Values are: RSA, MSCRYPTO, and NONE. |

The multithreaded Synchronization Manager maintains a collection of open database connections (context cache) that can be parceled out to the active task threads and be reused. By default, each server creates two connections during start-up. Additional contexts may be created dynamically, but the server (by default) maintains a maximum of only 10 context caches.

You can specify the minimum cache size using the MinCtxCache parameter and the maximum cache size using the MaxCtxCache parameter. A larger cache may be helpful for configurations where multiple Mobile Web Clients synchronize. Note, however, that an excessive number of inactive database connections can degrade system performance.

The Synchronization Manager creates the number of cached contexts specified by the MinCtxCache parameter at start-up. A cached context decreases the time to initialize a new synchronization session. Increase the cache size if you have a large number of Mobile Web Clients to synchronize simultaneously.

The Synchronization Manager can service up to the MaxTasks number of synchronization sessions simultaneously, but it only keeps at most the MaxCtxCache number of cached contexts in memory.

# Changing the Local Database Administrator Password

A Siebel Remote server administrator can change the local DBA password on Mobile Web Clients. The password is set when the administrator generates the database template that will be used to create the local databases for those clients.

To set the local DBA password, the administrator uses the New DBA password (NewDbaPwd) parameter of the Generate New Database server component. There are two ways to specify a value for this parameter:

- If the administrator wants to use a particular local DBA password for *one-time use* of the Generate New Database component, the parameter can be specified when the component is run.

  **NOTE:** A NewDbaPwd parameter value that is specified at runtime overrides the default parameter value specified by Siebel Systems and any value that is set in an administration view, but *this value is used only for only the current run of the Generate New Database component.*

- If the administrator wants to use a particular local DBA password for *multiple* uses of the Generate New Database component, the parameter value can be specified and saved in an administration view, as described later in this topic. If the parameter value is specified in an administration view, it does *not* need to be specified when the component is run.

If the administrator does not specify a value for this parameter by *either* of these methods, Generate New Database sets the local DBA password to the first 8 characters of the enterprise name if the name has 8 or more characters, or to the enterprise name concatenated with nnn, where nnn = 1234… as padding if the enterprise name has fewer than 8 characters. For example, by default, the local DBA password is set to ONECUSTO for an enterprise named OneCustomer or to SIEBEL12 for an enterprise named Siebel.

The following procedure describes how to set a local DBA password for use on multiple occasions:

### *To set a local DBA password for multiple uses of GenNewDb*

1. From the application-level menu, choose Navigate > Site Map > Administration - Server Configuration > Enterprises > Component Definition.
2. In the Component Definition list, use standard query techniques to select the Generate New Database component.
3. In the Component Parameters form, use standard query techniques to select the New DBA password parameter.
4. In the Value field, enter the local DBA password you want to use on multiple occasions, and then save the record.

For general information about generating database templates, see "Generating a New Database Template" on page 67. For a specific information about specifying parameters when the component is run, see "Running the Generate New Database Component" on page 68.

# Generating a New Database Template

A database template is a database which contains no end user data but which has the specific structure required for use with Siebel applications. You must run the Generate New Database Template component to create a new database template when you:

- Upgrade the server database.
- Want to use a collating sequence other than the default.
- Use Database Extensibility to add extension tables, extension columns, or extension indexes to the default Siebel schema.

**NOTE:** When you create a new extension column in the Siebel schema, and define the Datatype as Character (CHAR), there may be padding issues with Remote. Specifically, TxnMerge may not work properly if the Datatype is CHAR with a length of more than 1. Use VARCHAR as the Datatype if its length is more than 1.

The following topics provide additional information about working with new database templates:

- "Creating a New Empty Database File" on page 67
- "Running the Generate New Database Component" on page 68
- "Distributing Database Templates to Siebel Remote Servers" on page 70

## Creating a New Empty Database File

Two local database template files are provided with your Siebel application:

- sse_utf8.dbf is an unencrypted template.
- sse_encr.dbf is a template that is encrypted with standard Sybase encryption.

Both templates are created with Unicode encoding. Every collating sequence can be supported. The sorting sequence for the application is configured using the SortCollation parameter in the CFG file. If the parameter is not specified, the application will sort using the Unicode multilingual collation order.

**NOTE:** Some previous versions of Siebel Remote supported the creation of new empty database files through the use of a utility called dbinit, which was designed for use with SQL Anywhere databases. Siebel Systems no longer supports the use of this dbinit utility.

### Encrypting the Local Database

By default, the template that defines the local database schema is unencrypted. You can use an encrypted local database template to provide a layer of security against unauthorized access to the local database.

You must run the server component Generate New Database whenever you change the schema of the Siebel database. When you run Generate New Database, you can opt to use a template for the local database that uses standard Sybase encryption.

To use the encrypted local database template, the parameters that specify the SQL Anywhere database template file for the Generate New Database and Database Extract server components must have the same value.

- **Generate New Database.** The SQL Anywhere Database parameter must have the value sse_encr.dbf.
- **Database Extract.** For subsequent database extracts that employ the encrypted template, the Database template file name parameter must have the value sse_encr.dbf.

For information about generating a new database template, see "Running the Generate New Database Component" on page 68.

For information about doing a database extract, see "Extracting Databases for Mobile Web Clients" on page 103.

For a list of all Database Extract component parameters, see "Database Extract Parameters" on page 109.

## Running the Generate New Database Component

The following procedure describes how to generate a new database template by running the Generate New Database component.

### *To generate a new database template (GenNewDb)*

1. From the application-level menu, choose Navigate > Site Map > Administration - Server Management > Jobs.
2. In the Jobs list, click New.
3. In the Component/Job field, select Generate New Database.
4. In the Requested Server field, enter the name of the server on which you want the GenNewDb job to run.

   After the job is completed, the Execution Server field displays the name of the server that actually ran the job.
5. In the Job Parameters list, click New.
6. In the Name field, click the select button and use standard query techniques to select SQL Anywhere Database, and then click OK to return to the main window.

   The Value field typically defaults to sse_utf8.dbf and appears automatically. To use an encrypted template file, replace this value with sse_encr.dbf.
7. Modify the values of other parameters as necessary by repeating Step 5 and substituting different parameter names in Step 6.

   Table 10 on page 69 lists the parameters and default values for the Generate New Database component.

   You should set the UseDdlFile parameter to FALSE only when you run the Generate New Database component after a schema change.

**8** In the Jobs list, with the Generate New Database job still selected, click Start.

A new database file is generated. Typically, this takes a few minutes.

Table 10 shows the start-up parameters for generating new database components.

Table 10. Selected Parameters for the Generate New Database Component

| Name | Alias | Required /Optional | Default Value and Usage Notes |
|---|---|---|---|
| SQL Anywhere Database | DbfFile | Required | SQL Anywhere database filename to initialize. The default value, sse_utf8.dbf, is a Unicode file that supports all languages. Alternatively, sse_encr.dbf provides standard Sybase encryption of the local database template. |
| DBA Password | DbaPwd | Optional | Password for the DBA account. Default for SQL Anywhere is SQL. Set the DbaPwd parameter to the password for the DBA user ID in the empty database template file. |
| New DBA Password | NewDbaPwd | Optional | The password assigned to the local database administrator account on the Mobile Web Clients.<br>A parameter value specified at component runtime is used *only* for the current run of the Generate New Database component.<br>If no value is specified at runtime, the value specified in an administrative view is used, as described in "Starting Siebel Remote Server Components" on page 58.<br>If no value is specified at component runtime *or* in an administrative view, the default value is the first 8 characters of the enterprise name. If the enterprise name has less than 8 characters, the name is padded with consecutive digits 1234... |
| Table Space | TSpace | Optional | Space name in DB template to store Siebel tables. Do not specify the Table Space parameter unless you intend to build a custom empty database file using the specified table space. |
| Index Space | ISpace | Optional | Space name in DB template to store Siebel indexes. Do not specify the Index Space parameter unless you intend to build a custom empty database file using the specified index space. |

Table 10. Selected Parameters for the Generate New Database Component

| Name | Alias | Required /Optional | Default Value and Usage Notes |
|---|---|---|---|
| Use Transaction Log File | UseTxnLog | Optional | Use when creating a new template file. The default is TRUE. |
| Use DDL File | UseDdlFile | Optional | Use when creating a new template file. The default is FALSE—it means that the schema is read directly from the database.<br>If the value is set to TRUE, it means the schema is read from the DDL file.<br>When a schema change takes place in your environment, set the UseDDLFile parameter to FALSE. GenNewDb will then read the latest schema from the database rather than the DDL file. |
| Interface Tables | IFaceTbls | Optional | Create interface tables and indexes. The default is FALSE. |
| Warehouse Tables | WarehouseTbls | Optional | Create Warehouse tables and indexes. The default is FALSE. |
| Client Db Type | ClientDbType | Optional | Client database engine type. The default is SQL Anywhere. |

# Distributing Database Templates to Siebel Remote Servers

If your deployment includes multiple Remote servers, this section describes how to distribute the database templates. You can distribute it to each Siebel Server by running the distmpl .bat file. This file creates the proper directory on each Siebel Server and copies the database template into the directory.

Alternatively, run the generate new database component at each Siebel Remote server to create the necessary database template file.

### *To invoke distmpl*

1. In a DOS window, run the siebenv.bat file to set the appropriate environment variables.
2. Invoke distmpl.bat to create the destination directories and the database template files.

**Example 1**

Enter: `distmpl \\apsrvr1\siebapp`

where `appsrvr1` is the machine name of the remote server and `siebapp` is the remote server's directory.

### Example 2

Enter: `distmpl s:\siebapp`

where `s:` is the drive on which the remote server's disk is mounted and `siebapp` is the remote server's directory.

# 4 Setting Up Mobile Web Clients

Setting up a Mobile Web Client involves certain tasks for both the Siebel Remote server and the Mobile Web Client. This chapter discusses the required tasks and how to perform them. You must repeat each of these steps for each Mobile Web Client.

Whenever possible, you should implement the entire process of enabling new Mobile Web Clients, rather than relying on end users to finish the configuration.

The table below includes the required tasks for setting up Mobile Web Clients.

Table 11. Tasks for Enabling Mobile Web Clients

| Task | Where Performed |
|---|---|
| Setting Up Mobile Web Client Hardware and Software on page 73 | Mobile Web Client |
| Enabling Network Connectivity on page 74 | Mobile Web Client |
| Establishing Autodial Preferences on page 74 | Mobile Web Client |
| Setting Synchronization Preferences on page 74 | Mobile Web Client |
| Providing Credentials to Synchronize the Local Database on page 81 | Mobile Web Client |
| Enabling Encryption for Synchronization Networking on page 83 | Siebel Server and Mobile Web Client |
| Configuring the Mobile Web Client for Local Database Password Encryption on page 83 | Mobile Web Client |
| Changing the Local Database Password on page 85 | Mobile Web Client |
| Using a Different Data Source on page 85 | Mobile Web Client |
| Registering a Mobile Web Client on page 85 | Siebel Server |
| Using Routing Models on page 89 | Siebel Server |
| Limiting Views Available to Mobile Web Clients on page 98 | Siebel Server |
| Creating Mobile Web Client User Accounts and Privileges on page 100 | Siebel Remote server |

## Setting Up Mobile Web Client Hardware and Software

Install the necessary hardware and software on the Mobile Web Client. This step may include configuring users. You should not change the ODBC code page settings from multiple byte to single byte.

For more information, see *Siebel Installation Guide* for the operating system you are using.

# Enabling Network Connectivity

Install the necessary hardware and software on the Mobile Web Client to allow the client to exchange files with the Siebel Remote server. This step may include choosing communication settings and installing networking cards, modems, and software.

For more information, see *Siebel Installation Guide* for the operating system you are using, and *System Requirements and Supported Platforms* on Siebel SupportWeb.

# Establishing Autodial Preferences

If using Microsoft dial-up networking, mobile users on supported Windows platforms can set an option to automatically connect with their Siebel Remote server when synchronizing.

***To establish autodial preferences***

1. From the application-level menu, select File > Synchronize > Database.
2. In the Siebel Remote Synchronize dialog box, click Setup.
3. From the Siebel Remote Preferences dialog box, click the Connection picklist and choose the appropriate connection.
4. Click OK, and then Synchronize or close the dialog box.

**NOTE:** You must define your phone book entries before synchronizing.

# Setting Synchronization Preferences

This section describes synchronization preferences. The Siebel Remote client reads configuration parameters in the Siebel configuration file, such as siebel.cfg for Siebel Sales, to specify the location of the Siebel Server directories, the Siebel File Server directories, and the Siebel Database installation.

Depending on whether synchronization is started while using the Mobile Web Client or by using the stand-alone synchronizer, the configuration parameters are read from the CFG file that is specified in the Windows shortcut for the client or synchronizer, respectively. By default, the Installer sets up the shortcut to use siebel.cfg as the CFG file, but this setting can be changed as needed.

Before using Siebel Remote, you must set the values for the configuration parameters. The Siebel installation program creates a siebel.cfg file in the client bin directory with default values for each configuration parameter.

If synchronization is performed within the application, that is, using File > Synchronize > Database, configuration information is read from the CFG file of that particular application. For example, if Siebel Call Center is used, then configuration information is read from uagent.cfg.

Siebel 7.7 includes a TrickleSync option (known as autosynchronization in previous releases) to help maintain the frequency of synchronization. Frequent synchronization by mobile users can improve the performance of the system. After initializing their local databases, mobile users enable or disable TrickleSync from User Preferences > DB Synchronization.

For more information about setting up synchronization preferences, see the following topics:


- "Synchronization Parameters in Configuration Files" on page 75
- "Using TrickleSync" on page 76


# Synchronization Parameters in Configuration Files

The configuration parameters that Siebel Remote uses for synchronization include:

- **ClientRootDir.** Name of the Siebel client installation directory. This parameter is located in the [Siebel] section of the CFG file.
- **DockConnString.** Logical network address of the Siebel Server to which you want to connect for synchronization.

  **NOTE:** It is recommended that every synchronization session occur within the corporate firewall. If your deployment must support synchronization through the Internet from outside the firewall, VPN is a good alternative. If there are firewalls on the route from the synchronization client to the server (or from the VPN Server to the Siebel Server machine), then the port for synchronizing with the Siebel Server must be opened on the firewall, and this must be a port *other than* port 80.

  When not using a VPN connection, it is also possible that your Internet Service Provider (ISP) or another host on the route could block communication on this particular port.

  This parameter is located in the [Local] section of the CFG file, and it has the following format:

  *siebel server name:network protocol:sync port #: service:encryption*

  where:

  - *siebel server name* = logical network address of the Siebel Server to which you want to connect for synchronization.
  - *network protocol* = the name of the networking protocol to use. TCPIP is the only valid value and is the default value if nothing is specified.
  - *sync port #* = the TCP/IP port number dedicated to the Synchronization Manager. The default value is 40400 if nothing is specified.

    **NOTE:** You can override the default value by specifying a value for the Synchronization Manager (SynchMgr) component's Static Port Number parameter in the Administration – Server Configuration > Servers view and then restarting the Siebel Server. You can also specify this port number as a command-line option when starting the Siebel Server. If you change this value, client CFG files must be updated to match the server setting.
  - *service* = the TCP/IP service you are requesting. SMI is the only valid value and is the default value if nothing is specified.

- *encryption* = the encryption package you are using. The encryption facility must match the type used by the server. Both MSCRYPTO and RSA are supported.

All elements of the DockConnString value except the Siebel Server name are optional. If an item between other items is omitted, adjacent colons must be retained—as in the third example below, which omits the network protocol element. Examples of valid values for DockConnString are:

```
SIEBAPP1:TCPIP:40400:SMI:RSA

SIEBAPP1:TCPIP:9000

SIBAPP1::9000

SIEBAPP1
```

- **DockRepositoryName.** Name of the Siebel repository that you are currently using. This parameter must have the same value as the Siebel Server repository. This parameter is located in the [Siebel] section of the CFG file.
- **TableOwner.** Name of the database account on the local database where the Siebel schema is installed. The default is SIEBEL. This parameter is located in the [Local] section of the CFG file.
- **DockTxnsPerCommit.** Number of transactions that Siebel Remote applies to the local database before performing a commit. You should set this configuration parameter to a value that satisfies the needs at your site. The default is 500. This parameter is located in the [Local] section of the CFG file.
  - If you synchronize by using the File > Synchronize Database command in a Siebel application, set DockTxnsPerCommit to a high value. Merged transactions will not lock out other users. This enhances the performance of Siebel Remote.
  - If you are using the stand-alone synchronizer, set DockTxnsPerCommit to a high value only if the program will be the only active user on the database. If the program will apply transactions while the user is accessing the database through the application, set a low value to prevent locking out other users while merging transactions.

# Using TrickleSync

The purpose of TrickleSync (which was known as autosynchronization in releases prior to version 7.7) is to improve the overall usability of the system by increasing the frequency of synchronization sessions. Frequent synchronization decreases the volume of transactions for each session and therefore shortens the average connect time.

The TrickleSync agent runs in the background at scheduled times to perform automatic synchronization when connected to the network. One of the options is a synchronization reminder that prompts the user to synchronize if a specified period passes without a synchronization session.

There are two ways to enable TrickleSync:

- An administrator can enable TrickleSync for one or more Mobile Web Clients. See "Setting Up TrickleSync for Users as an Administrator" on page 79 for instructions.

- An initialized Mobile Web Client can enable TrickleSync for that specific Mobile Web Client. See the following procedure for instructions.

For requirements of the synchronization credentials that relate to TrickleSync, see "Providing Credentials to Synchronize the Local Database" on page 81.

### *To enable TrickleSync from an initialized Mobile Web Client*

1. Verify that the local database is initialized.
2. Verify that the Siebel TrickleSync program was added to the Windows Startup Group during the installation process.

   If not, add it by copying it from the Siebel Program group. This allows the TrickleSync agent to start when the Windows session is started.
3. Log in to the local database, and from the application-level menu, choose Navigate > Site Map > User Preferences > DB Synchronization.
4. In the TrickleSync section of the TrickleSync form, define your options using the information below.

| Field | Description |
|---|---|
| Enable TrickleSync check box | Required to enable TrickleSync. |
| User Confirmation check box | Optional. When selected, directs the TrickleSync agent to notify the user before beginning a synchronization and to wait for the user to click OK before proceeding. |
| Maximum Network Latency drop-down list | Required. Sets a network latency that will prevent the TrickleSync agent from invoking a synchronization session. For example, at a connection speed of 56K the threshold may be 200 to 300 milliseconds or higher. The administrator establishes the policy concerning an appropriate value for this setting, but individual users can alter the value for their own synchronizations. |

**NOTE:** Siebel releases prior to release 7.7 also included a Maximum Retries field. Beginning with release 7.7, there is no limit to the number of times that the TrickleSync functionality will attempt to complete a scheduled synchronization.

**5** In the Synchronization Frequency section of the TrickleSync form, select a frequency from the Synchronization Frequency drop-down list, then use the other fields to further define the frequency.

The following table shows Synchronization Frequency drop-down list values:

| Option | Description |
|---|---|
| System boot up | Perform next synchronization after the computer is started and operational. If no network connection is available, try again the next time the computer is rebooted. |
| Mobile Client Startup | Perform synchronization after the Mobile Web Client is started. If no network connection is available, try again the next time the client is started. |
| Minutes | Perform the synchronization at the interval in minutes that is specified in the *Minutes* picklist. TrickleSync synchronizations can be set to occur as frequently as every 10 minutes. |
| Hourly | Perform the synchronization every hour at the minute after the hour that is specified in the *Hourly at* field. For example, if *Hourly at* is set to 25, synchronization would be performed at 1:25, 2:25, and so on. If the computer is not operational at the specified time, then perform the synchronization at the earliest time when the machine is operational. |
| Daily | Perform synchronization every day at the specified time entered in the *Daily At* field. (For example, you can enter 10:00 PM to specify daily synchronization at 10:00 in the evening. You do not need to enter periods in PM.) If the computer is not operational at the specified hour, then perform the synchronization at the earliest time when the machine is operational. |
| Weekly | Perform synchronization every week on the specified day chosen in the *Weekly On* picklist, and at the specified time entered in the *Weekly At* field. If the computer is not operational at the specified time, then perform the synchronization at the earliest time when the machine is operational. |

**6** If you want the Mobile Web Client to display a synchronization reminder if a certain number of days pass without a synchronization, complete the following substeps:

**a** In the Synchronization Reminder section of the TrickleSync form, select the Enable Synchronization Reminder check box.

**b** In the Max Days Between Sync Sessions field, select the number of days after which the Synchronization Reminder dialog box will be displayed, assuming that the Mobile Web Client is up and no synchronizations have taken place during the selected number of days.

The reminder displays the message, `Perform database synchronization now?` The user can respond accordingly.

# Setting Up TrickleSync for Users as an Administrator

The following procedures describe how a Siebel administrator can set up the TrickleSync feature for one or more users. For general information about TrickleSync, see "Using TrickleSync" on page 76.

### *To set up TrickleSync for one user as an administrator*

1. Using an administrator login, from the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote > TrickleSync.
2. In the TrickleSync list, use standard query techniques to locate and select a record for a user who needs TrickleSync set up.
3. In the TrickleSync form, select the Enable TrickleSync check box.
4. In the Synchronization Frequency field, select the time units in which TrickleSync synchronization frequency will be specified.
5. In the corresponding time units field, select the time interval to use between TrickleSync operations for the selected user.

   For example, if you set Synchronization Frequency to Minutes, you can select values between 10 minutes and 55 minutes in the Minutes field.
6. If you wish, complete additional fields in the form, and then save the record.

   For information about the available fields, see "Using TrickleSync" on page 76.

   After the record has been saved, Transaction Processor and Transaction Router operations make sure that the TrickleSync setting changes are placed in the user's outbox on the server. The user must perform one manual synchronization to transfer these setting changes to the Mobile Web Client. After that manual synchronization, TrickleSync automatic synchronization takes effect at times specified by the values of the settings.

### *To set up TrickleSync for multiple users as an administrator*

1. Using an administrator login, from the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote > TrickleSync.
2. In the TrickleSync list, use standard techniques to select multiple users who need TrickleSync set up.
3. In the application menus, select Edit > Change Records, to display the Change Records dialog box.
4. In the 1st Field to Change area of the dialog box, select TrickleSync Enabled from the Field drop-down list.
5. In the accompanying Value field, enter Y.
6. In the 2nd Field to Change area of the dialog box, select TrickleSync Frequency from the Field drop-down list.

7 In the accompanying Value field, enter the time units in which TrickleSync synchronization frequency will be specified.

   Valid values are Minutes, Hourly, Daily, Weekly, System, and Mobile.

8 In the Third Field to Change area of the dialog box, select TrickleSync Schedule.

9 In the accompanying Value field, enter a date and time such as 11/17/2003 4:10 PM.

   For information about how TrickleSync interprets date and time information to schedule synchronization operations, see Table 12 on page 80.

10 Click OK to return to the main window.

   After the selected records have been changed, Transaction Processor and Transaction Router operations make sure that the TrickleSync setting changes are placed in the users' outboxes on the server. Each user must perform one manual synchronization to transfer these setting changes to the Mobile Web Client. After that manual synchronization, TrickleSync automatic synchronization takes effect for that client at times specified by the values of the settings.

Table 12. Interpretation of TrickleSync Frequency and TrickleSync Schedule Values

| If TrickleSync Frequency Value Is: | TrickleSync Schedule Interpretation of Date and Time Is: |
|---|---|
| Minutes | The absolute value of the minutes in the date and time value determines the number of minutes between synchronization attempts. For example, if TrickleSync Schedule is set to 11/17/2003 4:10 PM, synchronization attempts occur every 10 minutes. |
| Hourly | The absolute value of the minutes in the date and time value determines the number of minutes after hour when each synchronization attempt will occur. For example, if TrickleSync Schedule is set to 11/17/2003 4:10 PM, synchronization attempts occur at 10 minutes after every hour. |
| Daily | The absolute values of the hour and minutes in the date and time value determines the hour and minute of the day when each synchronization attempt will occur. For example, if TrickleSync Schedule is set to 11/17/2003 4:10 PM, synchronization attempts occur at 4:10 PM each day. |
| Weekly | The actual day of the week for the date specified in the TrickleSync Schedule value determines the day of the week when weekly synchronizations will be done. The absolute values of the hour and minutes in the TrickleSync Schedule value determines the hour and minute of the day when each synchronization attempt will occur. For example, if TrickleSync Schedule is set to 11/17/2003 4:10 PM, and 11/17/2003 happens to be a Monday, then synchronization attempts occur every Monday at 4:10 PM. |

Table 12. Interpretation of TrickleSync Frequency and TrickleSync Schedule Values

| If TrickleSync Frequency Value Is: | TrickleSync Schedule Interpretation of Date and Time Is: |
|---|---|
| System | TrickleSync does not use Date and Time information for this frequency value. Synchronization is performed after the computer is started and operational, with a working network connection. |
| Mobile | TrickleSync does not use Date and Time information for this frequency value. Synchronization is performed after the Mobile Web Client is started and operational, with a working network connection. |

# Providing Credentials to Synchronize the Local Database

If synchronization authentication is enabled, credentials must be provided in order to synchronize. One possibility is for the Mobile Web Client user to manually enter credentials at each synchronization.

Depending on the type of authentication that is used, a Mobile Web Client user may not be able to change the valid synchronization username and password from inside a Siebel application, although this can be done from outside the application. The local database password, however, can be changed by the Mobile Web Client user as described in "Configuring the Mobile Web Client for Local Database Password Encryption" on page 83.

The following options are provided to allow the user to synchronize without having to manually enter credentials.

- Using the Local Database Credentials to Synchronize
- Remembering the Synchronization Credentials

Both of these features can be managed actively by the Mobile Web Client user, as documented in this topic.

The features can also be managed passively. That is, whenever the user is prompted for credentials to synchronize, the dialog box includes fields to make any of the changes described in this section. The user is prompted to manually enter synchronization credentials if any of the following conditions apply:

- The Mobile Web Client user has not opted to remember the synchronization password or to use the local database password to synchronize.
- The Mobile Web Client user has opted to use the local database credentials to synchronize, but those credentials are not the valid synchronization credentials.
- Despite how the Mobile Web Client user has opted to provide synchronization credentials, the user is prompted for credentials whenever the synchronization credentials are changed by an administrator at the server level.

## Using the Local Database Credentials to Synchronize

The Mobile Web Client user can opt to have the local database credentials submitted automatically as part of synchronization requests. This requires that the local database password be identical to the valid synchronization password.

If synchronization uses the local database credentials and these credentials are the valid synchronization credentials, then the Mobile Web Client user is not prompted to enter credentials manually at synchronization.

### *To use the local database credentials for synchronization*

1. Log in to the local database.
2. From the application-level menu, choose Navigate > Site Map > User Preferences > DB Synchronization.
3. In the DB Synchronization form, check the Use Local Credentials check box, and then click Save.

   If you had existing entries in the User ID, Password, and Remember Credentials fields, they will disappear.

To discontinue using local database credentials for synchronization, clear the Use Local Credentials check box in the DB Synchronization form.

For information about setting the authentication method for synchronization, see "Selected Synchronization Manager Start-up Parameters" on page 64.

For information about changing the local database password, see "Configuring the Mobile Web Client for Local Database Password Encryption" on page 83.

## Remembering the Synchronization Credentials

The Mobile Web Client user can opt for the Siebel application to remember the synchronization credentials. For example, it is useful to remember the synchronization credentials in a situation in which the Mobile Web Client user changes the local database password so that it differs from the valid synchronization password.

Remembering the synchronization credentials is not available if the Mobile Web Client user sets synchronization to use the local database credentials.

If synchronization credentials are remembered, then the Mobile Web Client user is not prompted for credentials to synchronize. If the synchronization password is not remembered, and the local database credentials are not used to synchronize, then the user will be prompted to manually enter credentials for each synchronization.

The remembered synchronization password is stored in the local database with an added layer of 128-bit encryption.

**CAUTION:** For TrickleSync to perform automatic synchronizations when the Mobile Web Client is connected to the server, the synchronization credentials must be remembered or the local database credentials must be used for synchronization. If neither of these settings is active, then the application will prompt for credentials for each synchronization.

***To remember synchronization credentials***

1. Log in to the local database.
2. From the application-level menu, choose Navigate > Site Map > User Preferences > DB Synchronization.
3. In the DB Synchronization form, check the Remember Credentials check box.
4. Enter your User ID.
5. In the Password field, enter the synchronization password that was assigned by an administrator at the server level.
6. If the Use Local Credentials check box was previously checked, it should be cleared. If it is not cleared, click the check box to clear the check mark.
7. Click Save.

## Enabling Encryption for Synchronization Networking

To use encryption, both the server and the client must enforce encryption in their connection parameters. If these parameters do not match, connection errors will occur.

- For the server, set the Encryption Type parameter for the Synchronization Manager server component to MSCRYPTO or RSA. For information about these encryption types, see *Security Guide for Siebel eBusiness Applications*.

  You can use the Server Manager command line or the Administration - Server Configuration screen to set the Encryption Type parameter. For Server Manager, the appropriate command for using RSA encryption is as follows:

  ```
  srvrmgr> change param Crypt=RSA for comp SynchMgr
  ```

  After setting the parameter, you must restart the Synchronization Manager.

- For the client, encryption is the fifth parameter in the DockConnString for the local database that is defined in the application's .cfg file on the Mobile Web Client. Override the default NONE with MSCRYPTO or RSA. You may use either AASRVR:TCPIP:40400:SMI:MSCRYPTO or RSA, or APPSRV::::MSCRYPTO or RSA.

Siebel eBusiness Applications use MSCRYPTO or RSA to encrypt data traffic between the server and the Mobile Web Client. It is configured at the component level for replication with these clients and the server. It is *not* used to encrypt the local database nor the data in it. Also, it is not used for communication with the database.

**NOTE:** Secure Sockets Layer (SSL) is not a supported encryption method for synchronization.

# Configuring the Mobile Web Client for Local Database Password Encryption

You can enable RSA SHA-1 encryption, a one-way hashing algorithm, to be applied to the local database password before it is passed to the local database for authentication.

Encryption adds a layer of security by disconnecting the password stored in the local database from the unencrypted password entered by the user.

Encryption of the local database password requires configuration on the Remote server and on the Mobile Web Client.

To enable local database password encryption, the parameters in the application .cfg file on the Mobile Web Client should have the values shown in Table 13. Most of these settings will be made automatically, but you should confirm them.

Table 13. Mobile Web Client .cfg File Parameters for Local Database Password Encryption

| Section | Parameter and Value | Comment |
|---|---|---|
| [InfraSecMgr] | SecAdptName = DBSecAdpt | DBSecAdpt is the default name for the section in the .cfg file that defines the database security adapter. This is a default setting. |
| | SecAdptMode = DB | DB, or database authentication mode, is the only supported authentication mode for local database authentication. This is a default setting. |
| [DBSecAdpt] | SecAdptDllName = sscfsadb | The dll name for the default security adapter. This is a default setting and applies in Windows and UNIX environments. |
| | DataSourceName = Local | This security adapter section applies to the Local datasource section in this .cfg file. This is a default setting. |
| | DBSecAdpt_PropogateChange = TRUE | TRUE allows the user to change the local database password. This is a default setting. |
| [Local] | DSHashUserPwd = TRUE | Encrypt the local database password in the Local data source. If the Remote Server is configured to enable local database password encryption, then the Upgrade Wizard makes this setting when the local database is initialized. |
| | DSHashAlgorithm = RSASHA1 | Use RSA SHA-1 encryption. If the Remote Server is configured to enable local database password encryption, then the Upgrade Wizard makes this default setting when the local database is initialized. |

For information about configuring the Remote Server to support local database password encryption, see "Configuring the Remote Server for Local Database Password Encryption" on page 105.

# Changing the Local Database Password

By default, the password that the Siebel application uses to connect to the local database is the same as the password used to synchronize the local database with the Enterprise or Regional database. The Mobile Web Client user can redefine the local database password to be different from the valid synchronization password.

Maintaining different credentials for accessing the local database and for synchronizing the local database provides a layer of security against both unauthorized access to the local database and unauthorized access to data on the server database.

### *To change the local database password by using the Mobile Web Client*

1. Start the Mobile Web Client and log in to the local database.

   If the local database password has not been changed previously, then the password is the same as that used to initialize or synchronize the local database.

2. From the application-level menu, choose Navigate > Site Map > User Preferences > Profile.
3. Enter the new password in the Password and Verify Password fields, and then click Save.

# Using a Different Data Source

If your organization requires data and applications to be stored on separate devices, you may wish to use a different data source from the one specified by default in the [Local] section of the CFG file. Use the following procedure to modify the information in the [Local] section.

**CAUTION:** Be sure to modify the existing Local data source description in the CFG file. Do not add additional local data sources to the CFG file, as these are not supported.

### *To use a different local data source*

1. In the [Local] section of the CFG file, edit the data source information to reflect the actual location of your sse_data.dbf file.
2. In the ODBC Data Source Administrator, select the SEAW Local Db default instance and click Configure.
3. Click the Database tab.
4. Edit the Database file name to reflect the actual location of your database file and click OK twice.

# Registering a Mobile Web Client

This section describes how to register a Mobile Web Client. It assumes the Siebel Administrator has previously set up this user as a user in the Siebel application.

If you need to add a large number of Mobile Web Clients to the application, you can consider using EIM to perform a batch load. If your deployment includes Replication Manager, keep in mind that this EIM functionality is only supported on the HQ node. For more information on how to load data into interface tables and then populate base tables, see *Siebel Enterprise Integration Manager Administration Guide*.

Make sure that mobile users have access to the following views in their responsibilities:

- **Mobile User Summary View.** Shows current state of the Siebel Remote Client. Useful for obtaining information about when the client was extracted, when it was initialized, and the last synchronization session. This view is displayed by navigating to User Preferences > Mobile User Summary.
- **Dock Session Log.** Shows information about synchronization sessions, including information about conflicts and other results of those sessions. This view is displayed by navigating to User Preferences > Remote Status.
- **Auto Synchronization View.** Displayed by navigating to User Preferences > DB Synchronization. The feature known as Auto Synchronization in previous releases is now called TrickleSync. However, the view for configuring TrickleSync is still called the Auto Synchronization view. Allows each mobile user to perform the following tasks:
  - Enable, disable, or configure TrickleSync and synchronization reminders
  - Set up synchronization authentication credentials
  - Set parameters that govern synchronization session behavior

For additional information about setting up employees and adding access to views, see *Applications Administration Guide*.

### *To register a new Mobile Web Client on a node*

1. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote > Mobile Clients.
2. In the Parent Server form, use standard query techniques to select the appropriate node.

   If your deployment does not use Replication Manager, then the appropriate node is HQ.
3. In the Mobile Clients list, click New.
4. In the new Mobile Clients record, enter the Mobile Web Client name in the Mobile Client field.

   **CAUTION:** The Mobile Web Client name must be entered in uppercase letters and be eight characters or less. It is recommended that you use the Mobile Web Client User ID (see next step) as the Mobile Web Client name. It *can* contain only Roman, alphanumeric, and the _ or – characters. It *cannot* include spaces, periods, or other invalid characters ( / \ : * ? " < >|) as in the DOS file naming schema. Siebel Remote uses the Mobile Web Client name to create inbox and outbox directories on the Siebel Server.
5. In the User ID field, click the select button, and choose the User ID of the mobile user.

   User ID is used to access the user's local database during initialization and synchronization.

6 In the Routing Model field, click the select button, and choose the data routing model to which the mobile user belongs.

For information on data routing models, see "Using Routing Models" on page 89.

7 In the Language(s) field, click the select button, and choose the preferred language or languages for the mobile user. If the preferred languages are not available, click New and follow the instructions in the dialog boxes to add these choices.

A language preference allows the mobile user to download data in a preferred language, or languages, for the following dock objects that contain Translation Tables: LOV, Product, Literature, Catalog, Catalog Category. This helps optimize the size of the local databases.

Dock objects are logical groups of tables with special schema structures to synchronize data between a server database and a mobile database in a coherent manner.

By default, the Language(s) field is empty, which means the mobile user will receive data in *all* the languages for these dock objects.

Data for each of the other dock objects is routed to Mobile Web Clients based on the normal visibility rules.

8 Complete the remaining fields as appropriate.

The Sync Password field is used by the Synchronization Manager if the authentication method in the Siebel Server Component Parameters is set to Siebel. Set the password in this field and give it to the mobile user.

The App Server Name field is not populated until the Database Extract is run for the mobile user. At the time the Mobile Web Client record is created (S_NODE) the APP_SERVER_NAME field is NULL.

**NOTE:** If you use EIM to load mobile user records, records without an HQ node as the parent node do not appear in the Mobile Clients view. The parent node is stored in the following two columns: EIM_NODE.par_name and EIM_NODE.par_node_type_cd. Although these columns are not required for EIM, they are required for Siebel Remote. When you enter mobile users using the Mobile Clients view, these columns are populated by default.

9 If the Standard routing model is not used, then from the application-level menu, choose Navigate > Site Map > Administration - Application > Responsibility List > Responsibilities, and select the corresponding Responsibility with the Routing Model suffix.

The corresponding Responsibility with the Routing Model suffix relates to the data routing model assigned in Step 6 above. For more information about corresponding routing models and how these help optimize the size of local databases, see "Limiting Views Available to Mobile Web Clients" on page 98.

a In the Users list, add a new record.

b In the Add Users dialog box, select the mobile user and click OK.

# Routing Rules and Dock Objects

This section provides an overview of routing rules and dock objects.

Routing rules provide the logic for Database Extract and Transaction Router server components to perform their tasks. These rules reflect the data visibility and data access policies within the application.


- "Routing Rules"
- "Dock Objects" on page 88


For additional information about this topic, see *Configuring Siebel eBusiness Applications*.

# Routing Rules

Routing rules are SQL statements that Transaction Router and or Database Extract use to evaluate what records should be routed to each mobile user. Siebel releases can include more than a thousand active routing rules. They are designed to accomplish the following:

- Protect data integrity
- Allow the same visibility of data when users are connected to the server
- Facilitate access control
- Maintain application logic and functionality

The types of routing rules are the means to implement these requirements.

**NOTE:** Changes to routing rules provided by Siebel Systems require Siebel Expert Services working in conjunction with Siebel engineering. However, for extension tables generated using Database Extensibility, routing rules can be generated using a wizard provided within Siebel Tools.

# Dock Objects

A Dock Object is a logical grouping of tables with a special schema structure to synchronize data between a server database and a mobile database in a coherent manner. Routing rules belong to dock objects.

Generally, there are three types of dock objects in the Siebel architecture:

- Enterprise dock objects
- Private dock objects
- Limited dock objects

Enterprise dock objects are those objects that are visible to all users within the application. Examples include currency and catalog.

Private dock objects are those objects that will not be routed to mobile users. If data is created on Mobile Web Clients, it will be sent up to the server. However, updates to them will not be returned to the client. Data visibility of private dock objects to users is not used in the routing consideration.

**CAUTION:** After making any change to the database schema, run the Generate New Database Template component and reextract all mobile clients, or use Siebel Anywhere kits to distribute the change to all mobile clients. It is strongly recommended that you do this even if the schema change only affects private dock objects, because individual tables in a private dock object may become visible to mobile clients at a later time, and problems can occur if server and local database structures do not match.

Limited dock objects are those objects whose data may or may not be visible to a particular user—most user data is of this type. These objects have routing rules, as described earlier, that determine which records are routed to a particular mobile user.

Routing rules are SQL statements that determine whether a given piece of data should be routed to a given user. Routing rules embody the data visibility and data access built into Siebel eBusiness applications. Any transactions in the system are associated with a set of routing rules that may cause the transactions to be routed to a mobile user.

A routing model is a collection of routing rules. The next section includes a detailed description of the routing models available. A mobile user can now be defined as associated with any of the routing models. Thus, transaction routing will behave differently for each mobile user, depending on the routing model the user is associated with.

# Using Routing Models

This section describes the Routing Models available to reduce the amount of data replicated to mobile users. For a majority of mobile users, the MOBILE CLIENT - STANDARD, MOBILE CLIENT - EXTRACT ONLY, or Executive Management routing model is adequate. However, for users who need to minimize their local database size, using one of the specialized routing models may be appropriate.

**NOTE:** Before you deploy Siebel Remote with any of these specialized routing models, it is strongly recommended that you discuss this with a Siebel technical resource.

Routing models determine what data will be extracted to, and what follow-on transactions will be routed to, mobile users. By careful application of specific docking rules, local database sizes are reduced, and, as a result, so are synchronization times and transaction application times. The docking visibility rules determine which records from the server database are propagated to each mobile user.

The flexibility available through data routing models helps align more closely the extraction of data and routing of transactions with the specific needs of the mobile users. This helps minimize the size of the local databases and the connect time to download a database extract and to synchronize.

To match the data routed to mobile users assigned to some of the routing models, administrators can limit the views for these users. This functionality applies to mobile users assigned to the following out-of-the box routing models: Sales Rep Standard, Sales Mgr Standards, Field Technician, Analyst, Sr. Field Engineer, and Field Engineer.

For more information about limiting views available to mobile users and how to do this, see "Limiting Views Available to Mobile Web Clients" on page 98.

**CAUTION:** Make sure Routing Models are consistent with the responsibilities and positions of the mobile users. The responsibilities and position of an employee determine the access that person has to the Server database. Balancing the data routing model with a user's access helps to optimize the size of that user's local database. This also helps to minimize synchronization time.

Each mobile user will be associated with one Routing Model. A Routing Model includes a set of routing rules. Union of the routing rules determines whether a record will be routed to a Mobile Web Client.

The descriptions that follow include the Routing Models available out-of-the-box.

Routing models also impact what data will be extracted to, and what follow-on transactions will be routed to, regional servers. See "Routing Groups and Routing Rules" on page 164. for more details regarding routing groups for regional servers.

## Routing Model Enhancements in Siebel Release 7.7

In Siebel Release 7.7, the following enhancements have been made to routing model functionality:

- Routing model definitions can now exclude information (prevent it from being routed) at the dock object level. For example, the Enterprise dock object can now be excluded.
- Three new routing models are provided with the release. For information about them, see the following topics:
  - "Handheld User Routing Model" on page 93
  - "Life Science User Routing Model" on page 93
  - "Selective Retrieval Routing Model" on page 94
  - "Using the Selective Retrieval Routing Model" on page 94

## Field Sales Representative Standard Routing Model

Users assigned the Sales Representative Standard routing model receive a database extract and follow-on transactions dealing with information relevant to a salesperson.

To display a list of views typically associated with this routing model, from the application-level menu, choose Navigate > Site Map > Administration - Application > Responsibilities, and query for a responsibility containing the name of the routing model.

## Field Sales Manager Standard Routing Model

Users assigned the Sales Manager Standard routing model will receive a database extract and follow-on transactions dealing with information relevant to a sales manager. The views associated with this model include those that are typical for the Sales Representative Standard in addition to the manager's Team's Accounts, Contacts, and Opportunities.

## Field Technician Routing Model

Users assigned the Field Technician routing model receive a database extract and follow-on transactions dealing with information relevant to a field technician. It provides the Field Technician with the most basic or essential data to complete the required work in the most timely manner. To display a list of views typically associated with this routing model, from the application-level menu, choose Navigate > Site Map > Administration - Application > Responsibilities, and query for a responsibility containing the name of the routing model.

## Senior Field Engineer Routing Model

Users assigned the Senior Field Engineer routing model receive a database extract and follow-on transactions dealing with information relevant to a senior field engineer. It includes data to support interaction with others and the service center, access to product and parts data, access to solution knowledge bases, basic sales functionality, and customer and asset data. To display a list of views typically associated with this routing model, from the application-level menu, choose Navigate > Site Map > Administration - Application > Responsibilities, and query for a responsibility containing the name of the routing model.

## Field Engineer Routing Model

Users assigned the Field Engineer data routing model receive a database extract and follow-on transactions dealing with information relevant to a Field Engineer. This model provides the Field Engineer with all the data required to perform and debrief a complex job requiring an acceptable level of historical data regarding customer and asset history. With this data the Field Engineer can order parts, generate service invoices, debrief time and expense on site, and perform cycle counts. Debriefing is the reporting of material usage, time, and expenses to service managers after completing an activity. To display a list of views typically associated with this routing model, from the application-level menu, choose Navigate > Site Map > Administration - Application > Responsibilities, and query for a responsibility containing the name of the routing model.

## Consultant Routing Model

Users assigned the Consultant routing model receive a database extract and follow-on transactions dealing with information relevant to a consultant. The option is a limited use license of the professional services functionality. Using this routing model, the data will be limited to time sheets, expense reports, limited project information, calendar, and employee skills. To display a list of views typically associated with this routing model, from the application-level menu, choose Navigate > Site Map > Administration - Application > Responsibilities, and query for a responsibility containing the name of the routing model.

## Analyst Routing Model

Users assigned the Analyst routing model work at research firms. The information that is routed to users assigned to the Analyst model includes Contacts, Accounts, Assets, Service Requests, and Quotes. The analysts also receive Activities, but only those activities that are related to Quotes, Opportunities, or Service Requests. An analyst also is routed Project Items such as the Team Workbook, Time and Expense, and Time sheets. To display a list of views typically associated with this routing model, from the application-level menu, choose Navigate > Site Map > Administration - Application > Responsibilities, and query for a responsibility containing the name of the routing model.

## Minimal Data Routing Model

Users assigned the Minimal Data routing model receive a database extract and follow-on transactions dealing only with Calendar Items, My Contacts, My Accounts, and My Opportunities. To display a list of views typically associated with this routing model, from the application-level menu, choose Navigate > Site Map > Administration - Application > Responsibilities, and query for a responsibility containing the name of the routing model.

## Mobile Partner User Routing Model

Mobile Partner User is a routing model that is used by Partner Sales and Service Representatives that have implemented a mobile version of the Siebel PRM Partner Portal. Users assigned the Mobile Partner User routing model receive a database extract and follow-on transactions dealing only with Accounts, Activities, Assets, Contacts, Correspondence, Inventory Locations, Opportunities, Orders, Price Lists, Products, Proposals, Quotes, Revenues, Service Requests, and Sources.

## Mobile Client – Standard Routing Model

Unlike the routing models above, this model allows a full set of data related to the user being routed based on visibility rule configuration in the Siebel Repository. There are some exceptions to this. The Subordinate Activities are not routed to Supervisors. Also, a few other visibility rules do not apply.

## Mobile Client – Extract Only Routing Model

The difference between this and the Mobile Client – Standard model is that this model will not allow synchronization. Hence, a local database with this model reflects the snapshot of application data with respect to a user.

## Executive Management Routing Model

This is an extract-only routing model designed for executive managers so they can use the Mobile Web Client to perform their everyday tasks. The model excludes all manager rules so that only data visible to them will be routed to their Mobile Web Clients.

## Handheld User Routing Model

This routing model is similar to the Mobile Client - Standard routing model, but it excludes some routing rules related to Products that the Mobile Client - Standard routing model includes. It also routes the following kinds of additional data for those who use the Mobile Web Client and Siebel Remote to synchronize Siebel Handheld with the Enterprise database:

- Barcoding
- Siebel Handheld administrative data

## Life Science User Routing Model

Each user assigned to the Life Science User routing model receives a database extract and follow-on transactions containing data required by mobile users of the Siebel Life Sciences Siebel Industry Application (SIA).

This routing model is similar to the Mobile Client - Standard routing model, but it routes more data to Mobile Web Clients and contains some extra rules that are required by Siebel Life Sciences applications. These extra rules are for the following dock objects:

- Activity
- Assets
- Barcode
- DocAgreement
- Expense
- GroupNews
- Handheld Admin Application
- InvoicableCharge
- Order
- OrgSource
- Party
- Product
- Project
- ProjectItem

- PrspCon
- RepairPart
- ServiceRequest
- Time Sheet Item

This model can be used with any responsibility that is valid for the Mobile Client - Standard model. There is no dedicated Life Science User responsibility.

## Selective Retrieval Routing Model

The Selective Retrieval routing model allows users to designate individual records to be included in Mobile Web Client synchronizations. Appropriately used, this routing model can reduce the volume of data transferred during synchronization, which reduces both synchronization time and the size of the local database. The Selective Retrieval routing model and the Time Filters feature both confer these benefits, but Time Filtering excludes aged data from synchronization by all users, whereas the Selective Retrieval routing model operates on a per-user basis, with per-user specifications regarding which data to synchronize.

**NOTE:** This functionality is appropriate for mobile users who use *both* connected and disconnected (local) versions of Siebel applications. The feature may not be appropriate for users who use only the Mobile Web Client, or who require access to features such as quotes or forecasting. This is because Selective Retrieval limits application functionality in some areas, and because designating records for synchronization requires a direct connection to an Enterprise server (or a Regional server, if the Regional server contains the records to be designated for synchronization).

Siebel v7.7 supports the selective retrieval functionality for the following objects:

- Accounts
- Contacts
- Opportunities
- Projects
- Service Requests

To display a list of views typically associated with this routing model, from the application-level menu, choose Navigate > Site Map > Administration - Application > Responsibilities, and query for a responsibility containing the name of the routing model.

For information about how to use selective retrieval functionality, see "Using the Selective Retrieval Routing Model" on page 94.

## Using the Selective Retrieval Routing Model

This topic provides additional information about the Selective Retrieval routing model, which was introduced in "Selective Retrieval Routing Model" on page 94.

When a user who is assigned to the Selective Retrieval routing model designates a record for synchronization, that record will be synchronized for any Mobile Web Client that is associated with the designating User ID. Records that are related to the designated record will also be synchronized, to maintain data integrity. For example, if an account record is designated, the records for the contacts and addresses of that account are related and will be synchronized. Similarly, if a contact record is designated, the account record or records for that contact will be synchronized.

If the user designates records for synchronization before the database is extracted for any associated Mobile Web Client, the Transaction Router will have fewer items to handle on the first subsequent synchronization. However, a user can designate records for synchronization at any time.

When a selective retrieval user deselects a record for synchronization, that record will be removed from the local database of the Mobile Web Client associated with the deselecting User ID upon the next synchronization, along with related records.

To use the selective retrieval functionality, the user's Mobile Web Client must be assigned to the Selective Retrieval routing model. The user must also be assigned to a responsibility that corresponds to the Selective Retrieval routing model, such as the Selective Retrieval Routing Model responsibility.

For information about assigning a routing model when a Mobile Web Client is registered, see "Registering a Mobile Web Client" on page 85. For information about changing a routing model at a later time, see "Changing Routing Models" on page 143. For information about assigning responsibilities, see *Security Guide for Siebel eBusiness Applications*.

The remainder of this topic contains procedures that describe how either selective retrieval users or administrators can designate records for inclusion or exclusion from synchronization.

The following procedure describes how a selective retrieval user can designate a record to be included in that user's synchronizations.

### *To designate a record for selective retrieval during synchronization, as a user*

1. Log in to the Siebel application using a direct connection to the server, rather than the local database.

   Log in using the User ID that is associated with the Mobile Web Client that will be using the Selective Retrieval routing model and feature.

2. Navigate to the screen that contains the record you want to synchronize.

   Supported screens include Accounts, Contacts, Opportunities, Projects, and Service Requests.

3. Navigate to a standard view for the selected screen, such as the My Accounts view or the All Accounts view for the Accounts screen, or the Project List view for the Projects screen.

4. Use standard query techniques to locate and select a record that you want to include in future synchronizations.

5. In either the list or the form displayed for the record, click the Menu button and choose Make Available Offline.

The following procedure describes how an administrator can designate a record for synchronization by a specific selective retrieval user.

***To designate a record for synchronization by a specific user, as an administrator***

1. Log in to the Siebel application using an administrator User ID and a direct connection to the server.
2. Navigate to an administration view that displays the record you want to synchronize:
   - For Accounts, from the application-level menu, choose Navigate > Site Map > Accounts > Accounts Administration.
   - For Contacts, from the application-level menu, choose Navigate > Site Map > Contacts > Administration.
   - For Opportunities, from the application-level menu, choose Navigate > Site Map > Opportunities > Opportunities Administration.
   - For Projects, from the application-level menu, choose Navigate > Site Map > Administration - Data > Projects.
   - For Service Requests, from the application-level menu, choose Navigate > Site Map > Service Request > Service Request List, and use the visibility filter to display All Service Requests.
3. Use standard query techniques to locate and select a record that you want to include in future synchronizations for a specific user.
4. For the appropriate record type listed below, click the Menu button and select Columns Displayed, and then move the indicated field to Selected Columns and click Save.
   - For Accounts, display the Account Team field in the Accounts list.
   - For Contacts, display the Contacts Team field in the Contacts list.
   - For Opportunities, skip this step. (The Sales Team field is already visible in the Siebel Qualified Lead - Fast Track Career Consulting form.
   - For Projects, skip this step. (The Members field is already visible in the Details area of the Project form.)
   - For Service Requests, skip this step. (Service Requests have a single owner. The owner is the only user for whom the record can be designated for synchronization.)
5. Perform the action listed for the appropriate record type.
   - For Accounts, click the select button in the Account Team field to display the Account Team Member dialog box. In the Selected list, select the user to synchronize this record, scroll to the right, select the Available Offline check box, and click OK.
   - For Contacts, click the select button in the Contact Team field to display the Access List dialog box. In the Selected list, select the user to synchronize this record, scroll to the right, select the Available Offline check box, and click OK.
   - For Opportunities, click the select button in the Sales Team field to display the Team Members dialog box. In the Selected list, select the user to synchronize this record, select the Available Offline check box, and click OK.
   - For Projects, in Details area of the Projects form, click the select button in the Members field. In the Selected list, select the user to synchronize this record, scroll to the right, select the Available Offline check box, and click OK.

- For Service Requests, in the All Service Requests list, click the Menu button and choose Make Available Offline.

The following procedure describes how a selective retrieval user can designate a record to be removed from that user's future synchronizations and local database.

### *To remove a record from the local database and future synchronizations, as a user*

1. Log in to the Siebel application using *either* a direct connection to the server or the local database.

   Log in using the User ID that is associated with the Mobile Web Client that will be using the Selective Retrieval routing model and feature.

2. Navigate to the screen that contains the record you want to stop synchronizing and remove from the local database.

   Supported screens include Accounts, Contacts, Opportunities, Projects, and Service Requests.

3. Navigate to a standard view for the selected screen, such as the My Accounts view or the All Accounts view for the Accounts screen.

4. Use standard query techniques to locate and select a record that you want to stop synchronizing and remove from the local database.

5. In either the list or the form displayed for the record, click the Menu button and choose Make Unavailable Offline.

The following procedure describes how an administrator can designate a record to be removed from future synchronizations and the local database for a specific selective retrieval user.

### *To remove a record from a specific user's local database and synchronizations, as an administrator*

1. Log in to the Siebel application using an administrator User ID and a direct connection to the server.

2. Navigate to an administration view that displays the record that you want a specific user to stop synchronizing:

   - For Accounts, from the application-level menu, choose Navigate > Site Map > Accounts > Accounts Administration.
   - For Contacts, from the application-level menu, choose Navigate > Site Map > Contacts > Administration.
   - For Opportunities, from the application-level menu, choose Navigate > Site Map > Opportunities > Opportunities Administration.
   - For Projects, from the application-level menu, choose Navigate > Site Map > Administration - Data > Projects.
   - For Service Requests, from the application-level menu, choose Navigate > Site Map > Service Request > Service Request List, and use the visibility filter to display All Service Requests.

3 Use standard query techniques to locate and select a record that you want a specific user to stop synchronizing.

4 For the appropriate record type listed below, click the Menu button and select Columns Displayed, and then move the indicated field to Selected Columns and click Save.

- For Accounts, display the Account Team field in the Accounts list.
- For Contacts, display the Contacts Team field in the Contacts list.
- For Opportunities, skip this step. (The Sales Team field is already visible in the Siebel Qualified Lead - Fast Track Career Consulting form.
- For Projects, skip this step. (The Members field is already visible in the Details area of the Project form.)
- For Service Requests, skip this step. (Service Requests have a single owner. The owner is the only user for whom the record can be removed from synchronization.)

5 Perform the action listed for the appropriate record type.

- For Accounts, click the select button in the Account Team field to display the Account Team Member dialog box. In the Selected list, select the user to stop synchronizing this record, scroll to the right, clear the Available Offline check box, and click OK.
- For Contacts, click the select button in the Contact Team field to display the Access List dialog box. In the Selected list, select the user to stop synchronizing this record, scroll to the right, clear the Available Offline check box, and click OK.
- For Opportunities, click the select button in the Sales Team field to display the Team Members dialog box. In the Selected list, select the user to synchronize this record, clear the Available Offline check box, and click OK.
- For Projects, in Details area of the Projects form, click the select button in the Members field. In the Selected list, select the user to stop synchronizing this record, scroll to the right, clear the Available Offline check box, and click OK.
- For Service Requests, in the All Service Requests list, click the Menu button and choose Make Unavailable Offline.

# Limiting Views Available to Mobile Web Clients

Every mobile user is assigned a routing model to help optimize the size of that user's mobile database. To match the data routed to mobile users assigned to some of these routing models, administrators can limit the views for these users. Administrators should limit views for all mobile users except those who use the Standard routing models or models that are similar to the Standard models, such as Life Sciences User routing model or the Handheld User routing model.

Determining what views a mobile user sees depends upon the responsibilities assigned to that user and the views attached to these responsibilities. The list of views attached to a specific responsibility are listed and administered in the Responsibilities view within Siebel's Administration - Application screen. For more information about this topic, see "Using the Responsibilities View" on page 99.

Each view attached to a responsibility includes a Local Access flag. Limiting views is accomplished by turning off the Local Access flag for views that a particular mobile user does not need.

**NOTE:** A particular user may have two or more responsibilities that include the same view. If Local Access=False (cleared) for this view in *any* of the user's responsibilities, then the view is *not* available on the Mobile Web Client. This view will not appear in any Navigation element, including the Site Map.

One way to limit views that are available to mobile users is to make sure that responsibilities that include "Routing Model" in the responsibility name include even views that mobile users should *not* see. Then, turn off Local Access for those views in the Routing Model responsibilities. Provided that each mobile user is assigned to a Routing Model responsibility, taking this action is sufficient to limit the mobile users' access to the applicable views, so that you need not inspect local access settings for other responsibilities.

The responsibilities with the Routing Model suffix include: Sales Representative Routing Model, Sales Manager Routing Model, Minimal Data Routing Model, and Analyst Routing Model.

A mobile user will have access to that user's normal views when operating either as a dedicated Web client or a Web client. The limitation on the visibility of views only applies while operating as a Mobile Web Client.

## Using the Responsibilities View

The Responsibilities view within the Administration - Application screen includes a Responsibilities list and two subordinate lists:

- Views list of the associated views added to individual responsibilities.
- Users list of the users assigned to individual responsibilities.

  It also displays a Routing Model field that is read-only.

In this view you can do the following:

- Adjust the Local Access flag—turn it off or turn it on.
- Add views to a responsibility.
- Read the Routing Model assigned to a user.

  **NOTE:** To assign users to routing models, you would choose Navigate > Site Map > Administration - Siebel Remote > Mobile Clients from the application-level menu, and proceed as described in "Registering a Mobile Web Client" on page 85.
- Add users to a responsibility.

The procedure below describes how to limit the visibility of views for mobile users assigned to the routing models listed at the beginning of this section.

### *To limit the visibility of views for Mobile Web Clients*

1 Determine the views a mobile user does not need.

2 From the application-level menu, choose Navigate > Site Map > Administration - Application > Responsibility List > Responsibilities, and select the responsibility with a view this mobile user does not need.

3 In the Views list, select the view this mobile user does not need.

4 Clear the Local Access check box to remove the check mark.

5 Repeat Steps 3 and 4 for all the views the mobile user does not need that are associated with this responsibility.

## Using the Views View

The Views view within the Administration - Application screen includes a Views list and a subordinate Responsibilities list. This allows an administrator to determine which responsibilities are associated with a particular view. An administrator can also determine whether those responsibilities allow a view to be displayed on the Mobile Web Client.

The Views list includes the entire list of views in the application. The Local Access flag is a default setting for administering these views, for example adding views to responsibilities in the Responsibilities view discussed in "Using the Responsibilities View" on page 99.

The Responsibilities list includes the associated responsibilities added to the individual views in the Views list. The Local Access flag determines whether a view is available for mobile users with that responsibility.

# Creating Mobile Web Client User Accounts and Privileges

If you want to authenticate Mobile Web Clients when they synchronize, you need to create accounts for each client depending on the authentication method you choose. This section describes the requirements for the authentication methods available for synchronization.

- If you use AppServer authentication, you must create a user account on the server for each Mobile Web Client. Examples include:
  - If the Remote server is running in a supported Windows environment, create a Windows user account and password for each Mobile Web Client you defined. You can create these user accounts and passwords either on the Remote server or on the domain.
  - If the Remote server is running in a UNIX environment, create the user on the application server that will host Siebel Remote. For more information on how to create users in this environment, see *Siebel Installation Guide for UNIX: Servers, Mobile Web Clients, Tools*.

    **NOTE:** The user name and password must both be in uppercase letters and be eight characters or less. They may contain only single-byte, alphanumeric, and the _ or – characters. They may *not* include spaces, periods, or other characters that are invalid in a DOS file naming scheme, such as ( / \ : * ? " < > |) .

  - If you use the Mobile Web Client's change password functionality while connected to the server, the server password changes and is stored in a database-specific table, not a Siebel table. Therefore it is not synchronized to the local database. The mobile user would still use that user's old password to access the user's local database until the node is reextracted.

- If you use Database authentication, you must create a database account and password for each Mobile Web Client.
- If you use the Siebel authentication method, you must set the password for each Mobile Web Client on the Mobile Clients view in the Administration - Siebel Remote screen, and you must provide the password to mobile users so they can use it to synchronize with the Siebel Server.
- If you use SecurityAdapter authentication, then you must create accounts as required by your third-party authentication system. For example, you may have to create records in an LDAP or ADS directory.

If you do not use authentication, then you do not need to create any of these accounts or passwords.

# 5 Extracting Databases for Mobile Web Clients

This chapter describes database extraction. Extracting databases for Mobile Web Clients involves tasks for both the Siebel Remote server and the Mobile Web Clients. Table 14 shows these tasks. You must repeat each of these steps, for every Mobile Web Client.

Table 14. Tasks for Enabling Mobile Web Clients

| Task | Where Performed |
|---|---|
| Database Extraction for a Mobile Web Client on page 103 | Server |
| Initializing a Mobile Web Client Database on page 111 | Mobile Web Client |
| Enabling the Stand-Alone Synchronizer on page 113 | Mobile Web Client |
| Viewing Session Details on Mobile Web Clients on page 115 | Mobile Web Client |

## Database Extraction for a Mobile Web Client

The database extract process retrieves data visible to a specific mobile user from the server database. It retrieves data according to routing rules that define the level of access to information for each Mobile Web Client. It creates compressed files that contain data to be loaded into a local database when a Mobile Web Client initializes the database.

The database extraction process uses the new database template created by running the Generate New Database Template component. This template lets the database extraction process provide up-to-date database schemas to either new or existing mobile clients. It is strongly recommended that you distribute all database schema changes to all mobile clients, even if the changes are of types that are not visible to the mobile clients, such as changes to private dock objects. This is important because individual tables that are currently in a private dock object may become visible to mobile clients at a later time, and problems can occur if server and local database structures do not match.

Beginning with Siebel 7.5, the Database Extraction component includes an enhancement when dealing with a list of mobile users to extract. It will identify visible instances for all members of a list of nodes. Then it identifies the commonly visible instances and extracts the records only once for all these nodes. Then it extracts instances outside the common set for each node. This will help reduce the time for extracting large numbers of mobile users. The parameter that enables this is Optimal Mode. When Optimal Mode is set to TRUE, the time required for extraction can also be affected by parameters such as Nodes Per Group and Extract All Repository Files. For more information about Database Extraction parameters, see Table 15 on page 109.

**NOTE:** Multiple instances of the database extract components can be run simultaneously. In order to reduce contention, each task should be assigned to use a different temporary table called S_DOCK_INITM_n, where n is 1 to 48. Siebel Remote supports up to 100 such tasks. The Siebel Schema includes 48 S_DOCK_INITM tables. If you need additional temporary tables for the process to use, you can create them using Siebel Tools.

Before running a database extract for a client, you must make sure that your organization's reporting hierarchies are updated. From the application-level menu, choose Navigate > Site Map > Administration - Group > Positions to verify that the user you are about to extract has a valid position in your organization's hierarchy. The resulting information is used by the system's routing rules, and may affect the outcome of the database extract. For more information on positions, see *Applications Administration Guide*.

**CAUTION:** If a mobile user's position or routing model changes, reextract the Mobile Web Client's database to delete records that should no longer be visible to the user based on the user's new position.

Additionally, this improves Transaction Router performance because the reorganization process will generate many transactions on the server, which will create a backlog.

Administrators can start several instances of dbxtract and reduce the contention by using more than one table. The parameter is TS Table Number with a default of 1.

During the cleanup of dobjinst.dbf database tables, administrators can choose truncation or deletion of the tables. The parameter is Truncate TS Table with a default of FALSE.

**CAUTION:** If two instances of dbxtract use the same table, do *not* set TruncateTSTable to TRUE—one instance can truncate the records entered from another instance.

The Save Client Transactions feature prevents the loss of local transactions a mobile user may have entered into the local database that were not included in the server db extract. This feature is valid for normal re-extract of a mobile user's local database and will not work during a major upgrade process.

The default setting for the server parameter Save Client Transactions is TRUE. If the db extract for a Mobile Web Client occurs when this parameter is set to TRUE, Siebel Remote will invoke an action before applying the new db extract. Remote will extract transactions from the current local database that are not yet synchronized with the server and store them in the mobile user's inbox as DX files.

After the current local database is replaced with the new extract, Remote applies the DX files from the mobile user's inbox to the new local database. These include the saved transactions that were not synchronized with the server. These transactions are then sent to the server during the next synchronization session.

***To run a database extract for a Mobile Web Client***

1. From the application-level menu, choose Navigate > Site Map > Administration - Server Management > Jobs.
2. In the Jobs list, click New.
3. In the Component/Job field, select Database Extract from the picklist.
4. In the Requested Server field, enter the name of the server on which you want the Database Extract job to run.

   **NOTE:** After the job is completed, the read-only Execution Server field displays the name of the server that ran the job. For a Database Extract Job, this is the same as the Requested Server.
5. In the Job Parameters list, which is located below the Jobs list and the Job Detail form, click New and add the necessary parameters.

   The required parameter for Database Extract is Client Name.

   The value for the Client Name parameter is the name of the Mobile Web Client.
6. In the Jobs list, with the Database Extract record still selected, click Start.

   The Mobile Web Client database is extracted. This may take a few minutes.

**NOTE:** For limited-visibility objects, attaching many children to the same parent record can degrade the router performance and the database extract performance. The reason this may happen is that for each child the visibility for all other children must be checked. Whenever there are more than 10,000 child records attached to a parent (such as contacts attached to an account), the database extract performance and router performance need to be tested thoroughly. In case performance degradation is observed, it is necessary to limit the number of children per parent.

# Configuring the Remote Server for Local Database Password Encryption

You can enable RSA SHA-1 encryption, a one-way hashing algorithm, to be applied to the local database password before it is passed to the local database for authentication.

Encryption adds a layer of security by disconnecting the password stored in the local database from the unencrypted password entered by the user.

Encryption of the local database password requires configuration on the Remote server and on the Mobile Web Client.

- The Encrypt client Db password parameter (alias EncryptLocalDbPwd) on the Database Extract server component (alias DbXtract) determines whether the local database password is encrypted for the clients of given database extract. If set to TRUE, then RSA SHA-1 encryption is applied to the password the user enters, else it is not.

- Local database password encryption requires that the database security adapter on the Mobile Web Client be used for authentication. Parameters must be set in the .cfg file for the application on the Mobile Web Client for which password encryption is implemented.

  For information about configuring the Mobile Web Client to support local database password encryption, see "Configuring the Mobile Web Client for Local Database Password Encryption" on page 83.

For information about doing a database extract, see "Database Extraction for a Mobile Web Client" on page 103.

For a list of all database extract component parameters, see "Parameters of the Database Extract Component" on page 109.

# Database Extraction for Multiple Users

There are times when you need to extract a list of users or hundreds of users. The procedures in this section show how to do this.

### *To extract a list of users*

1. Create a list (or two) containing the mobile users in a text file.
2. Put one user name on each line.
3. Start dbxtract server task.

   Use @filename (including the path) as the value for the Client Name parameter.
4. If you have more than one list, start another dbxtract instance.

   Be sure to specify a different value for the TS Table Number parameter.

### *To extract hundreds of users*

1. Separate users into multiple lists (about 50 to 100 per list).
2. Start a dbxtract task for each list.
3. Note that mobile users should be extracted on the Applications server against which the users will be synchronized.

## Example of Extracting Databases for Multiple Users

This section describes the steps in the overall process of creating database extracts for multiple Mobile Web Clients. The purpose is to provide a broad perspective on the tasks involved.

It begins after installation of the Siebel application, to include registration of every user. The administrator receives the following business requirement: create database extracts for new mobile users AMARTIN, CCHENG, PSINGH and RMARLOW.

The requirement implies that to create a database extract you also need a database template. A database template is created only once. Additionally, in order to create database extracts for multiple users at the same time, you need to create a text file containing the user IDs of these mobile users. Finally, you submit the component request for creating the database extract.

The steps below outline the process for the example business requirement:

1. Log in to your Siebel eBusiness Application as System Administrator.
2. If you have not yet created a database template since installing, upgrading, adding a new server, or making any database changes, perform the following sub-steps to create a database template:
   a. From the application-level menu, choose Navigate > Site Map > Administration - Server Management > Jobs.
   b. Create a new job for Generate New Database, as described in "Generating a New Database Template" on page 67.
   c. Submit the job.

      The task of creating a database template takes a while. To check the progress of this task, from the application-level menu, choose Navigate > Site Map > Administration - Server Management > Components, select the Generate New Database component, and click the Tasks view tab to view status information.
3. In order to extract databases for multiple users, create a text file called ExtractUserList.txt containing the user IDs of the new mobile users, as described earlier in this topic.
4. From the application-level menu, choose Navigate > Site Map > Administration - Server Management > Jobs and create a job for Database Extract.
5. For the Database Extract job, create a Client Name job parameter, and assign it a value of the format @*AppServer*\ExtractUserList.txt, where *AppServer* is the UNC path to the directory where the file of user names is located.

Once the task of creating database extract is completed, go to the directory d:\sea14010\siebsrvr\docking to see subdirectories for AMARTIN, CCHENG, PSINGH, and RMARLOW. To confirm successful extraction, check for compressed files in the outbox subdirectories of each of these client directories.

# Performing a Database Extract to a CD Directory

By default, a database extract stores the compressed file on the Siebel Remote server. Users log on to the Siebel Remote server to download data to initialize their local databases. However, a database extract can also store the compressed file in a directory that you specify in the database extract parameters. This allows you to copy the compressed database file from the specified directory and then make an image of the files on a CD-ROM or other media device, which you can then distribute to users. Mobile users can initialize their local databases directly from the CD-ROM rather than having to download to the Siebel Remote server.

**NOTE:** The user still needs a network connection for this method. Remote clients still need to synchronize before retrieving a database extract from the CD-ROM.

Beginning with Siebel release 7.7, a new parameter in the Mobile Web Client's configuration (.cfg) file specifies the directory that will be used to store files to download for the initialization of the client's local database. You may need to change the value of this parameter if you want a Mobile Web Client to initialize a local database from a CD-ROM or a directory other than the default location on the Siebel Remote Server.

The new parameter is DbinitLocalSource, located in the [Local] section of the client's .cfg file. In previous releases, the FileSystem parameter was used for this purpose. Beginning with release 7.7, any FileSystem parameter in the [Local] section of a Mobile Web Client's .cfg file will be ignored.

**NOTE:** It is possible to extract groups of users simultaneously. Therefore, it is possible to use one CD for more than one user. The names of the files contain the client_names.

### *To perform a database extract to a CD directory*

1. Complete Step 1 through Step 4 of "To run a database extract for a Mobile Web Client" on page 105.
2. In the Component Request Parameters list, click New and add the necessary parameters.
   a. The required parameter for Database Extract is Client Name.

      The value for the Client Name parameter is the name of the Mobile Web Client.

   b. The parameter for extracting to a CD is CD Directory. Specify the name of the directory to which you want to extract files

      For example, type D:\temp-extract\username.

   c. Modify the values of other parameters as necessary.

      Table 15 on page 109 lists the parameters and default values for the database extract component.
3. In the Component Requests form, click the menu button and then click Submit request.

   The Mobile Web Client database is extracted to the directory specified in the CD directory parameter. You can make an image of these files on a CD-ROM or other media device.

# Parameters of the Database Extract Component

Table 15 lists the parameters and default values for the database extract component.

Table 15. Database Extract Parameters

| Name | Alias | Required /Optional | Default Value and Usage Notes |
|---|---|---|---|
| CD directory | CDDir | Optional | Name of a directory to which snapshot files are copied for use by the CD-ROM initialization method. |
| Client Name | Client | Required | Name of the Mobile Web Client for which you are performing a database extract. This corresponds to the Mobile Client Name field in the Mobile Clients view.[1] |
| Database Init Method | InitMethod | Optional | Method for creating the Siebel Remote database. Default is SQL Anywhere. |
| Database template file name | DbTmplFile | Optional | Name of the SQL Anywhere Database Template file. Default is sse_utf8.dbf.<br>This entry should match the entry you make for the SQL Anywhere Database parameter in the Generate New Database component. If you use the encrypted local database template, enter sse_encr.dbf in both places. |
| Encrypt client Db password | EncryptLocalDb Pwd | Optional | If TRUE, the local database password is encrypted using RSA SHA-1 encryption before being passed to the local database for authentication. Default value is FALSE. |
| Extract all Repository Tables | ExtractRepos | Optional | Specifies that the repository tables will be included in a db extract. Values are TRUE and FALSE. |
| Last Extract Date | ExtractSince | Optional | When specified, Database Extract extracts Mobile Web Clients only if they have not been extracted since this date-time value. The value must use the format YYYY-MM-DD HH:MM:SS. |
| Last Sync Date | SyncSince | Optional | When specified, Database Extract extracts Mobile Web Clients only if they have not been synchronized since this data-time value. The value must use the format YYYY-MM-DD HH:MM:SS. |

Table 15. Database Extract Parameters

| Name | Alias | Required /Optional | Default Value and Usage Notes |
|---|---|---|---|
| Maximum data file size | DatFileSize | Optional | Sets the maximum size of a data file in megabytes. Minimum size is 1. Maximum size is 1000. Default is 500. |
| Message Language Code | Language | Optional | Extract messages for this language. Default is ENU. |
| Move Siebel Remote client | Move | Optional | Specifies if Database Extract should set the Mobile Web Client's Siebel Server name to the local Siebel Remote server if the Mobile Web Client is currently registered on another Siebel Remote server. The default is FALSE.[2] |
| Nodes Per Group: | NodesPerGroup | Optional | Number of users in one group to be extracted together when OptMode is TRUE. Default is 35. |
| Optimal Mode | OptMode | Optional | Specifies whether to use the optimal mode to extract a group of users. Default is FALSE. |
| Save Client Transactions | SaveTxns | Optional | Save pending client transactions during database initialization. Default is TRUE. This feature will not work during the upgrade process.<br>Save Client Transactions is valid *only* for Mobile Web Clients. |
| Specify the mobile client vers | ClientVersion | Optional | Specifies the client software version. This is important during upgrades. Default is 2000 for v7.x. |
| Truncate TS Table | TruncateTSTable | | Specifies whether the S_DOCK_INITM_n table should be truncated instead of deleted after the database extract task completes. Used in the cleanup phase. Default is FALSE.<br>When running the database extract (dbxtract) component with the Truncate TS Table parameter set to TRUE, the user ID used for running the server component must have administrative privileges in the database server. |

Table 15. Database Extract Parameters

| Name | Alias | Required /Optional | Default Value and Usage Notes |
|---|---|---|---|
| TS Block Size | TSBlockSize | Optional | Block size for dobjinst.dbf database in bytes. The default is 0. The block size is automatically calculated depending on the total number of rows visible to the Mobile Web Client. |
| TS Cache Size | TSCacheSize | Optional | Cache size for dobjinst.dbf database in kilobytes. The default is 2048 kilobytes. |
| TS Table Number | TSTableNum | | Number of dobjinst.dbf database tables available for dbxtract: 1 to 48. Default is 1. The ending number of the S_DOCK_INITIM_n table. |

1. You can specify a list of node names delimited by commas. If the first character is @, this parameter specifies the name of a file that contains a list of nodes. Client names in the file must be separated by one of the following delimiters: new line, comma, space, tab, period, or semicolon. You can specify wildcards, using * to match zero or more characters and ? to match a single character.

2. This option applies to the Siebel Remote clients, including Mobile Web Clients and regional nodes. Database Extract returns an error if you try to extract a Mobile Web Client that was previously extracted on a different server. You can move the client to the current Siebel Server by setting the Move parameter to TRUE. You can also move a regional node to the current Siebel Server by using the Move parameter. The Transaction Router, Transaction Merger, and DB Extract processes on the old server stop servicing the Mobile Web Client or regional node in the next iteration.

You may get an error message if the target node is used by another Siebel Remote server process. For example, if another Siebel Remote server process were accessing the inbox or outbox directory for sjones, you would receive the following error message:

```
Target node “sjones” is currently in use by another server process. Try again later.
```

In this case, you should try to rerun a database extract in a few minutes when the file may be available and unlocked.

**NOTE:** After you have performed a database extract for a Mobile Web Client, the client database must be initialized before any data exchange between the client and the server can occur. This includes the uploading of any client database changes.

# Initializing a Mobile Web Client Database

The volume of information that must be downloaded from the Siebel Remote server to initialize a Mobile Web Client’s database is usually substantial. You should establish a LAN (rather than a modem or WAN) connection between the server and the Mobile Web Client for this process.

Alternatively, the local database can be initialized from a CD-ROM or other media if compressed files have been copied into the folder specified as FileSystem parameter. For more information, see "Performing a Database Extract to a CD Directory" on page 107.

**NOTE:** To initialize a Mobile Web Client database, the TableOwner parameter in the CFG file must be set to Siebel (the default).

### *To initialize the Mobile Web Client database using the GUI*

1. Establish a connection between the Siebel Remote server and the Mobile Web Client.
2. In the Mobile Web Client's Siebel program group, click the Siebel Remote icon.

   **NOTE:** Verify that the icon is pointing to the appropriate CFG file. The default is siebel.cfg.
3. In the Siebel Remote Parameters dialog box, enter the appropriate information.

   In the Client Name field, enter the registered Siebel client name.

   In the User Name field, enter the login name of the mobile user.
4. Enter the password.

   The password must match the authenticator password.
5. Click Continue.
6. Monitor the process for errors by clicking the opposing arrows in the lower right corner of the screen.

### *To initialize the Mobile Web Client database from the command line*

- The Mobile Web Client database can also be initialized from the command line using the stand-alone synchronizer (siebsync.exe). For information on how to use the stand-alone synchronizer, see "Enabling the Stand-Alone Synchronizer" on page 113.

### *To initialize the Mobile Web Client database during login*

- Another way to initialize the Mobile Web Client is to log in to the local database when starting the application. When your Siebel application cannot find a local database, it will attempt to initialize the local database following the procedures described above.

# Enabling the Stand-Alone Synchronizer

A mobile user can control the synchronization start-up process by running the Siebel Remote client in a third-party scheduling program. Table 16 shows the command-line options for the Siebel Remote client.

**NOTE:** If a local connection to the client is open and the user uses the Siebel Remote icon to synchronize, synchronization works successfully. However, if a new database extract exists for that user, the synchronization fails, because it is trying to overwrite the sse_data.dbf file, which is already in use. Therefore, the stand-alone Siebel Remote client should only be used when the Mobile Web Client is not in use.

Table 16. Stand-Alone Synchronizer Command-Line Options

| Option | Definition | Required /Optional | Default Value and Usage Notes |
|---|---|---|---|
| /? | Help | Optional | Provides online help for usage. |
| /a | Autostart mode | Optional | Available modes are Y or N. |
| /ApplyTxns | Apply transactions during this synchronization | Optional | Determines whether or not transactions will be applied to the local database during the current synchronization operation. Available modes are Y or N. Defaults to Y. A value of N directs transactions to be downloaded and uploaded, but not applied. |
| /c | Configuration file | Required | The default is siebel.cfg. |
| /comm | Communication parameters | Optional | Used for modem connections. |
| /d | Data source | Optional | The default is local. |
| /help | Help | Optional | Provides online help for usage. |
| /i | Iterations | Optional | Sets the number of iterations. |
| /l | Language | Optional | Language to use for the docking session. |
| /n | Client name | Required | Name of Mobile Web Client. The value must be entered using uppercase letters. |
| /p | User password | Required | Database connection password. The value must be entered using uppercase letters. |
| /p2 | Confirmation password | Required | Required when initializing. |

Table 16. Stand-Alone Synchronizer Command-Line Options

| Option | Definition | Required /Optional | Default Value and Usage Notes |
|---|---|---|---|
| /ParallelApply | Apply transactions in parallel with data transfers | Optional | Determines whether transactions will be applied to local database while other transactions are still being transferred between client and server, or whether all transactions will be applied after data transfer is complete. Available modes are Y or N. Defaults to Y. This option is ignored if the ApplyTxns option is set to N. |
| /RecvAutoUpdFiles | Receive auto updates to files | Optional | Determines whether Mobile Web Client will download updates to requested files. Available modes are Y or N. |
| /RecvFiles | Receive files | Optional | Determines whether Mobile Web Client will download files from server. Available modes are Y or N. |
| /RecvPubFiles | Receive published files | Optional | Determines whether Mobile Web Client will download published files. Available modes are Y or N. |
| /RecvTxns | Receive transactions | Optional | Determines whether Mobile Web Client will download transactions from server. Available modes are Y or N. |
| /SendTxns | Send transactions | Optional | Determines whether Mobile Web Client will send transactions to server. Available modes are Y or N. |
| /sleep | Sleep time | Optional | Number of seconds to sleep between iterations. |
| /u | User name | Required | Database connection user name. The value must be entered using uppercase letters. |
| /v | Verbose mode | Optional | Available modes are Y or N.[1] |

1. You may want to run the Stand-Alone Synchronizer unattended by setting the verbose mode to N. After synchronization, the Stand-Alone Synchronizer returns 0 if the process succeeded, or a nonzero Siebel error code if it failed.

Using certain command line options, the process of uploading and downloading transactions can now be separated from the process of applying downloaded transactions to the local database. The purpose of this option is to let users choose the most convenient timing for these two portions of a synchronization, rather than requiring them to be performed together. The following procedures briefly describe how to transfer and apply transactions separately.

***To download and upload transactions without applying them – command line***

- To download and upload transactions without applying them to the Mobile Web Client's local database, use the following command-line syntax:

  ```
  siebsync.exe /n ClientName /u UserName /p Password /p2 Password /ApplyTxns N
  ```

***To apply previously downloaded transactions – command line***

- To apply transactions that were downloaded to the Mobile Web Client during an earlier synchronization option, use the following command-line syntax:

  ```
  siebsync.exe /n ClientName /u UserName /p Password /p2 Password /SendTxns N /RecvTxns N /RecvFiles N
  ```

# Viewing Session Details on Mobile Web Clients

The Remote Status view displays information about database updates based upon the value of the system preference MRG:User Friendly Notification. For more details about system preferences that affect the way Remote manages database changes, see "Setting Siebel Remote System Preferences" on page 48. There are three applets in this view: Session Summary, Session Actions and Session Actions Details. The first two applets typically appear, while display of the Session Actions Details applet depends upon the activities of the synchronization session.

The previous synchronization details are archived on the local database and not stored on the server database. On [Local], the previous synchronization details could be found in the Site Map > User Preferences > Remote Status view. On [Server], only the latest synchronization details are stored. These can be referenced from views in the Administration - Siebel Remote screen.

The lists below describe the fields in the three applets in the Remote Status View.

- Session Summary. Lists every synchronization session that occurred since the last local database initialization. There are three fields:
  - Session Date. Timestamp of the beginning of the synchronization session.
  - Session Status. Result of the synchronization session.
  - Session Result Summary. Key summary information of the synchronization session.
- Session Actions. Provides detailed information of the major action of the synchronization session. Usually, there are five major actions. There are two fields in this applet.
  - Item Name. Name of major action.
  - Results Summary. Summary result of the particular action.
- Session Actions Details. Provides more detail depending upon the activities of the synchronization session.
  - Item Type. Type of object, such as Account or Opportunity.
  - Item Name. The actual field.
  - Action. The kind of conflict or action.

- Updated By. Self-explanatory.
- Date Updated. Self-explanatory.
- Item Details. Detailed information as to what was done.

# 6 Configuring and Using Mobile Web Client

This chapter provides supplementary instructions regarding synchronization and Siebel QuickStart for Mobile Web Client users:



For additional information about configuration tasks for Siebel Mobile Web Clients, many of which can be performed by either users or administrators, see Chapter 4, "Setting Up Mobile Web Clients."

# User Synchronization Options for Mobile Web Client

This section describes some user synchronization options supported by Siebel Remote and Siebel Mobile Web Client in the Siebel Remote synchronization dialog box and in the Synchronization Parameters section of the User Preferences > DB Synchronization view.

## Siebel Remote Synchronization Dialog Box

The Siebel Remote synchronization dialog box shown in Figure 9 contains a list of synchronization actions. This dialog box appears when you choose File > Synchronize Database in a Siebel application or when you run the stand-alone synchronizer.

Depending on your database options, the list may appear different from that in the example. For instance, if you do not have a local database, the dialog box may contain only the Connect to Server action.

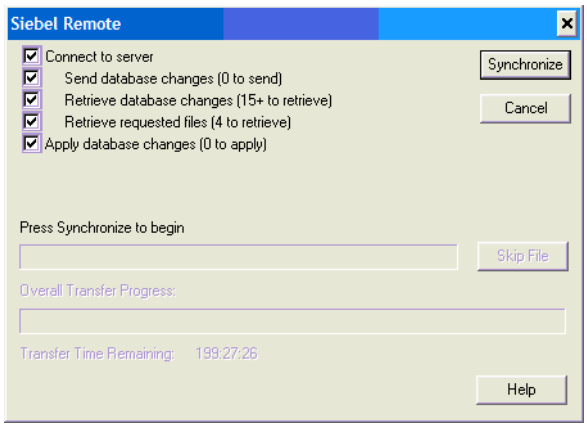


Figure 9. Siebel Remote Synchronization Dialog Box

### Synchronization Actions

This section describes the available synchronization actions in the Siebel Remote synchronization dialog box.

- **Connect to server.** Connects to the server, dials the phone number if necessary, and performs version, schema, and database initialization checking. If a Local Database Initialization (dbinit) or upgrade is pending, this action downloads and applies the pending Local Database Initialization or upgrade.

  This action must run successfully for the following three actions to be invoked: Send database changes, Retrieve database changes, and Retrieve requested files.

- **Send database changes.** Sends the local transactions to the server by sending one or more DX files and associated attachment files to the server. This action displays the number of transactions that still need to be sent.

- **Retrieve database changes.** Retrieves transactions from the server by retrieving one or more DX transaction files and associated attachment files from the server. These files are generated by the Transaction Router on the Siebel Remote Server. This action displays the number of transactions that still need to be retrieved.

- **Retrieve requested files.** Retrieves user-requested attachment files from the server. This action displays the number of files that still need to be retrieved.

- **Apply database changes.** Applies database changes retrieved from the server to the local database. This action displays the number of transactions that still need to be applied.

### Synchronization Action States

In the Siebel Remote synchronization dialog box, you can enable or disable each action by checking the check box to the left of the action. Explanations of the possible check box states are shown below:

**Enabled.** This action will run when its turn comes.

**Disabled.** This action will not run when its turn comes.

**Not runnable.** This action cannot run because an action it depends upon did not (or will not) run.

**Finished.** This action is finished.

**Skipped.** This action was skipped.

While an action is running, you can click the check box for the action to disable it before the action has finished. If an action is running when you disable it, it stops as soon as possible, and synchronization continues with the next action.

The action status check boxes persist throughout the session. If you start synchronization with the option Apply database changes disabled, it remains disabled throughout synchronization. If you restart Siebel applications, the action status check boxes return to their default state (enabled).

### Item Status Field

The word *pausing* appears in the Item Status field. Actions are made up of one or more items and may be hidden. The items (and sub-items) that appear in this field update the synchronization status for the user. Below the Item Status field is the item progress indicator, which indicates the remaining processing time for the item.

The progress indicator provides reasonably accurate time estimates for long-running items, such as transferring files or merging transactions, but less accurate estimates for short-running items, such as connecting to the server or database.

### Skip File Button

The Skip File button is enabled whenever synchronization is performing an item or sub-item that is not critical to the current action. Retrieving an optional attachment file is the only action that can be skipped. Clicking this button skips the current item or sub-item, and the current action continues. Double-clicking this button generates a prompt and skips all remaining optional items for this action.

### Overall Transfer Progress Panel

This panel appears when communications with the server are active. The progress indicator displays the progress toward completing the actions that require the communications line.

### Transfer Time Remaining Field

This field contains the estimated time for completing the actions.

**NOTE:** The estimated time may be inaccurate when large attachment files are transferred.

### Synchronize/Stop Command Button

The Synchronize button starts synchronization. You can click this button even while synchronization is still performing hidden actions, such as connecting to the local database and extracting local database changes. When you click this button, the synchronization dialog box is hidden (unless SHIFT is held down), and the label on the button changes to Stop.

Clicking the button again (Stop) stops synchronization if it is still running. When synchronization finishes, the button changes to OK. Clicking the button again hides the synchronization dialog box.

If the dialog box is hidden while synchronization is running, invoking synchronization redisplays the dialog box. If synchronization is finished, invoking synchronization resets and redisplays the dialog box.

### Cancel/Hide Button

Clicking the Synchronize button automatically hides the synchronization dialog box, unless you hold down SHIFT when you click the button. If the dialog box is hidden, you can redisplay it by clicking the corresponding icon in the system tray.

If the synchronization dialog box is displayed:

- If synchronization has not started, clicking Cancel closes the synchronization dialog box and aborts synchronization.
- If synchronization has started, the label on this button changes to Hide. Clicking this button hides the synchronization dialog box.

## Synchronization Parameters in User Preferences

Navigating to User Preferences > DB Synchronization displays the following Synchronization Parameters settings. The two File options control which files to synchronize. File options can be changed while synchronization is running.

- **Connection.** The Connection box allows users to select a dial-up networking connection. Leaving the box blank causes synchronization to use an existing network connection. Changes to the dial-up networking options takes effect the next time synchronization makes a connection to the server.
- **Retrieve Published Files.** When this check box is selected, synchronization retrieves all newly published files.
- **Retrieve Auto Update Files.** If this check box is selected, synchronization retrieves updates to local files marked for auto-update. Auto-update files that are not local or have not been requested are not retrieved.

  The auto-update flag is local to each node, so users can choose to update files automatically, without affecting other users. If you do not choose the Retrieve Auto Update Files option during this synchronization session, then synchronization marks the files that need to be updated. Synchronization then makes all auto-update files current during the next session, when Retrieve Auto Update Files is enabled.

  When the Mobile Web Client receives a transaction on a file table, synchronization decides whether to retrieve the file locally. The default is not to download files to the Mobile Web Client unless the file is requested. File screens generally have a request field that users can select to download the file during the next docking session (if the Retrieve Requested Files action is enabled). Files are also uploaded to the server by docking.

  If you change a file, you must upload the new file with the transaction to make the corporate database consistent. These files are stored near the corporate database on a file server. Name files carefully to avoid conflicts, such as two users attempting to modify a file at the same time.
- **Auto-Start Siebel Remote.** If this check box is selected, Siebel Remote begins synchronization as soon as you invoke synchronization. The Siebel Remote synchronization dialog box is not displayed. If you want to display the dialog box throughout your session, press SHIFT when you click Start.

# Siebel Mobile Web Client and Siebel QuickStart

Siebel QuickStart is an agent that is preloaded on a mobile user's machine at startup and reduces the time required to launch the Siebel Mobile Web Client.

**NOTE:** Siebel QuickStart can be used with the Siebel Mobile Web Client only, when connecting to the local database or Sample Database. It does not apply to Siebel Dedicated Web Client or Siebel Web Client.

Siebel QuickStart is enabled and loaded for the first time from the Siebel application login screen.

Siebel QuickStart applies to subsequent instances of the same type of Siebel application session—running the same application as the same user and with the same startup options. The speed increase provided by QuickStart does not take effect on the initial Siebel login. Subsequent logins of the same type of Siebel application session benefit from QuickStart, until the user disables QuickStart.

Siebel QuickStart stores the encrypted Siebel user name and password in the siebel.local.client cookie. For additional information about using cookies with Siebel applications, see *Security Guide for Siebel eBusiness Applications*.

**NOTE:** Users should disable Siebel QuickStart for existing application sessions before enabling it for another application session.

## Enabling and Disabling Siebel QuickStart

### *To enable Siebel QuickStart*

1. Start a Siebel application with the Mobile Web Client. For example, double-click the shortcut for Siebel Call Center for your applicable language.
2. In the Siebel login screen, check the Enable Siebel QuickStart check box.

   The QuickStart agent is loaded into memory for the rest of the Microsoft Windows session, and is loaded again when the user logs into Windows subsequently, unless the user disables it.

### *To disable Siebel QuickStart from the login screen*

1. Start a Siebel application with the Mobile Web Client. For example, double-click the shortcut for Siebel Call Center for your applicable language.
2. In the Siebel login screen, clear the Enable Siebel QuickStart check box, if it is checked.

   The QuickStart agent is not loaded for this Siebel application session, and does not load subsequently unless it is explicitly specified.

***To disable Siebel QuickStart from the system tray***

- Right-click the Siebel icon in the system tray and select Disable On Startup.

  If the Siebel application is not running, the QuickStart agent exits. The agent does not load subsequently unless it is explicitly specified.

  If the Siebel application is running, the QuickStart agent stays loaded until you log out of the Siebel application session. The agent does not load subsequently unless it is explicitly specified.

## Options for Siebel Icon in the System Tray

When the Siebel application is running *or* the QuickStart agent is loaded, an icon appears in the system tray. Right-clicking this icon displays several choices:

- **Exit.** Exits the QuickStart agent for the current Windows session. This option is available only when the QuickStart agent is loaded and the Siebel application is *not* running. If QuickStart is enabled, the agent loads again the next time the user starts the Siebel application the same way, or starts Windows.
- **Disable On Startup.** Disables Siebel QuickStart the next time the user starts the Siebel application or starts Windows. Also exits the QuickStart agent, in the manner described in the previous section. This option is available only when the QuickStart agent is loaded, whether or not the Siebel application is running.
- **About.** Displays information about Siebel applications. This option is available when the Siebel application or the QuickStart agent are running, or both.
- **Help.** Displays *Siebel Online Help*. This option is available when the Siebel application or the QuickStart agent are running, or both.

## Using View Precaching with Siebel QuickStart

When the QuickStart agent is loaded, views specified using the parameters in the `[Preload]` section of the configuration file, such as siebel.cfg, are preloaded (precached) during startup of the Siebel application. In subsequent application sessions, navigating to a precached view is faster.

In the [Preload] section, specify the names of the views to be precached as the values for configuration parameters named View1, View2, and so on.

For more information about specifying configuration file parameters, see *Siebel System Administration Guide*.

## Specifying QuickStart Availability with Siebel Packager

By default, the siebel.exe program has Siebel QuickStart available. However, administrators may choose to package client installations with QuickStart unavailable—so the Enable Siebel QuickStart check box does not appear on the Siebel login screen and thus cannot be enabled by end users.

When preparing model Siebel client installations to include in packages you create using the Siebel Packager utility, you can disable Siebel QuickStart. To do this, rename or remove the siebel.exe program and then change the name of siebel1.exe to siebel.exe.

For more information on using the Packager utility, see *Going Live with Siebel eBusiness Applications*.

## QuickStart and AutoStopDB Configuration Parameter

When you are using Siebel QuickStart, you may also decide to set the configuration parameter `AutoStopDB` to FALSE for the local database. Both of these features have a similar performance benefit when the Siebel application is started and exited multiple times within the same Windows session.

For more information about the AutoStopDB parameter, see *Siebel System Administration Guide*.

# 7 Administering Siebel Remote

This chapter describes administrative tasks for operations and maintenance of Mobile Web Clients. It also addresses routing rules, dock objects, and dealing with failures and recovery. Table 17 addresses these tasks.

Table 17. Tasks for Administering Siebel Remote

| Task | Where Performed | When Performed |
|---|---|---|
| Starting and Stopping Siebel Remote Server Components on page 126 | Siebel Server Manager | Database start-up, Siebel Server start-up |
| Monitoring Siebel Remote Operations on page 128 | Siebel Server Manager | Daily |
| Data Synchronization on page 139 | Client Machine | Daily or more often |
| Sending Messages to Mobile Users on page 141 | Siebel Remote server | As needed |
| Refreshing a Client Database on page 141 | Siebel Server Manager | As needed |
| Deactivating and Reactivating a Mobile Web Client on page 142 | Siebel Client connected to Siebel Server | As needed |
| Deleting a Mobile Web Client on page 143 | Siebel Client connected to Siebel Server | As needed |
| Changing Routing Models on page 143 | Siebel Client connected to Siebel Server | As needed |
| Adding New Mobile Users on page 144 | Siebel Client connected to Siebel Server | As needed |
| How to Set Client-Side Logging on page 144 | Various | As needed |
| Event Tracing for Locking on page 147 | Siebel Server Manager | As needed |
| Handling Failure and Recovery on page 148 | Various | As needed |

# Starting and Stopping Siebel Remote Server Components

You can use either Administration views or the Siebel Server Manager to start and stop any Siebel Remote server components.

**NOTE:** Event logs provide helpful information for diagnosing problems that may arise. To enable event logging for Remote components, see *Siebel System Administration Guide*. For example, to receive useful information for the dbxtract component, enter the following in the command-line interface for srvrmgr:

```
change evtloglvl GenericLog=5, Trace=5, SQL=5, EVENT_GENERIC_PROFILE=5 for comp Dbxtract

start task for comp Dbxtract server <server_name> with Client=<client_name>, SQLFlags=2,
TraceFlags=7
```

The following procedures describe how to use srvrmgr to enable or disable the Siebel Remote component group, and how to disable individual components within the enabled group.

### *To enable the Remote component group*

1. In the command-line interface, change to the Siebel Server bin subdirectory.
2. Enter the following command to invoke the Line Mode Server Manager:

   ```
   srvrmgr /e <enterprise server> /g <gatewayserver>  /u <username> /p <password>
   ```

3. From the Server Manager command line, enter:

   ```
   enable compgrp remote
   ```

   This will enable all Siebel Remote components: Synchronization Manager, Transaction Processor, Transaction Router, Transaction Merger, Replication Agent, Database Extract, Parallel Database Extract, and Generate New Database.

4. From the GUI, synchronize the components by navigating to Administration - Server Configuration > Enterprises> Synchronize.
5. Click Synchronize.

   For more information about synchronizing components, see *Siebel System Administration Guide*.

### *To disable the Remote component group*

- From the Server Manager command line, enter:

  ```
  disable compgrp remote
  ```

### *To enable Remote component group for a specific server*

- From the Server Manager command line, enter:

```
enable compgrp remote for server <server_name>
```

This will enable Remote only on certain Application servers in the enterprise, rather than on all of them.

***To disable individual components of the Remote component group***

- From the Server Manager command line, enter:

  ```
  disable compdef <component definition>
  ```

  Component definition will be `synchmgr`, `txnproc`, `txnroute`, `txnmerge`, `repagent`, `dbxtract`, `pdbxtract`, or `gennewdb`, depending upon the situation.

For additional information on starting and stopping server components using the Server Manager, see *Siebel System Administration Guide*.

# Transaction Processor

- You can run only one transaction processor on each Siebel Remote server.
- At startup, the transaction processor verifies that another transaction processor is not running on the same Siebel Remote server.

  **NOTE:** The optimal number of Remote Servers, and therefore the number of Transaction Processors, will depend upon the number of mobile users, volume of transactions generated by the system, and other aspects of the system. For assistance with hardware planning and sizing, contact Siebel Expert Services.

# Transaction Router

The Transaction Router takes transactions from DOCKING\TXNPROC and constructs DX files. It then sends these DX files to the outbox corresponding to the appropriate mobile user.

## Multiple Transaction Router Processes

You can start multiple Transaction Router processes on the Siebel Remote server to increase the throughput of transactions to mobile user outboxes.

- Each Transaction Router process can simultaneously route transactions from the database server to a different Mobile Web Client. For example, if you start four Transaction Router processes, the four Transaction Router processes can route transactions to four different Mobile Web Clients simultaneously.
- Using multiple Transaction Router processes reduces the total time needed to route transactions to Mobile Web Clients.

**NOTE:** The optimal number of Transaction Routers will depend upon the number of mobile users, volume of transactions generated by the system, hardware configuration and other components of the system.

You can start as many parallel Transaction Router processes as the database server and Siebel Remote server can support:

- You should monitor the database server and Siebel Remote server to make sure the parallel Transaction Router processes do not overload the systems.
- Plan to start with two Transaction Router processes and increase the number as needed.
- You will typically run multiple Transaction Router processes per server.

**CAUTION:** Do not run a Transaction Router and Ddlsync at the same time. It can cause Transaction Router to shut down.

Also, do not run visutl on the Siebel Server while Transaction Routers are running.

### Routing Transactions from the Database Server

Only one Transaction Router process can route transactions from the database server to a Mobile Web Client at any one time:

- The Transaction Router locks the Mobile Web Client's outbox directory to prevent other Transaction Router processes from routing transactions to the same Mobile Web Client.
- Other Transaction Router processes skip the Mobile Web Client if another Transaction Router process is already processing this client.
- After the Transaction Router finishes routing transactions to the Mobile Web Client, the Transaction Router releases the lock and searches for another unlocked Mobile Web Client to process.

## Transaction Merger

Transaction Merger pulls DX files from the inbox in the server Docking Directory and applies these to the server. It also resolves conflicts.

- You must run at least one transaction merger on each Siebel Remote server.
- You can run multiple transaction mergers on the same Siebel Remote server.

**NOTE:** The optimal number of Transaction Mergers will depend upon the number of mobile users, volume of transactions generated by the mobile users, hardware configuration, and other components of the system.

# Monitoring Siebel Remote Operations

This section describes how to monitor the status of Mobile Web Clients and Siebel Remote operations on the server. The information is divided into the following sections:

- "About Automatic Notification of Critical Conditions" on page 129
- "Monitoring Mobile Web Client Status" on page 129
- "Checking Siebel Remote Transaction Backlogs" on page 136

■ "Monitoring Transaction Logs" on page 137

■ "Monitoring Siebel Remote Server Operation Status" on page 139

# About Automatic Notification of Critical Conditions

During processing, certain critical conditions can occur on the Siebel Remote server related to the transaction router, transaction processor, or the transaction merger component. You can configure your Siebel server implementation so that, if one of these conditions does occur, a notification alert is generated and an email message is sent to you automatically. The purpose of this notification alert is to reduce the time it takes you to detect and respond to these problematic conditions.

The following conditions can generate email notification messages:

- Components fail to start
- Standard system failure conditions, such as the component failing or exiting abnormally
- Running out of disk space on Siebel Server, which prevents event logs from being written
- The Siebel system reaches defined critical throughput thresholds (numbers of transactions in a specified time period called the Monitor Data Calculate Period) for Siebel Remote in areas such as the following:
  - Visible-event transactions (transactions that affect the amount of data that can be seen and who can see it, such as addition of an Account)
  - Position rule transactions (certain visible-event transactions that affect who can see specified data, such as assigning a user to an Account)
  - All types of transactions combined
  - Routing throughput of the Transaction Router

The email notification alert feature must be enabled by defining the critical throughput thresholds and related alert notification settings. For information on enabling email notification, see *System Administration Guide*.

# Monitoring Mobile Web Client Status

You can use views in the Administration - Siebel Remote screen to monitor the status and progress of Mobile Web Clients. This topic discusses the following views:

- Client Status View
- Client Diagnostics View
- Synchronization Sessions View
- Upload Statistics View
- Download Statistics View

## Using the Client Status View

The Client Status view displays the current status of Mobile Web Clients, including:

- The last time each Mobile Web Client synchronized
- The product versions on each Mobile Web Client
- The last time each Mobile Web Client was extracted and initialized
- Free Disk (Bytes)

Siebel Remote updates the data in this screen at the start and end of each synchronization session. However, the data in this screen is not updated in the following two cases:

- During an initialization

  The Free Disk (Bytes) field will also be reset to zero during dbxtract.

- After applying changes to the client database, *if* those changes are applied after data transfers are complete

In both cases, the actions occur after the Mobile Web Client disconnects from the server. Therefore, the data in the status screen is not updated until the next synchronization session. At that time, users can check this information to help avoid running out of disk space on their laptops.

When using Siebel Anywhere, the following occurs: the Product Versions field is not populated until the Mobile Web Client upgrades to another version. Once the upgrade is complete, the Product Versions field is updated with the version information after the next synchronization session.

### *To use the Client Status view*

1. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote.
2. In the link bar, select Client Status.
3. In the Mobile Clients/Replication Servers list, select the Mobile Web Client that you want to monitor.

   The fields in the Siebel Remote Client Status form provide details about the status.

| Field | Description |
|---|---|
| **Extracted on Server** | |
| Last Sessions | Time when the extract for this client was done |
| Seconds | Time it takes to extract the client |
| File name | First file name for the extracted records |
| Rows extracted | Row count of extracted records |
| Snapshot (Kbytes) | Total size in kilobytes for extracted records |
| # of Files in File System | Number of files in file system for this client |
| File System (Kbytes) | Size of files in file system, in kilobytes |
| Max Transaction | Maximum transaction ID when this client was extracted |

| Field | Description |
|---|---|
| **Initialized on Client** | |
| Last Session | Time when the local database initialization was done |
| Seconds | Time it takes to initialize the local database |
| Free Disk (Kbytes) | Free disk space available on the mobile client's laptop drive |
| Database (Kbytes) | The size in kilobytes of the local database |
| # of Files in File System | Number of files in local file system |
| File System (Kbytes) | Total size in kilobytes of files in local file system |
| Product Version | System information for the machine where local database resides |
| **Current status** | |
| Last Session | Time of the last synchronization session |
| Seconds | Duration of the last synchronization session |
| Free Disk (Kbytes) | Free disk space available on the Mobile Web Client's laptop drive |
| Database (Kbytes) | Current size of the local database |
| # of Files in File System | Number of files in the local file system |
| File System (Kbytes) | Current size in kilobytes of files in the local file system |
| Total Sessions | Total number of times this client has synchronized so far |
| Product Version | Current system information for the client machine |

## Using the Client Diagnostics View

The Client Status Diagnostics view provides data routing information about the processors, mobile clients, and regional nodes. Table 18 lists and describes the fields on the Siebel Remote Client Diagnostics form.

Table 18. Fields on Siebel Remote Client Diagnostics Form

| Field | Description |
|---|---|
| Name | Node name |
| Type | INIT, ROUTE, RECEIVE, MERGE, CLEAN, SESSION |
| Local | Flag to show whether it is for current or remote databases |
| Created | Creation time for the record |
| Last updated | Last update time for the record |
| Last file | Last file number processed for the node |
| Last transaction id | Last transaction number processed for the node |

Table 18. Fields on Siebel Remote Client Diagnostics Form

| Field | Description |
|---|---|
| Last trxn duration (sec) | Time it took to process transaction in the latest session |
| Total duration (sec) | Total Time it took to process transaction for the node |
| Last trxn size (bytes) | Size of transactions processed for the node during last session |
| Total (bytes) | Total size of transactions processed for the node |
| Last trxn operations | Number of operations processed for the node during last session |
| Total operations | Total number of operations processed for the node |
| Last transactions | Number of transactions processed for the node during last session |
| Total transactions | Total number of transactions processed for the node |
| Total attachment (bytes) | Total size of file attachments processed for the node |
| Total attachments | Total number of file attachments |
| Last attachment (bytes) | Size of file attachments for the node during last session |
| Last attachments | Number of file attachments for the node during last session |
| Additional information | More detailed information for the node |

## Using the Synchronization Sessions View

The Synchronization Sessions view displays information about every mobile user and each user's synchronization session for a given period of time. This can be accessed only by the Systems Administrator.

### *To use the Synchronization Sessions view*

1. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote.
2. In the link bar, select Synchronization Sessions.
3. In the Synchronization Period form, select the time frame for the period of time you are investigating.
4. In the Mobile Users list, select the Mobile Web Client that you want to examine.
5. In the Synchronization Sessions list, run a query using the appropriate values for the fields.

   See Table 19 for descriptions of the fields in this list.

Information displayed on the form and two list applets includes:

- Synchronization period with From and To fields. Required.
- Mobile users and the Application server for each.

- Synchronization session details.

Table 19. Fields in Synchronization Sessions List

| Field | Description |
|---|---|
| Synchronization Starts | Date and time when the synchronization session started. |
| Synchronization Ends | Date and time when the synchronization session ended. This will be empty if the synchronization session did not complete successfully. |
| Transactions | The number of transactions replicated to the mobile user during the session. |
| Transaction Size (MB) | The total size in MB of the transactions replicated to the mobile user during the session. |
| Client Merge Duration (Min) | Time (minutes) taken to merge transactions on client. |

## Using the Upload Statistics View

The Upload Statistics view displays information about transactions and attachment files that are:

- Created on each Mobile Web Client
- Received on the server
- Applied to the server

To check that transactions are made to the server after the client synchronizes, use this screen to verify the last set of transactions sent from the client to the server, and the last transaction applied to the server.

### *To use the Upload Statistics view*

1. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote.
2. In the link bar, select Upload Statistics.
3. In the Mobile Clients/Replication Servers list, select the Mobile Web Client that you want to monitor.

Table 20 gives descriptions of the fields on the Siebel Remote Upload Statistics form. View the appropriate column on the form to see statistics for items created on the client, items received on the server, or items applied on the server.

Table 20. Fields on Siebel Remote Upload Statistics Form

| Field | Description |
|---|---|
| Last Session | Duration of the last synchronization session. |
| Last File | Last transaction file created on the client, received on the server, or applied on the server. |
| Last Transaction | Last transaction created on the client, received on the server, or applied on the server. |
| Transaction Count - Last | Number of transactions created on the client, received on the server, or applied on the server during the last synchronization session. |
| Transaction Count - Total | Total transactions created on the client, received on the server, or applied on the server since the last database extract. |
| Operation Count - Last | Number of operations created on the client, received on the server, or applied on the server during the last synchronization session. |
| Operation Count - Total | Total number of operations created on the client, received on the server, or applied on the server since the last database extract. |
| KBytes - Last | Size of transactions created on the client, received on the server, or applied on the server during the last synchronization session, in kilobytes. |
| KBytes - Total | Size of transactions created on the client, received on the server, or applied on the server since the last database extract, in kilobytes. |
| Duration (Seconds) - Last | Duration of last synchronization session created on the client, received on the server, or applied on the server, in seconds. |
| Duration (Seconds) - Total | Total duration of all synchronization sessions since the last database extract, in seconds. |
| Attachment Files - Last | Number of attachment files created on the client, received on the server, or applied on the server during the last synchronization. |
| Attachment Files - Total | Number of attachment files created on the client, received on the server, or applied on the server since the last database extract. |
| Attachment Size (KBytes) - Last | Size of attachment files created on the client, received on the server, or applied on the server during the last synchronization session, in kilobytes. |
| Attachment Size (KBytes) - Total | Size of files created on the client, received on the server, or applied on the server since the last database extract, in kilobytes. |

## Using the Download Statistics View

The Download Statistics view displays information about transactions and attachment files that are:

- Created on the server
- Received on the client
- Applied to the client

### *To use the Download Statistics view*

1. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote.
2. In the link bar, select Download Statistics.
3. In the Mobile Clients/Replication Servers list, select the Mobile Web Client that you want to monitor.

Table 21 describes the fields on the Siebel Remote Download Statistics form. View the appropriate column on the form to see statistics for items created on the server, items received on the client, or items applied on the client.

Table 21. Fields on Siebel Remote Download Statistics Form

| Field | Description |
|---|---|
| Last Session | Date and time of last session created on the server, received on the client, or applied on the client. |
| Last File | Last transaction file created on the server, received on the client, or applied on the client. |
| Last Transaction | Last transaction created on the server, received on the client, or applied on the client. |
| Transaction Count - Last | Number of transactions created on the server, received on the client, or applied on the client during the last synchronization session. |
| Transaction Count - Total | Total transactions created on the server, received on the client, or applied on the client since the last database extract. |
| Operation Count - Last | Number of operations created on the server, received on the client, or applied on the client during the last synchronization session. |
| Operation Count - Total | Total number of operations created on the server, received on the client, or applied on the client since the last database extract. |
| KBytes - Last | Size of transactions created on the server, received on the client, or applied on the client during the last synchronization session, in kilobytes. |
| KBytes - Total | Size of transactions created on the server, received on the client, or applied on the client since the last database extract, in kilobytes. |
| Duration (Seconds) - Last | Duration of last synchronization session, in seconds. |

Table 21. Fields on Siebel Remote Download Statistics Form

| Field | Description |
|---|---|
| Duration (Seconds) - Total | Total duration of all synchronization sessions since the last database extract of the node, in seconds. |
| Attachment File Count - Last | Number of attachment files created on the server, received on the client, or applied on the client during the last synchronization. |
| Attachment File Count - Total | Number of attachment files created on the server, received on the client, or applied on the client since the last database extract of this node. |
| Attachment Size (KBytes) - Last | Size of the file attachments created on the server, received on the client, or applied on the client during the last synchronization session. |
| Attachment Size (KBytes) - Total | Size of all file attachments created on the server, received on the client, or applied on the client since the last database extract of this node. |

# Checking Siebel Remote Transaction Backlogs

Beginning with Siebel release 7.7, the Transaction Backlog view provides administrators with information about the number of Siebel Remote transactions that are backlogged. A backlog is a set of transactions that is waiting to be routed by the Transaction Router. The following kinds of backlog information are available:

- Count of all backlogged transactions in the Enterprise (for all mobile users, on all Siebel Remote Servers)
- Count of all backlogged transactions on a selected server (for all mobile users who have transactions waiting on the selected Siebel Remote Server)
- Count of all backlogged transactions for a selected mobile user, on the Siebel Remote Server for that user

Backlog information is available for the sum of all Siebel Servers in the enterprise, for individual Siebel Servers, and for individual mobile users. For general information about Siebel Servers, see *Siebel System Administration Guide*.

Use the following procedure to check the combined backlog for all mobile users in the Enterprise.

### *To check the combined transaction backlog for all mobile users in the Enterprise*

1. From the Site Map, select Administration - Siebel Remote > Transaction Backlog.
2. In the Enterprise form, click Calculate Backlog.

   The number of backlogged transactions appears. This number is the sum of all the individual server backlogs.

Use the following procedure to check the backlog for a selected Siebel Remote Server.

### *To check the transaction backlog for a selected Siebel Server*

1. From the Site Map, select Administration - Siebel Remote > Transaction Backlog.
2. In the Server(s) list, select the server you want to check, and click Calculate Server Backlog.

Use the following procedure to check the backlog for a selected mobile user.

**NOTE:** The count of backlogged transactions for a selected mobile user may include some transactions that are not visible to that user.

### *To check the transaction backlog for a selected mobile user*

1. From the Site Map, select Administration - Siebel Remote > Transaction Backlog.
2. In the Users list, use standard query techniques to select the user you want to track, and then click Calculate Mobile Client Backlog.

## Monitoring Transaction Logs

Use the Administration - Siebel Remote views to monitor transaction logs.

### *To monitor transaction logs*

1. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote.
2. In the link bar, select Transaction Log.

   The Transaction Log list displays information about each transaction. The Operation Types for Siebel Remote transactions are as follows:

| Operation Type | Description |
|---|---|
| **D** | Delete single row |
| **E** | Delete multiple rows |
| **F** | Delete cascading rows |
| **I** | Insert single row |
| **U** | Update single row |
| **V** | Update multiple rows |
| **X** | Insert set-based rows |
| **Y** | Update set-based rows |
| **Z** | Delete set-based rows |
| **G** | Merge multiple rows |

The fields in the Transaction Log list are described below.

| Field | Description |
|---|---|
| Node Number | ID of the node that generated the transaction |
| Operation | Insert (I), update (U), delete (D), … |
| Item | Table name of the transaction |
| File Flag | Flag to indicate whether the transaction is on a file attachment-related table |
| Transaction ID | Sequence number of the transaction |
| Transaction Row ID | ROW_ID of the transaction |
| Transaction Updated | Timestamp showing when the transaction was last updated |

The fields in the Detail form are described below, except for fields that are identical to those in the Transaction Log list.

| Field | Description |
|---|---|
| Transaction Updated by | ROW_ID of the user who last updated the transaction |
| Transaction Conflict ID | ID of the conflict for the base table record |
| Transaction Mod ID | Version number of the base table record |
| Visibility Level | Visibility event level for the transaction: Enterprise(E), Limited(L) |
| Visibility Event | Whether the transaction is a visibility event |
| Related Visibility Event | Whether the transaction is a related visibility event |
| Dock Object Code | Dock object code of the transaction instance |
| Primary Table Row ID | ROW_ID of Primary table for the dock object instance |
| Length (Long Log data) | Real length of the long column of the transaction data |
| Log Data 1, 2, 3, 4, long | These fields hold transaction data |

3 To monitor transaction logs by operation, select Transaction Log By Operation from the link bar.

The fields in the Transaction Log By Operation view provide information about the number of transactions for each operation using the following fields:

| Field | Description |
|---|---|
| Operation | Type of the operation |
| Number of transactions | Total number of transaction for the operation type |

4 To monitor transaction logs by node, select Transaction Log By Node from the link bar.

The Transaction Log By Node view displays the number of transactions for each node. A node is a Mobile Web Client or Regional server. The following fields appear in the view:

| Field | Description |
|---|---|
| Node Name | Mobile client node name |
| Number of transactions | Total number of transactions for the operation type |

**NOTE:** This view only shows transactions from mobile users and regional nodes. It does not include transactions created by nonmobile users who connect directly to the server.

5 To monitor transaction logs by table, select Transaction Log By Table from the link bar.

The Transaction Log By Table view displays the number of transactions for each table in the Siebel database. The following fields appear in the view:

| Field | Description |
|---|---|
| Item | Table name |
| Number of transactions | Total number of transactions for the operation type |

## Monitoring Siebel Remote Server Operation Status

Use the Server Manager to monitor Siebel Remote server tasks that are running or have completed by viewing their state values and statistics.

State values contain information about the current operation of a server task or the server component for which the task is running. Statistics are recorded at the task level for server tasks. When the task completes its operation, task-level statistics roll up to the component and server levels.

You can use the Server Manager to view the state values and statistics for Siebel Remote server components and tasks. For more information on state values and statistics, see *System Monitoring and Diagnostics Guide for Siebel eBusiness Applications*.

# Data Synchronization

This section discusses the implications of data synchronization for the administrator. These explanations will help you respond to and coordinate synchronization issues for Mobile Web Clients. For details regarding the synchronization process, refer to "Siebel Remote Flow Diagram" on page 18 and "About Synchronizing a Mobile Web Client" on page 33.

**NOTE:** A client that is extracted gets the latest transactions. The routers no longer need to route transactions that were generated before the extract started. Extracting all Mobile Web Clients again allows the Transaction Processor to delete all transactions that were created before the extraction from the transaction log.

# Managing Synchronization Frequency

You are responsible for developing guidelines for synchronization frequency, the frequency at which your mobile users synchronize. You should recommend an appropriate interval between synchronization sessions, taking into consideration your organization and its activities. The frequency and schedule you define for the Transaction Router should also be considered.

Initially, you may recommend that mobile users synchronize once a day. You can then use the following factors to evaluate your synchronization traffic and determine if your synchronization schedule needs refinement.

**Connection time.** Frequent synchronization reduces the volume of transactions to be transmitted during a synchronization session, thereby reducing connection time for that session. You should evaluate the difference in connection time and cost between less frequent but longer synchronization sessions, and more frequent but shorter sessions.

**Disk space requirements.** Frequent synchronization reduces the number of transactions accumulated between sessions, thereby reducing the demand for disk space on both the Siebel Remote server and the client. You should evaluate the availability of disk space and determine the trade-off between increasing disk space and increasing synchronization frequency.

**Database volatility.** Your synchronization schedule determines when changes to the server database are reflected in client databases. In industries with high sales volatility, frequent synchronization can be a significant advantage. You should investigate the value of frequent synchronization to your organization in such a case.

Remember that the Transaction Router detects and routes database changes to client outboxes for subsequent transmission during synchronization. Therefore, the operating status of the Transaction Router on each application server will affect data availability for mobile users. In a highly volatile environment, you may want to run multiple Transaction Routers for each application server. Additionally, close monitoring of this component is highly recommended.

**Mobile users' productivity.** Frequent synchronization affects your mobile users' time, and potentially affects their productivity. You should consider your users' environments and their convenience in completing a synchronization session.

**Server modem connections.** The ratio of Mobile Web Clients to server modem connections influences your synchronization schedule. If your ratio is high, you may need to assign specific synchronization times to your users.

# TrickleSync Synchronization

TrickleSync functionality performs database synchronization automatically. Once enabled and configured by the mobile user, the TrickleSync Agent runs in the background at scheduled times. It will perform automatic synchronization when connected to the network. This can improve the overall usability of the system by increasing the frequency of synchronization sessions.

This is invoked on the Mobile Web Client. One of the options is a synchronization reminder that prompts the user to synchronize if a specified period passes without a synchronization session. For more information about TrickleSync, see "Using TrickleSync" on page 76.

# Sending Messages to Mobile Users

The message-of-the-day feature enables you to send messages to mobile users. When a mobile user synchronizes with the server, the synchronizing client displays the message of the day to the mobile user each time docking occurs.

### *To enable the message of the day*

- Place the motd.txt file in the admin subdirectory within each Siebel Remote server root directory.

  For example, if Siebel is installed on D:\sea700, then place the modt.txt file in:

  ```
  D:\sea700\siebsrvr\admin
  ```

  Mobile users will see the message when they synchronize again.

# Refreshing a Client Database

Occasionally, you may need to refresh the local database for a Mobile Web Client. For example, whenever there is a change to the system preference parameters, a refresh or reextract is necessary for the new settings to take effect. See "Setting Siebel Remote System Preferences" on page 48 for more details.

### *To refresh a client database*

1. If the Mobile Web Client has transactions ready for uploading, the user should synchronize and send changes to the server.
2. Make sure that the Transaction Merger successfully applied transactions to the server database.

   There are three ways to determine when transactions from a particular client have been processed:

   - Use the Siebel Server Manager to check whether the Transaction Merger successfully applied transactions for the Mobile Web Client.
   - Inspect the client's inbox directory on the Siebel Remote server. There should not be any files with the .dx extension in the inbox directory.
   - Use the Siebel Client Status screen to check whether Transaction Merger successfully applied transactions for the Mobile Web Client.
3. Run Database Extract for the Mobile Web Client.

   Make sure that the parameter Save Transactions is TRUE.

   For information on running a database extract, see "Creating Mobile Web Client User Accounts and Privileges" on page 100.

4 When you have completed a database extract, notify the user to reinitialize the Mobile Web Client's local database.

For information on initializing a Mobile Web Client database, see "Initializing a Mobile Web Client Database" on page 111.

**NOTE:** After you have performed a database extract for a Mobile Web Client, the client database must be initialized before any data exchange between the client and the server can occur. This includes the uploading of any client database changes.

# Deactivating and Reactivating a Mobile Web Client

To deactivate and reactivate Mobile Web Clients, use the following procedure:

### *To deactivate or reactivate a Mobile Web Client*

1 From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote.

2 In the link bar, select Mobile Clients.

3 In the Parent Server form, use the record navigation buttons or standard query techniques to select the appropriate server.

4 In the Mobile Clients list, select the Mobile Web Client.

5 To deactivate the Mobile Web Client, do the following:

   a In the End Date field, enter a new effective end date that is the current or future date.

   This will cause the Transaction Router, Transaction Merger, and Synchronization Manager to stop processing requests for the client.

   b Click the menu button, and then click Save Record.

   c Delete the docking folder for the Mobile Web Client from the Siebel Remote directory on the Siebel Remote server.

   This will prevent the mobile user from establishing a synchronization session with the server.

   d Stop the Transaction Processor and any Router tasks to clear out any cached user information.

6 To reactivate a Mobile Web Client, do the following:

   a Clear the End Date field so that it does not contain a date.

   **NOTE:** Even if the date is in the future when you are clearing End Date, it is still necessary to reextract and reinitialize.

   b Click the menu button, and then click Save Record.

   c Reextract and reinitialize the Mobile Web Client.

# Deleting a Mobile Web Client

When you delete a Mobile Web Client, exercise care to keep the user status in sync across the Enterprise Server components. Before deleting a mobile user, be sure to set the end date to the current date or an earlier date. Ideally, shut down the Siebel Remote Enterprise Server components before actually deleting the user record. If the Enterprise Server components must be kept running, wait at least several hours to make sure they have been updated with the new status before actually deleting the user records.

### *To delete a Mobile Web Client*

1. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote.
2. In the link bar, select Mobile Clients.
3. In the Parent Server form, use the record navigation buttons or standard query techniques to select the appropriate server.
4. In the Mobile Clients list, select the mobile client you want to delete.
5. In the End Date field, set a new effective end date that is the current date or an earlier date.

   This will cause the Transaction Router, Transaction Merger, and Synchronization Manager to stop processing requests for the Mobile Web Client.
6. Click the menu button, and then click Save Record.
7. Select the mobile client record for which you set an end date, click the menu button, and click Delete Record.

   Repeat Step 4 through Step 7 for each Mobile Web Client you wish to delete.
8. Delete the docking folder for the Mobile Web Client from the Siebel Remote directory on the Siebel Remote server.

   This will prevent the mobile user from establishing a synchronization session with the server.

   The docking folder is located in the Docking subdirectory within the Siebel Server root directory. See "Creating Siebel Server Directories for Mobile Web Clients" on page 28 for more details about these folders.

# Changing Routing Models

There are times when an administrator may need to change the Routing Model for a particular mobile user.

**NOTE:** Routing models are not related to responsibilities. If a routing model is changed, make sure the user has corresponding responsibilities so that the user does not have access to views for which data is not routed by the new Routing Model.

***To change the routing model for a mobile user***

1. Before changing a mobile user's routing model, instruct that user to synchronize with the server and not to invoke any local database changes until after the next synchronization.
2. Change the user's routing model.
   a. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote.
   b. In the link bar, select Mobile Clients.
   c. Select the mobile client record and make your changes.
   d. Save your changes.
3. Perform a database extract.

   See "Creating Mobile Web Client User Accounts and Privileges" on page 100.
4. Prompt the mobile user to synchronize.

   The mobile user can download the new extract at the beginning of the synchronization session.

   This process will reinitialize the Mobile Web Client database. See "Initializing a Mobile Web Client Database" on page 111.
5. After downloading the new extract, the mobile user can resume normal operation.

# Adding New Mobile Users

To add new mobile users, do the following:

1. Register or create the Mobile Web Client. For details, see "Registering a Mobile Web Client" on page 85.
2. Perform a database extract for the new Mobile Web Client. For details, see "Creating Mobile Web Client User Accounts and Privileges" on page 100.
3. Perform client initialization. For details, see "Database Extraction for a Mobile Web Client" on page 103.

# How to Set Client-Side Logging

This section describes how Mobile Web Clients can set the parameters to control client-side logging. Each heading that follows addresses a different aspect of the process.

# Event Levels

Users set the type of information to be collected by temporarily changing the environment variable SIEBEL_LOG_EVENTS.

### *To temporarily change the environment variable SIEBEL_LOG_EVENTS*

1. Open a DOS Window and change the current directory to the <ClientRootDir>\bin directory.
2. Enter the following command:

   ```
   set SIEBEL_LOG_EVENTS=<event level>
   ```

   **NOTE:** <event level> can be between 0 and 5. Higher levels collect more detailed information and use more disk space. With the default value of 1, minimal information is collected in the trace file. Set event level to 3 or higher to troubleshoot a problem, or if assistance is required from Siebel Technical Support.

   When setting to the higher levels, check that sufficient disk space to is available.

3. Launch the Siebel client application with the appropriate parameter from the same DOS Window.

   For example, to launch the Siebel application with user=USERNAME, password=PASSWORD, and CFG (name and path of the CFG file), and then to log in to local database, enter:

   ```
   siebel /u USERNAME /p PASSWORD /c CFG/ d local
   ```

This environment variable can also be changed permanently. To do this, follow the procedures below.

### *To permanently change environment variable SIEBEL_LOG_EVENTS in Win2000*

1. In the Windows Start menu, choose Settings > Control Panel > System > Advanced Environment Variables.
2. Click New and enter the following fields:

   Variable Name = SIEBEL_LOG_EVENTS

   Variable Value = <event level>

3. Click OK.

   The newly added environment variable will be seen in your User Variables list.

   The new setting becomes effective the next time you launch the Siebel Client application.

### *To permanently change environment variable SIEBEL_LOG_EVENTS in Win NT*

1. In the Windows Start menu, choose Settings > Control Panel > System > Environment Tab.
2. Click the Variable box and set to SIEBEL_LOG_EVENTS.
3. Click the Value box and set to <event level>.

4 Click Set; then the newly added environment variable will be seen in User Variables list.

The new setting becomes effective the next time you launch the Siebel Client application.

***To permanently change environment variable SIEBEL_LOG_EVENTS in Windows XP***

1 In the Windows Start menu, choose Settings > Control Panel > System > Advanced tab.

2 Click the Environment Variables button.

3 In the User variables area, click New.

4 In the Variable name field, enter SIEBEL_LOG_EVENTS.

5 In the Variable value field, enter <event level>.

6 Click OK to display the newly added environment variable in the User variables list.

The new setting becomes effective the next time you launch the Siebel Client application.

# Log File Naming

There are two types of log file naming conventions:

- By default, the program log file names are <program>.log. (Siebel.log or UpgWiz.log.)
- The Process/Task Id log file name is Syncthrd_nnn_yyy.log.

  Where nnn specifies the process-id and yyy specifies the task-id.

# Archiving

The Archive_number is a positive integer that determines how many log files will be archived. By default, only 10 archived log files are retained and the oldest log file is deleted.

***To change the archive_number***

1 Open a DOS Window and change the current directory to the <ClientRootDir>\bin directory.

2 Enter the following command:

```
set SIEBEL_LOG_ARCHIVES=<archive_number>
```

**NOTE:** For log files that belong to 1, each execution of the program creates a new log, <program>.log, while archiving the previous versions as <program>_1.log, <program>_2.log, and so on, pushing down the numbers in an increasing order chronologically.
Log files that belong to 2 will not be deleted regardless of the value of the SIEBEL_LOG_ARCHIVES variable.

## Log File Location

The location of the log file of a client program is determined by the following set of variables.

If SIEBEL_LOG_DIR is set as SIEBEL_LOG_DIR=<dir>, the log file will be created in that directory. Make sure this directory exists and there is access permission to write a file in that location.

If no SIEBEL_LOG_DIR is specified, the log file will be created in the <ClientRootDir>\log directory.

# Event Tracing for Locking

Beginning with Siebel 7.5, Siebel releases include enhanced tracing capability for Remote and Replication Manager components. This will improve system diagnostics. Trace files track SQL statements the components issue and include some information about the task or function in progress at the time.

The addition of the locks to the tracing mechanism will display the reason locks were acquired, or released, for the designated components. A lock is a handle used by server components to determine which component has access to a specific object such as a DX file in the Transaction Processor directory. The inclusion of locks in the trace files provides additional information for troubleshooting problems. This will help administrators who are working with Siebel support staff to lessen the contention between critical server components for Remote and Replication Manager.

For example, assume there is a problem with the Transaction Processor (TXNPROC). It may be locking certain objects such as DX files in the docking directory and not releasing these. If you set the Log Level to 4 or 5, information about the locks will be captured in the log file for TXNPROC.

This functionality is for the Remote and Replication Manager server components on the HQ or regional nodes, and is not available to the mobile user. These components include:

- Database Extract
- Parallel Database Extract
- Replication Agent
- Synchronization Manager
- Transaction Merger
- Transaction Processor
- Transaction Router

Tracing and logging of locks for the components includes the following objects on the application server:

- Dobjinst.dbf
- DX files
- Inbox
- Outbox
- Visdata.dbf

Use the standard Siebel event tracing mechanisms to enable the tracing for these locks. Logging is controlled by the trace level (level 4 or higher) of the component. The procedures that follow describe how to do this. This does not require restarting of the application server.

### *To set tracing for locks using the GUI*

1. From the application-level menu, choose Navigate > Site Map > Administration – Server Configuration > Servers > Components > Events.
2. In the Components list, select the component you need to trace.

   Choices include the components listed earlier in this topic.
3. In the Events list, select the Event Type you want to trace.

   Use Event Description field to help determine which Event Type to choose.
4. In the selected Events list record, set Log Level = 4.

   The log file for the component selected above will contain the tracing information.

   Repeat this procedure for additional traces of locks you want to trace.

### *To set tracing for locks using the command line*

- From the Srvrmgr command line, enter:

```
srvrmgr: change evtloglvl <event type> = 4 for componentname
```

# Handling Failure and Recovery

Siebel Remote is designed to minimize the impact of a software, communications, or hardware failure. This section describes the most likely failures and how to recover from them:

# Siebel Remote Transmission Failure

Mobile Web Clients may experience occasional transmission failures. These failures may be caused by noise on the telephone line. The Siebel Remote Synchronization Client and Synchronization Manager inspect and verify the integrity of every Siebel Remote transmission. If an error is detected, Siebel Remote automatically retransmits the files until the synchronization is successful.

# Siebel Remote Server Failure

All Siebel server programs are designed to recover automatically from a failure on a Siebel Remote server. After returning the system to an operational state, use the Server Manager to restart the Siebel Server components. For information on using the Server Manager, see *Siebel System Administration Guide*.

# Siebel Remote Server Media Failure

Media failures on Siebel Remote servers can cause serious disruptions of data synchronization with mobile users. After the Transaction Router routes transactions to files on the Siebel Remote server, the Transaction Processor deletes those transactions from the server database master Txn table.

You should run your Siebel Remote server with a redundant disk configuration. This minimizes the risk of data loss if a device fails that contains inbox/outbox directories for Mobile Web Clients. If a media failure does occur that results in file corruption or loss on a Siebel Remote server, you need to perform the following procedure.

***To recover from a media failure on the Siebel Remote server***

1. Fix the directories on the disk.
2. Instruct the user to send changes to the Siebel Remote server.
3. Make sure Transaction Merger applied the transactions to the server database.
4. Run Database Extract and reinitialize the Mobile Web Client's local database.

# File Server Media Failure

The Siebel file server stores attachment files such as literature files and submitted correspondence files. Literature files are more or less static and can be recovered from the most recent backup. Any attachment files created after your last backup may be lost. You should configure the file server with a redundant disk configuration to minimize the risk of data loss.

# Database Server Failure

If your RDBMS fails, the database administrator must diagnose and rectify the problem. When the system returns to an operational state, use the Siebel Server Manager to restart the Siebel Remote components. Siebel Remote components automatically recovers their process state from the last committed transaction. A reextraction of the Mobile Web Clients may be necessary.

If the database is recovered up to the point of failure, no action is required because there is no loss of data. However, if the database is recovered up to a point of time prior to the point of failure, then you must reextract and reinitialize all Mobile Web Clients. In this case, follow the steps below after restoring the database backup:

1. Disable the Synchronization Manager component.
2. Stop the Transaction Router, Transaction Merger and Transaction Processor tasks if they are running.
3. Reextract all Mobile Web Clients.
4. Start the Transaction Processor task with the `TS DB Recreate` parameter set to TRUE.
5. Start the Transaction Router and Transaction Merger tasks.
6. Enable the Synchronization Manager component.

After you complete the steps above, the next action depends on when the last backup was completed, when the failure occurred, and when the user synchronized.

- If the user synchronized at a time between the time the backup was completed and the time the failure occurred, you must:

  a. Rename the locale database `<SiebelClientRoot>\local\sse_data.dbf` and `dicdata.dat`.

  b. Reinitialize the Mobile Web Client.

  Any changes in the Mobile Web Client database that were not sent to the server will be lost.

- If the user synchronized before the backup was completed, Mobile Web Clients can download a new database.

  If the database extract was executed with the `Save Client Transaction` parameter set to TRUE, none of the Mobile Web Client changes will be lost.

# Server Database Records Truncated or Changed

If records are truncated or deleted on the Server database and these transactions are sent to the Mobile Web Clients, this cannot be changed or reversed. Even if the database is restored, this will cause data mismatch and corruption. After restoring the database to the time it failed, you must reextract all the mobile users.

# Database Server Media Failure

A head crash or other media failure on the Siebel database server may render the database unusable. Therefore, your database administrator should take preventive measures to protect against such occurrences, such as disk mirroring or online backups and RAID (redundant array of independent disks).

If restoring the database results in a permanent loss of transactions from the server, the Transaction Router may have routed some of those lost transactions to Mobile Web Clients prior to the crash. Full recovery may then require a second step: resynchronizing the server and client databases.

### *To diagnose and restore database synchronization*

1. Stop the Transaction Router, Transaction Merger, and Transaction Processor on the Siebel Remote server, and use Siebel Server Manager to disable the Synchronization Manager.
2. After the database administrator has returned the database server to an operational state, use the Siebel Server to determine if Transaction Router sent data to any clients after the last backup of the database server.

   These are the clients whose databases require reinitialization. To do this:

   a. Look at the route log file txnroute_<taskid>.log to see if there is anything about not processing a client because of a corrupted file.

   b. If the log indicates that dobjinst.dbf (visibility database) is bad, you must reextract the database for the mobile user identified in the log.

      For dbxtract, see "Creating Mobile Web Client User Accounts and Privileges" on page 100.

      **NOTE:** A database extract may not be enough to restore a database. For example: On Monday a backup is invoked for two mobile users (A and B). On Tuesday user A synchronizes, yet that database is lost on Wednesday. User A's database is restored from Monday's backup.

      Then a dbxtract is invoked for Users A and B. User B synchronizes without any errors. However, user A receives a mismatch error because the routed values are different between the client and the server. User A must delete the local DB before acquiring a new database.
3. Notify these users that their databases must be reinitialized.
4. Run Database Extract for the affected clients.

   For information on running Database Extract, see "Creating Mobile Web Client User Accounts and Privileges" on page 100.
5. Notify the affected users that their client databases will be reinitialized the next time they synchronize.

   For information on initializing a Mobile Web Client database, see "Initializing a Mobile Web Client Database" on page 111. Since this is a potentially lengthy process, users may want to finish this process at different times. For example, users located close to a field office may want to take advantage of a LAN connection to reduce download time. Others may want to reinitialize during the evening or night, when telephone rates are lower.

# Siebel Client Database Failure

It is impractical to perform backups of Siebel Mobile Web Clients, because the information contained on the local databases is a subset of the information stored on the server database. Therefore, in the case of a laptop or local database failure, the procedure is to reextract and reinitialize the local database for this Mobile Web Client. Any change made on the local database since the last docking will be lost. It is strongly recommended that the synchronization with the server be executed regularly.

If the client machine loses power during a merger process, then the Local database may be corrupted. To avoid this, make sure the client machine has sufficient power before synchronization.

If a client database becomes unusable because of a media failure or other event, you must refresh the client database. This requires that you run Database Extract for the client. Siebel Systems does not support restoration of local databases because it may result in inconsistency between the local and server databases. For information on running Database Extract, see "Creating Mobile Web Client User Accounts and Privileges" on page 100.

**NOTE:** Depending on the kind of failure, database changes and file attachments that were awaiting upload during the next synchronization session may also be lost. In this case, the user must reenter them.

# Recovery from Client Initialization Failure

There may be times when initialization of the Mobile Web Client fails and you cannot continue the process. The procedure that follows describes how to recover from this.

***To recover from client initialization failure that cannot continue***

1. On the client machine, go to the $Siebel-Root\bin directory.
2. Find the file upgwiz.ucf and delete it.
3. Go to the $Siebel-Root\upgrade directory and delete all files.
4. Go to $Siebel-Root\local directory and delete all the content.
5. Rerun the initialization process.

# Restoring the File System After Recovery from a Previous Image

If the docking directory and file system were recovered from a previous image, then the docking directories may contain data that has already been sent to the Mobile Web Client or to the server. In this case, follow the steps below after restoring the file system:

1. Stop the Transaction Router, Transaction Merger, and Transaction Processor tasks if they are running.

2. Remove all the subdirectories under the <Siebel Server>\Docking directory, except the <Siebel Server>\Docking\Txnproc subdirectory.
3. Reextract all Mobile Web Clients.
4. Start the Transaction Processor task with the parameter TS DB Recreate set to TRUE.
5. Start the Transaction Router and Transaction Merger tasks.
6. Reinitialize all Mobile Web Clients.

# 8 Siebel Remote Reports

This chapter describes standardized status reports, how Mobile Web Clients can set the parameters to control client-side logging, and enhanced tracing capability for Remote and Replication Manager components.

Siebel provides a number of online reports that show critical information about Siebel Remote and Replication Manager components and system status. These allow the administrator to obtain additional information about the Remote components and mobile users. This information will enhance the Siebel administrator's ability to monitor and manage a Siebel Remote and Replication Manager deployment.

This information will also enhance the support, and it can reduce administrative overhead.

## Status Reports

There are nine standardized reports available for administrators, with the following types of information:

- Synchronization frequency
- Transaction backlog
- Transaction Processor
- Mobile Users Usage

The nine reports are available on-line from the related views within the Administration - Siebel Remote screen. To access these reports, click the Reports button and select the desired report.

Table 22 lists the nine on-line reports accessed from the appropriate view within the Administration - Siebel Remote screen.

Table 22. Standardized Administrative Status Reports - Remote and Replication Manager

| Name of Report | Description | View |
|---|---|---|
| Mobile User Summary Report | This report is available on all server nodes that contain mobile users. Below is a list of information available in the report:<br>■ Date and time of last extraction<br>■ DB Extract Size (MB)<br>■ DB Extract Rows<br>■ Local DB Size (MB)<br>■ Date and time of last initialization<br>■ Date and time of last synchronization<br>  ■ Number of transactions<br>  ■ Size of transactions in MB<br>  ■ Duration of session<br>  ■ Duration of last merge on client<br>■ DX files last routed from the Server<br>■ DX files last routed to the Server<br>■ DX files received and merged into the Server by this user<br>■ DX files received and merged on to the Local database of this user<br>This report also indicates whether the user is active. Otherwise, it displays last accessed date, and the name of the Routing Model associated with the mobile user. | Client Status<br>Note that a Mobile Web Client can access this report from the User Preferences > Mobile User Summary view, provided that that view is included in the client's responsibilities. |
| Transaction Processor Status Report | This report shows the status of all active and inactive transaction processor entries in the current Enterprise server. It also includes details such as:<br>■ Date and time of last run<br>■ Duration of last run<br>■ Number of transactions and operations copied<br>■ Last DX file created and removed from the docking\txnproc folder on the corresponding application server | Processor Status |

Table 22. Standardized Administrative Status Reports - Remote and Replication Manager

| Name of Report | Description | View |
|---|---|---|
| Transaction Backlog Status Report | This report provides information concerning pending transactions in the queue from active mobile users and regional nodes with respect to the most recent transaction created on the HQ at the time the report was run. Fields in this report include the Mobile Web Client name and the number of pending transactions. | Transaction Log |
| Synchronization Frequency Report | This report displays the frequency of synchronization by every mobile user for a given period of time. Each line of the report includes Mobile User ID, App Server Name, and the number of times the user has synchronized in a given period. | Synchronization Sessions |
| Synchronization Session Report | This report provides information about synchronization sessions of every mobile user for a given period of time. The report provides mobile user information and details of their synchronization sessions. | Synchronization Sessions |
| Regional Node Status Report | This report is available on all server nodes that have subordinate regional nodes. This report should contains the following information about the regional node:<br>■ Date and time of last extraction<br>■ Date and time of last initialization<br>■ Date and time of last synchronization<br>■ DX files last routed to the node from the parent server<br>■ DX files last routed by the regional node to the parent server<br>■ DX files received and merged into the parent server<br>■ DX files received and merged into the regional database<br>The report should also indicate whether the regional node is currently active or end dated and its routing group. | Client Status |
| Active Mobile Users Usage Graph | This report is available on all server nodes that have mobile users. The report takes start date as a parameter from the user and it generates the following details for each day between the start date and the current date, in the form of a graph:<br>■ X-axis includes all the days in the specified range of dates<br>■ Bar graph displays the number of cumulative inits per day<br>■ Line graph displays the number of unique syncs per day. Number of unique syncs means number of mobile users that synchronized per day. | Synchronization Sessions |

Table 22. Standardized Administrative Status Reports - Remote and Replication Manager

| Name of Report | Description | View |
|---|---|---|
| Active Mobile Users Usage Table | This report is available on all server nodes that have mobile users. The report takes start date as a parameter from the user and it generates the following details for each day between the start date and current date in the form of a table:<br>■ Date column displaying all the days in the specified range of dates<br>■ Daily inits column displaying the number of initializations per day<br>■ Cumulative initializations column displaying the number of cumulative inits per day<br>■ Daily unique syncs column displaying the number of mobile users that synchronized per day | Synchronization Sessions |
| Transaction Router Backlog Report | This report provides information about:<br>■ The number of transactions that could potentially be routed to the active mobile users or regional nodes. This is calculated with respect to the most recent transaction created on the HQ at the time the report was run.<br>■ The latency of the active mobile users and regional nodes with respect to the time the report was run. The report displays the total, average, maximum, and minimum backlog and latency.<br>Latency is the time difference between the last transaction routed to the mobile user or regional node and the time the report was run. | Transaction Log |

The procedure that follows describes how to access these reports.

### *To access on-line Remote reports*

1 Navigate to the appropriate view within the Administration - Siebel Remote screen that has status reports available.

   For example, you may want reports from the Synchronization Sessions view.

2 Click the Reports button and select the desired report.

These reports are accessible only to the Administrator except for the Client Status report. The Client Status report is available to Mobile Web Clients as long as their responsibilities include this view. For additional information about responsibilities, see *Applications Administration Guide*.

Table 23 describes a report that is available only to Mobile Web Clients. To access this report, the Mobile Web Client must navigate to User Preferences > Remote Status, and then click the Reports button and select Current Siebel Remote Session.

Table 23. Standardized Report for Mobile Web Clients - Remote and Replication Manager

| Name of Report | Description | View |
|---|---|---|
| Current Siebel Remote Session | This report provides the following kinds of information about the selected session:<br>■ List of actions performed during the session, such as connecting to server, sending database changes, retrieving database changes, and retrieving requested files<br>■ Description of each action performed, including quantities of transactions, operations, and files involved<br>■ Action item details, where applicable | User Preferences > Remote Status |

# 9 Siebel Replication Manager Concepts

This chapter compares Siebel Replication Manager with Siebel Remote and discusses the benefits. Also included are the architecture, routing groups, routing rules, and components of Replication Manager. The final section covers items supported on regional servers.

# Siebel Replication Manager

Siebel Replication Manager replicates structured and unstructured data between parent enterprise servers and subordinate enterprise servers called Regional Nodes. Each regional node is a separate Enterprise Server with a Siebel Database, Siebel Servers, and Siebel File System for a set of connected and mobile users. The set of users is based on the set of registered users at the regional node.

Each regional node is created as a child of a parent node. A parent node is another enterprise server that is either the Headquarters Node (the master node) or another regional node. A regional node contains either a full replica or a subset of the parent data, consisting of both database records and file attachments. Siebel Replication Agent periodically synchronizes each regional node with its parent node to keep the data current at both locations.

Typically, a regional node is geographically separated from the Headquarters Node, but this is not a necessary condition.

This chapter discusses how Siebel Replication Manager works. The next three chapters explain how to install, configure, and administer Siebel Replication Manager and describe how to upgrade Regional Databases.

## Comparison with Siebel Remote

Siebel Replication Manager is based on the same architecture as Siebel Remote. It is very similar to Remote but is different in the following ways:

- Siebel Remote supports data synchronization with Mobile Web Clients. Remote provides each mobile user with a laptop-based extension of the Siebel database and File System. Mobile users can access local data while they are disconnected from the parent Siebel database and File System, and must periodically synchronize their local data with the parent database.
- Siebel Replication Manager supports server-based data replication within a hierarchical set of enterprise servers. Each replicated node in a Siebel Replication Manager deployment supports multiple users. Replication Manager synchronizes data between the parent node and its subordinated nodes. Each subordinate node is an enterprise server containing a subset of users from its parent enterprise server.

## Benefits of Siebel Replication Manager

Replication Manager solves the performance issue caused by the network bandwidth limitations and latency. Placing data closer to a cluster of connected users than the Headquarters Node can improve response time. This also allows users continuous access to Siebel applications even if the network link to the Headquarters Node is unreliable or only intermittently available.

Similarly, the ability of Siebel Remote mobile users to synchronize against a local Regional Node can decrease network costs and improve the performance and reliability of the synchronization process.

# Siebel Replication Manager Architecture

Replication Manager is a data replication technology that replicates data throughout a network of Siebel enterprise nodes. Replication Manager uses a hierarchical model to replicate data, starting with the master parent node, known as the Headquarters (HQ) node. The HQ node can replicate data to multiple subordinate nodes, known as Regional nodes. These nodes, in turn, can have subordinate nodes as well.

A Replication Manager implementation has a single master parent node, known as the Headquarters Node, and one or more Regional Nodes. The Headquarters Node contains the master set of data used by the Regional Nodes. A Regional Node can be:

- Subordinate to the Headquarters Node, and synchronize directly against it.
- Subordinate to another Regional Node, and synchronize directly against this regional node. When some Regional Nodes are created as children of other Regional Nodes, this is called a hierarchy of Regional Nodes.

Figure 10 shows a sample Siebel Replication Manager configuration, with two Regional Nodes that are children of the parent Headquarters Node.

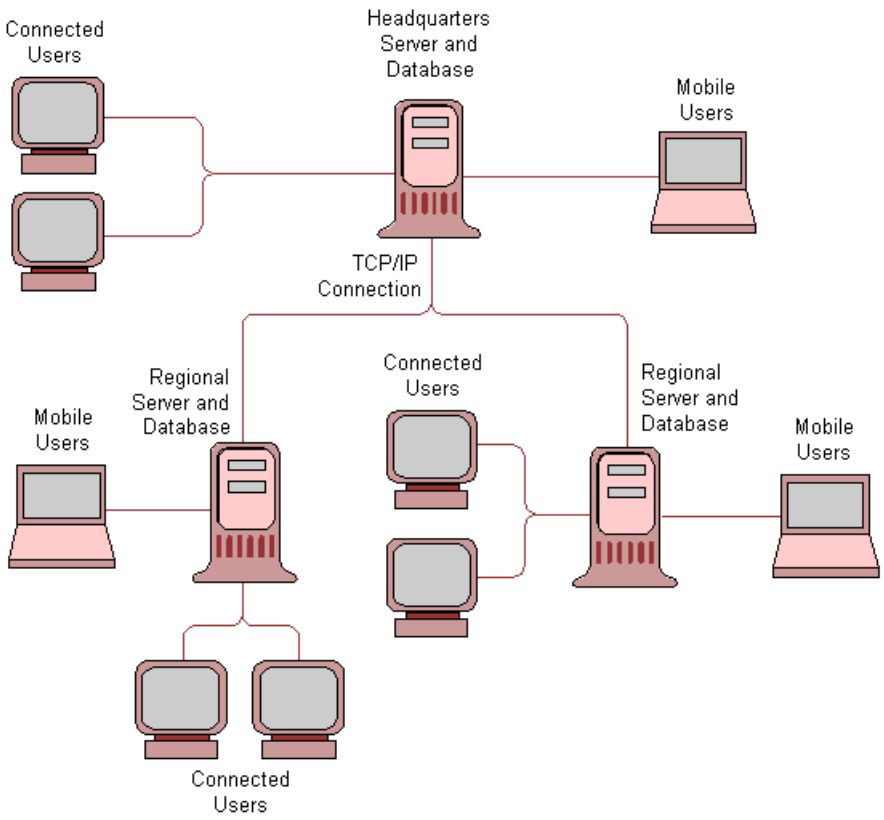


Figure 10. Example of a Siebel Replication Manager Configuration

The Headquarters Node comprises the master Siebel Database Server, Siebel File System, and one or more Siebel Servers. The master Database Server and File System contain the entire set of database records and file attachments used by the nodes. Siebel Servers at the Headquarters Node manage replication to:

- Regional Nodes that are children of the Headquarters Node
- Siebel Remote Mobile Web Clients whose users synchronize against the Headquarters Node

A Regional Node can support both connected users and Siebel Remote Mobile Web Clients whose users synchronize against the Regional Server.

# Routing Groups and Routing Rules

Each Regional Node contains either a full replica of the parent node's Database Server and File System or a subset of that data. You choose which of these to use by associating a routing group with the regional database. Do this when you register the Regional Node in the Administration - Siebel Remote view. This will determine how much of the data from the parent node will be replicated to the Regional Database. See "Registering a Regional Node" on page 175 for more detail about registering a regional node.

Routing group options include:

- **Regional Server – Full Copy.** When the regional database is assigned this routing group, all user data is replicated. The regional database is considered a full copy, or full replica, of the parent database. However, the full replica should *not* be used as a backup system for the headquarters node for the following reasons:
  - Some system-related data may not be replicated.
  - A regional node may require re-extraction if the parent node cannot be restored properly.
  - Not every user is replicated.
  - A regional node cannot be converted to an HQ node.

  **CAUTION:** If the Headquarters Node crashes, the full replica cannot be used as a new Headquarters Node for the reasons described above.
- **Regional Server – Standard.** When the regional database is assigned this routing group, the data replicated to the regional database is the union of data visible to connected users assigned to the Regional Node. Siebel Replication Manager applies standard routing rules to determine what data at the parent node is replicated to the Regional Node.

There are trade-offs between these two routing groups. Table 24 describes the pluses and minuses of each group.

Table 24. Trade-Offs Between Regional dB Routing Groups

| Routing Group Regional dB | Pluses | Minuses |
|---|---|---|
| Regional Server – Full Copy | Fast data routing - no routing rule is used. | More data stored at Regional node. |
| | Full access to every user's data. | More network traffic. |
| | | Requires more powerful hardware. |
| Regional Server – Standard | Only necessary data is routed. | Txn Routers have to determine what should be routed, therefore more processing time. |
| | Less data stored at Regional node, therefore requires less resources. | |
| | Less network traffic. | |

Also, consider the following guideline when using the Standard Regional Routing Group: if you will route more than half of the data on the parent node to the regional node, it may be more beneficial to use the Full Copy Routing Group rather than the Standard Regional Routing Group for optimal performance.

Routing rules determine the subset of Dock Object instances that Siebel Remote replicates to each Mobile Web Client. Dock objects are groupings of tables in the database that logically form Siebel business components. Dock objects are similar to business components.

Siebel applications provide pre-configured user routing rules. See "Dock Objects and Routing" on page 30 for more information.

Routing groups *cannot* be modified.

**NOTE:** If a dedicated user is assigned to multiple databases (one Regional or Headquarters database, and one Siebel Remote database), the data visible to that user is synchronized to each of those databases. However, a mobile user is assigned to *only one* database—HQ or Regional.

# About Filtering Regional Workflows

Beginning with Siebel 7.5, workflow administration allows filtering out of workflow processes that only apply at the regional level. Therefore, only records associated with workflows needed on the Mobile Web Client will be routed to those nodes. This decreases the volume of data to help optimize local databases. For more information about this topic, see *Siebel Business Process Designer Administration Guide*.

# Components of Siebel Replication Manager

This section lists the various pieces of software involved in setting up Regional Databases and in the normal operations of a Siebel Replication Manager implementation. For details on the listed components or utilities, see "Starting and Stopping Siebel Remote Server Components" on page 126.

Applicable components and utilities for the relevant Headquarters Node, Regional Node, and Mobile Web Client are listed below for each of two cases:

- Regional Node without Mobile Web Clients or child Regional Nodes
- Regional Node with Mobile Web Clients or child Regional Nodes

## Without Mobile Web Clients or Child Regional Nodes

For a Regional Node that has no Mobile Web Clients or child Regional Nodes, the following components and utilities are involved:

- On the Headquarters Server's Siebel Server, with references for descriptions:
  - **Transaction Processor.** See "Transaction Processor" on page 60.
  - **Transaction Router.** See "Transaction Router" on page 61.
  - **Transaction Merger.** See "Transaction Merger" on page 63.

- **Synchronization Manager.** See "Synchronization Manager" on page 64.
- **Database Extract (dbxtract).** See "Database Extract" on page 29.
- **Generate New Database (gennewdb).** See "Running the Generate New Database Component" on page 68.
- **Parallel Database extract (pdbxtract).** A version of the database extract component that invokes extractions in parallel. It is designed as an interactive component, using server infrastructure features to perform the tasks of data extractions in parallel for a large regional node. For more information, see "Extracting the Regional Database" on page 177.

- On the Regional Server's Siebel Server:
  - **Regional Database Initialization (srvrinit).** Program that loads dbxtract onto the Regional server.
  - **Replication Agent (repagent).** Server component that replicates the database to a Regional Node, according to the assigned Routing Group.

## With Mobile Web Clients or Child Regional Nodes

For a Regional Node that has either Mobile Web Clients or child Regional Nodes, or both, the following components and utilities are involved:

- On the Headquarters Server's Siebel Server:
  - Transaction Processor
  - Transaction Router
  - Transaction Merger
  - Synchronization Manager
  - Database Extract (dbxtract)
  - Parallel Database Extract (pdbxtract)
  - Generate New Database (gennewdb)
- On the Regional Server's Siebel Server:
  - Regional Database Initialization (srvrinit)
  - Replication Agent (repagent)
  - Transaction Processor
  - Transaction Router
  - Transaction Merger
  - Synchronization Manager
  - Database Extract (dbxtract)
  - Parallel Database Extract (pdbxtract)
  - Generate New Database (gennewdb)

# Components Supported on a Regional Node

Most Siebel Server processes *should be run only* on the Headquarters Server. Although it is possible to perform certain tasks from any Regional Server, doing so causes a delay before the data is replicated to the Headquarters Database and down to other Regional Databases. Thus data conflicts can occur.

You should only perform administration tasks (such as setting up and deactivating Mobile Web Clients using the Replication Server Administration view) when you are connected to the Headquarters Server. When using the Replication Server Administration view, be careful not to delete the Headquarters Server. Doing so will cause the processes to abort. You will then need to restore a backup or run SQL to rebuild the Headquarters Server record and the Mobile Web Clients.

**CAUTION:** Do not delete the Headquarters Server under any circumstances.

Although most Siebel Server processes should be run only on the Headquarters Server, certain features and server components can be used on Regional Nodes. Table 25 lists the features and server components that are supported on Regional Nodes, along with information about certain specific exceptions.

Table 25. Features and Server Components Supported on Regional Nodes

| Item | Type | Member of | Notes |
|---|---|---|---|
| Advanced Search | feature | | |
| Basic Siebel Call Center functionality | feature | | Associated with Call Center Object Manager server component.<br>Includes Quotes, Opportunities, Forecasting, Service Requests, Households, Campaigns, and SmartScripts. |
| Business Integration Batch Manager | server component | Enterprise Application Integration (EAI) component group | The only component in the EAI group that is not supported on Regional Nodes is Enterprise Integration Manager. |
| Business Integration Manager | server component | Enterprise Application Integration (EAI) component group | |
| Call Center Object Manager | server component | Siebel Call Center component group | |
| Charting | feature | | |
| Communications Configuration Manager | server component | Communications Management component group | Used by Computer Telephony Integration (CTI) feature. |

Table 25. Features and Server Components Supported on Regional Nodes

| Item | Type | Member of | Notes |
|---|---|---|---|
| Communications Outbound Manager | server component | Communications Management component group | |
| Communications Session Manager | server component | Communications Management component group | Used by Computer Telephony Integration (CTI) feature. |
| CTI (Computer Telephony Integration) | feature | | Associated with Communications Configuration Manager and Communications Session Manager server components.<br>Live call transfer from a user on one Regional Node to a user on another Regional Node is not supported. |
| Database Extract | server component | Siebel Remote component group | All components in the Siebel Remote component group are supported. |
| Document Server | server component | Siebel eDocuments component group | For Correspondence, Proposals and Presentations features. |
| EAI Object Manager | server component | Enterprise Application Integration (EAI) component group | |
| eService Object Manager | server component | Siebel Call Center component group | |
| Generate New Database | server component | Siebel Remote component group | |
| MQSeries AMI Receiver | server component | Enterprise Application Integration (EAI) component group | |
| MQSeries Server Receiver | server component | Enterprise Application Integration (EAI) component group | |
| MSMQ Receiver | server component | Enterprise Application Integration (EAI) component group | |
| Outbound Communications | feature | | Associated with Communications Outbound Manager server component. |

Table 25. Features and Server Components Supported on Regional Nodes

| Item | Type | Member of | Notes |
|---|---|---|---|
| Parallel Database Extract | server component | Siebel Remote component group | |
| Replication Agent | server component | Siebel Remote component group | |
| Report Server | feature | | Can be used to generate reports on the Regional Node. |
| Runtime Events | feature | | Runtime Events that contain wait states are not supported on Regional Nodes. |
| Basic Siebel Sales functionality | feature | | Associated with Sales Object Manager server component.<br>Includes Quotes, Opportunities, Forecasting, Service Requests, Households, Campaigns, and SmartScripts |
| Sales Object Manager | server component | Siebel Sales | |
| Field Service Object Manager | server component | Field Service component group | |
| State Model | feature | | |
| Synchronization Manager | server component | Siebel Remote component group | |
| Transaction Merger | server component | Siebel Remote component group | |
| Transaction Processor | server component | Siebel Remote component group | |
| Transaction Router | server component | Siebel Remote component group | |
| Workflow Process Batch Manager | server component | Workflow Management component group | |

Table 25. Features and Server Components Supported on Regional Nodes

| Item | Type | Member of | Notes |
|---|---|---|---|
| Workflow Processes | feature | | Associated with Workflow Process Batch Manager and Workflow Process Manager components.<br><br>The following kinds of Workflow Processes are *not* supported on Regional Nodes:<br><br>Processes that are defined as persistent by an administrator.<br><br>Processes that contain wait states with durations expressed in minutes, hours, or days.<br><br>Processes that are triggered by Workflow Policies.<br><br>Workflow Policies and persistent workflows are not supported in a replicated environment. |
| Workflow Process Manager | server component | Workflow Management component group | |

# 10 Implementing Siebel Replication Manager

This chapter describes the instructions and procedures for implementing Siebel Replication Manager. Before you proceed with the implementation, you must completely install and test the Siebel Database Server, Siebel Server or Servers, and Siebel File System in the Headquarters Node.

For instructions on completing these tasks, see the Siebel Server installation guide for the operating system you are using.

As an overview, to implement each Regional Node you must implement four main phases:

- Install a new Enterprise Server
- Define the Regional Node
- Extract the Regional Database
- Initialize the Regional Database

Table 26 shows the tasks for Implementing Replication Manager.

Table 26. Tasks for Implementing Siebel Replication Manager

| Task | Where Performed |
|---|---|
| Installing a New Enterprise Server on page 172 | New Server |
| Installing the Siebel Name Server on page 172 | New Server |
| Installing the Siebel Server on page 172 | New Server |
| Installing the Siebel Database Server on page 173 | New Server |
| Installing the Siebel File System on page 174 | New Regional File Server |
| Defining the Regional Node on page 174 | Parent Node |
| Registering a Regional Node on page 175 | Parent Node |
| Adding Mobile Users to the Regional Node on page 176 | Parent Node |
| Extracting the Regional Database on page 177 | Parent Server |
| Initializing the Regional Database on page 180 | New Server |
| Configuring Regional Node for Mobile Web Clients on page 186 | New Server |
| Starting Replication Agent on page 186 | Regional Node |
| Setting Up Additional Application Servers on Regional Nodes on page 188 | New Server |

# Installing a New Enterprise Server

Installation of the Regional Node essentially requires setting up a new Enterprise Server with a different name from that of the HQ Enterprise. This includes the following Siebel components:

- Siebel Name Server
- Siebel Database Server
- Siebel Server
- Siebel File System

For more information on installing these components, see the Siebel Server installation guide for your operating system.

**NOTE:** Setting up the Siebel Database Server on the Regional Node is very different from setting it up on the HQ Server.

## Installing the Siebel Name Server

The Siebel Name Server includes a persistent store for configuration parameters and run-time information. The Name Server is part of the logical entity called Siebel Gateway. Each Siebel Server accesses the Name Server at start-up and periodically throughout its operation. Consequently, there must be a reliable network connection to the Name Server. For this reason, it is strongly recommended that you install a Name Server local to your Regional Node. For more information about installing the Siebel Name Server, see the Siebel Server installation guide for the operating system you are using.

This server must be installed and running before you can proceed with the remainder of the Regional Node installation. For more information on the Siebel Name Server and Siebel Server, see *Siebel System Administration Guide*.

## Installing the Siebel Server

Each Regional Node must have at least one Siebel server, which is used to synchronize the Regional Node with its parent. This Siebel Server may also support Siebel Remote mobile users. Depending on the number of users and applications to be supported, additional Siebel Servers may be required.

Before you install the Regional Siebel Server, you may first need to install the correct ODBC driver for the RDBMS vendor. For an Oracle RDBMS, you do not need to install an ODBC driver. For other supported RDBMS vendors, you must install an ODBC driver. For the name of the required ODBC driver and version information, see the system requirements and supported platforms documentation for your Siebel application. System requirements and supported platforms documentation for all Siebel applications can be found on Siebel SupportWeb at http://ebusiness.siebel.com/supportweb/.

For instructions on installing the Siebel Server, see the Siebel Server installation guide for the operating system you are using. When prompted by the Siebel Server installation program, be certain to specify the correct connectivity information for the Siebel Name Server, Siebel File System, and Siebel Database Server on the Regional Node. Each Regional Node should have a unique Enterprise Server name associated with it.

## Installing the Siebel Database Server

For Regional Nodes that support smaller numbers of mobile or connected users, the Regional Database Server may be installed on the same physical server that will support the Regional Siebel Server.

**NOTE:** The Regional database server must be the same version as its parent database server.

Before you initialize the Regional Database, you must install the RDBMS software on the Regional Database Server to create an empty database with adequate space for data and index storage areas. Do this in accordance with the RDBMS vendor's documentation and the guidelines for configuration and space allocation. For more information, see the Siebel Server installation guide for the operating system you are using.

You must also create the database accounts for the Siebel Tableowner and Siebel Administrator and grant them the necessary privileges. Use the Application RDBMS Tool to create these users and to grant their privileges. For example, for an Oracle RDBMS, grant these accounts connect, resource, and DBA privileges. For MS SQL, the Tableowner needs to have Security Administrator privileges on regional nodes. For DB2, the Tableowner needs to have DBA privileges. For more information, see the Siebel Server installation guide for the operating system you are using.

**NOTE:** In order to successfully implement a srvrinit on an Oracle Regional Node, the table owner has to have certain privileges. This can be done without granting DBA privileges to the table owner. The procedure that follows describes how to enable srvrinit to work in Oracle without granting DBA privileges to the table owner.

### *To enable srvrinit in Oracle without granting DBA privileges to the table owner*

- Start SQL*Plus and log in as the user system or sys or a DBA user and then execute the following command:

```
grant sse_role, connect, alter user, create user, create table, create session to

<table owner> with admin option
```

  If only sse_role is granted, an error message will be displayed.

After the RDBMS has been installed and configured and you have created the accounts for the Siebel tableowner and Siebel administrator, installation of the Regional Database Server is complete. The database objects, such as tables and indexes, are automatically created when you extract and initialize the database. These procedures are described in "Extracting the Regional Database" on page 177 and "Initializing the Regional Database" on page 180.

**NOTE:** The Regional Database Initialization program uses the default storage parameters for the data and index tablespaces. Oracle database administrators should set storage parameters appropriately before installing the Regional Database. For information on setting storage parameters, see the Siebel Server installation guide for the operating system you are using. For larger tables, you can modify the storage parameters after the tables are created.

## Installing the Siebel File System

Each Regional Node requires a local Siebel File System. Attachment files in Siebel File Systems are replicated whenever the data rows with which they are associated are replicated. For information on creating the Siebel File System, see the *Siebel Installation Guide* for the operating system you are using.

The File System Manager (FSM) server component manages the File System and handles all requests for files that reside within the Siebel File System. FSM interacts directly with the Siebel File System to invoke requests for access to files. Most server components use FSM to access files by submitting requests to Server Request Manager. However, Siebel Remote components do not use FSM to access the File System. Instead, Siebel Remote components use Synchronization Manager to access the File System.

# Defining the Regional Node

Each Regional Node must be defined by registering the Regional Database and adding any mobile users for the database. After the Regional Node has been defined, you can extract the Regional Database on its parent node and initialize it.

**NOTE:** After the Regional Node is extracted, you can still continue to add Mobile Web Clients to it.

- If you are implementing a single tier of Regional Nodes, the Headquarters Node is always the parent node.
- If you are implementing two or more tiers of Regional Nodes (called a hierarchy of Regional Nodes), the lower tiers will have Regional Nodes as their parent nodes.

**NOTE:** Setup of parent nodes must be completed before you can define children for them.

Before you continue with the process of defining a Regional Node, you must finish the setup of your organization and territory structures and run the Assignment Manager on the Headquarters Node. If you make changes to the organization and territory structures after extracting and initializing the Regional Database, you may have to reextract the Regional Database on its parent node and reinitialize it. Otherwise, the transaction routers may be overloaded.

If you are implementing a Regional Node whose parent node is another Regional Node, you should synchronize the parent node with the Headquarters Node so that the parent node has up-to-date data. This reduces the amount of data to replicate after you initialize the Regional Node.

## Registering a Regional Node

To register a regional node, use the procedure that follows.

### *To register a new Regional Node*

1. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote.
2. In the link bar, select Replication Servers.
3. In the Parent Server form, select the appropriate node as the parent.

   Usually, this is HQ.

   The Parent Server form shows the registered databases, whether they are Headquarters or Regional.

   The Regional Databases list shows regional nodes that are children of a parent database currently selected in the Databases list applet. Only the Headquarters Database or a Regional Database can be the parent of another database; a Siebel Remote mobile database cannot be a parent.
4. In the Regional Databases list, click the menu button and select New Record.
5. In the Database Name field, enter a name for the Regional Database, such as SIEBEL_EUROPE.

   **NOTE:** The database name you specify can contain any alphanumeric characters (letters must be uppercase only), dashes (-), or underscores ( _ ). The name *cannot* include spaces, a period (.), or any special characters( / \ : * ? " < > |). The name cannot exceed 30 bytes (30 characters in a single-byte character set, or 15 characters in a double-byte character set). Three names are reserved and unavailable for use: TXNPROC, OUTBOX, and INBOX. The database name must be unique within the Headquarters Node and the Regional Nodes and need not match an existing database name.
6. In the Description field, type a description of the Regional Database, such as "European Regional Database."
7. Specify the Routing Group to which the Regional Database will belong.
   a. In the Routing Group field, click the select button.
   b. In the Pick Routing Group selection dialog box, select one of the following:
      - Regional Server – Full Copy
      - Regional Server – Standard

      The Full Copy option replicates all nonsystem data from the parent database to the Regional Database. Full Copy disregards routing rules. The Standard option uses routing rules to determine what data to replicate to the Regional Database. For more information about routing groups and routing rules, see "Routing Groups and Routing Rules" on page 164.

8 In the Database Users field, select the Siebel users who will work as connected users against the Regional Database.

**NOTE:** Every mobile user on the regional node must also be a connected user, and at least one connected user for the Regional Node must be specified before the regional database can be extracted successfully.

a In the Users field, click the select button.

b In the Users selection dialog box, click New.

c In the Add Users list, select the appropriate user.

d Click Add.

e Repeat the above steps to add all users to the regional node.

f Click Close.

If the Regional Database will support mobile users, implement the steps in "Adding Mobile Users to the Regional Node" on page 176 If the Regional Database will support only connected users, skip to "Extracting the Regional Database" on page 177.

# Adding Mobile Users to the Regional Node

If a regional node will support Siebel Remote mobile users, continue with the following procedure to add the mobile users to the regional node. If this regional node will support only connected users, skip this section and proceed to "Extracting the Regional Database" on page 177.

**NOTE:** Every mobile user on the regional node must also be a connected user.

### *To add mobile databases to a regional node*

1 From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote.

2 In the link bar, select Mobile Clients.

3 In the Parent Server list applet, select the Server Name representing the regional node where the mobile users will work.

If this were a follow-on from the example on the previous page, this would be the newly created regional node.

4 Follow Step 3 through Step 8 of the procedure, "To register a new Mobile Web Client on a node" on page 86.

5 Repeat Step 4, immediately above, for each mobile user assigned to the regional node.

**NOTE:** A mobile user can be assigned to only one Headquarters server or Regional node. Every mobile user must also be set up as a connected user in the Regional Node. Mobile Web Clients must be synchronized and then reextracted if they are to be reassigned to a different parent server.

# Extracting the Regional Database

Each Regional Database must be extracted on a Siebel Server of its parent node. The Regional Database synchronizes against the Siebel Server on which it is extracted. Database parallel extracts use multithreaded components to make these tasks faster than single-thread operations. Parallel dbxtract requires additional hardware, but can reduce the time to do this significantly.

This section describes the two types of dbxtract: single-thread and parallel. Before beginning either type of database extraction, you must synchronize the server components that are enabled. See *Siebel System Administration Guide* for details.

**CAUTION:** You must specify at least one connected user for the Regional Node before extracting the regional database, or the extraction will fail. For information about specifying connected users for a Regional Node, see "Registering a Regional Node" on page 175.

### *To extract database for Regional server with single thread – GUI*

1. From the application-level menu, choose Navigate > Site Map > Administration - Server Management > Jobs.
2. In the Jobs list, click New.
3. In the Component/Job field, select Database Extract.
4. In the Requested Server field, enter the name of the server on which you want the Database Extract job to run.

   **NOTE:** After the job is completed, the read-only Execution Server field displays the name of the server that ran the job. For a Database Extract Job, this is the same as the Requested Server.

   Complete the rest of the fields in the record as appropriate.
5. In the Job Parameters list, which is located below the Jobs list and the Job Detail form, click New and add the necessary parameters as described in the following substeps:
   a. In the Name field, click the select button to display the Job Parameters dialog box.
   b. Select Client Name and click OK to return to the main window.
   c. In the Value field of the Client Name job parameter record, enter the name of the regional node.
   d. Click New to create another new job parameter record.
   e. In the Name field, click the select button to display the Job Parameters dialog box.
   f. Select Database Init Method and click OK to return to the main window.
   g. In the Value field of the Database Init Method job parameter record, replace the default value with DDL.
6. Add any other job parameters as appropriate.

   For a list of parameters you can specify, see Table 27 on page 179.
7. In the Jobs list, with the Database Extract job selected, click Start.

   The Status field changes from Creating to Queued.

### *To extract database for Regional server with single thread - Srvrmgr command line*

- From the Srvrmgr command line, enter the following command all on one line:

```
start task for comp dbxtract server <server_name> with
client=<regional node name>, initmethod=ddl
```

The parallel dbxtract server component (pdbxtract) is designed to improve performance when extracting large databases for one or more Regional Nodes. It is also possible to use pdbxtract when extracting Mobile Web Client databases, but pdbxtract may not improve performance significantly during Mobile Web Client extractions, so the practice is not recommended. You can run pdbxtract as a component request, from within a Siebel application, or you can run it as a task from a srvrmgr command line. The following two procedures describe these two methods of using pdbxtract.

**NOTE:** If you plan to use pdbxtract for concurrent extraction of two or more Regional Node databases, be sure to specify all of the applicable Regional Nodes as clients in a single pdbxtract job or srvrmgr command. Due to the design of the pdbxtract component, this produces better performance than starting multiple pdbxtract component requests or tasks that attempt to run concurrently.

### *To parallel extract database for Regional server – GUI*

1. From the application-level menu, choose Navigate > Site Map > Administration - Server Management > Jobs.

   **NOTE:** Check the value of the parameter Maximum Number of WorkQ Threads from the Administration - Server Configuration > Components view. The Max Tasks field for the Parallel Database Extract component displays the maximum number of worker threads per work queue. This parameter determines the parallelism of the process. You may need to use the Columns Displayed command to view the Max Tasks field.

2. In the Jobs list, click New.
3. In the Component/Job field, select the Parallel Database Extract.
4. In the Requested Server field, enter the name of the server on which you want the Parallel Database Extract job to run.

   After the job is completed, the Execution Server field displays the name of the server that actually ran the job.

   Complete the rest of the fields as appropriate.

5. Perform the following steps to create Job Parameters records with the specified Names and Values:

   a. In the Job Parameters list, which is located below the Jobs list and the Job Detail form, click New.

   b. In the Name field, click the select button to display the Job Parameters dialog box.

   c. Select Client Name and click OK to return to the main window.

d In the Value field, enter the name of the regional node.

To specify more than one regional node, create a text file and place the name of each regional node on its own line of the file. Then, when specifying a Value for the Client Name parameter, enter the path and file name, preceded by the *at* sign (@). For example, a Windows client might specify @D:\workdir\regnodes.txt as the Value for Client Name.

e In the Job Parameters list, click New.

f In the Name field, click the select button to display the Job Parameters dialog box.

g Select Database Init Method and click OK to return to the main window.

h In the Value field, enter DDL.

6 Add any other parameters for the job as appropriate.

For a list of parameters you can specify, see Table 27 on page 179.

7 In the Jobs list, with the Parallel Database Extract record selected, click Start.

The Status field changes from Creating to Queued.

### *To parallel extract database for Regional server – Srvrmgr command line*

■ From the Srvrmgr command line, enter the following command, all on one line:

```
start task for comp pdbxtract server <server_name> with
client=<regional node name>, initmethod=ddl
```

To specify more than one regional node, separate the regional node names with commas. For example, if RN1, RN2, and RN3 are the names of three regional nodes, the client portion of the command would be entered as follows:

```
client=RN1,RN2,RN3
```

For information on running a Database Extract, see "Creating Mobile Web Client User Accounts and Privileges" on page 100.

Table 27 lists component parameters you can specify for the Database Extract task.

Table 27. Parameters for Database Extract

| Parameter | Name | Required /Optional | Description |
|---|---|---|---|
| CD Directory | CDDir | Optional | The directory on the machine of the parent node to which you write the extract files. |
| Client Name | Client | Required | The name of the Regional Database as entered when you registered the Regional Node. See "Registering a Regional Node" on page 175. |

Table 27. Parameters for Database Extract

| Parameter | Name | Required /Optional | Description |
|---|---|---|---|
| Database Init Method | InitMethod | Required | Specify DDL. You must set the Database Init Method parameter to DDL. Otherwise, the database initialization will fail. |
| Maximum data file size | DatFileSize | Optional | Sets the maximum size of a data file in megabytes. Minimum size is 1. Maximum size is 1,000. Default is 500.<br>For Parallel Dbextract recommend Minimum size 100 MB. |
| Language Code | Language | Optional | Extract messages for the specified language. The default is ENU, for U.S. English. |
| Specify the mobile client version of Siebel | ClientVersion | Optional | Specifies the client software version. Default is 2000 for v7.x. |

The following example shows a sample portion of the parent node's server directory tree after you run Database Extract tasks for Regional Nodes named SIEBEL_EUROPE and SIEBEL_PACIFIC:

```
SIEBEL
   DOCKING
      SIEBEL_EUROPE
         INBOX
         OUTBOX
      SIEBEL_PACIFIC
         INBOX
         OUTBOX
```

After you initialize a Regional Database, the Regional Node's directory structure will include similar folders for any additional Regional Nodes (given a hierarchy of Regional Nodes) and for any mobile users.

# Initializing the Regional Database

The Regional Database initialization process retrieves the database extract created in the previous step, and loads it into the Regional Database. Database objects are created and populated with data.

If you will be reinitializing a regional node with Mobile Web Clients, it is important to conduct the following test. Verify the Mobile Web Clients that synchronized with this node can reinitialize without deleting their local databases.

The Regional Database is initialized from the Siebel Server using the Regional Database Initialization program. This must be done on the Regional Server:

- In Windows, the Regional Database Initialization program is called srvrinit.exe. You must run srvrinit.exe from the DOS command prompt.

  **NOTE:** You must set the appropriate environment variables before running srvrinit.exe. Set environment variables by running the batch file `siebenv.bat`, located in the *siebel_server* directory. Run this batch file in the same command shell in which you will be running srvrinit.

- In UNIX, the Regional Database Initialization program is called srvrinit. You can run srvrinit from the command-line interface only.

In both operating systems, the Regional Database Initialization program is located in the bin subdirectory under the Siebel Server root directory.

In order for srvrinit to finish successfully, you need to make sure that sufficient transaction space (Rollback Segment) is available on the Regional Server Database. You may want to verify with your DBA and to increase or alter existing rollback segments if necessary.

You must also make sure that the combination of database user ID and password that you will use to run srvrinit exists on both the HQ Server and the Regional Node. Srvrinit only allows you to specify one set of credentials, but srvrinit is required to use those credentials for both machines. If credentials on the two machines are different, the process may not start, or may fail before completion.

If the Applications Server has multiple CPUs and disk arrays, running srvrinit in parallel can dramatically improve performance.

It is possible to reduce the size of the rollback segment required for srvrinit by reducing the dbxtract parameter Maximum data file size (DatFileSize). This is because the srvrinit task commits per file. Therefore the smaller the file, the smaller the rollback segment. See "Database Extract Parameters" on page 109 for details about DatFileSize.

### *To initialize the Regional Database in the GUI*

1. In the same window where you ran siebenv.bat, navigate to <server install on the regional node>\bin and enter `srvrinit.exe`.

   The Regional Database Initialization dialog box appears.

2. In the Siebel Remote Parameters fields, specify the correct values for each of the parameters used by the Regional Siebel Server to connect to the parent node. For a description of these parameters, see Table 28 on page 183.

3. In the Regional Database Parameters fields, specify the correct values for each of the parameters used by the Regional Siebel Server to connect to the Regional Database. For a description of these parameters, see Table 28 on page 183.

4. To start the initialization process, click Start Initialization.

   The database extract will be downloaded from the server. The Siebel Upgrade Wizard launches automatically and completes the steps required to initialize the Regional Database. If you do not see a dialog box on the screen as the process executes, usually this indicates something is wrong. If this happens, check the log file in the log directory.

   There will be two log files. The first one is srvrinit.log. The second one is either upgwiz.log on Windows or srvrupgwiz1.log on UNIX.

After the initialization completes successfully, setup of the Regional Database is complete. For instructions on administering the Regional Database, see Chapter 11, "Administering Siebel Replication Manager."

The Regional Database Initialization program (srvrinit) has two functions:

- It connects to the parent remote server and downloads the database extract for the regional node.
- It starts the Siebel Upgrade Wizard to carry out the actual database initialization.

### *To initialize the Regional Database from the command line interface*

- Change to the bin subdirectory under the Siebel server root directory and enter:

  ```
  srvrinit flags
  ```

  For a list of flags, see Table 28 on page 183.

**NOTE:** If the Regional Database initialization fails (for example, due to insufficient storage space on the database), you must recover from the failure before restarting Replication Manager. To do this, restart the Regional Database Initialization program.

The built-in state logging ability of the Siebel Upgrade Wizard provides the ability to restart the Siebel Upgrade Wizard. If you encounter an error during the second phase (Siebel Upgrade Wizard) of the regional database initialization process, examine the log file (UpgWiz.log on Windows and srvrupgwiz1.log on UNIX) and take appropriate corrective actions.

**NOTE:** The following error messages can be ignored: ORA-01921: role name 'SSE_ROLE' conflicts with another user or role name, ORA-01017: invalid username/password; logon denied, and ORA-00942:table or view does not exist.

After applying the proper fix, simply restart the regional initialization process by running Siebel Upgrade Wizard again. The Siebel Upgrade Wizard remembers the progress of the initialization process and begins from where it left off. The executable name is siebupg.exe on Windows and srvrupgwiz on UNIX.

However, there may be occasions when you simply want to restart the Regional Database Initialization process from scratch. To do so, follow the procedure below.

### *To restart Regional database initialization process from scratch – Windows*

1. On the Regional Node, navigate to $SiebelRoot/bin.
2. Run the Siebel Upgrade Wizard (type the command or double-click siebupg1.exe).
3. Click Cancel when you are prompted to choose Yes to retry, No to abort, or Cancel to abort and clean up.
4. Rerun the initialization process.

### *To restart Regional database initialization process from scratch – UNIX*

1. On the Regional Node, navigate to $SiebelRoot/bin.

2 Find the file upgwiz.ucf and delete it.

3 Navigate to $SiebelRoot/upgrade and delete all files with the name state.log under it.

4 Rerun the initialization process.

Table 28 on page 183 lists selected parameters and corresponding flags for the srvrinit command-line utility.

Table 28. Parameters and Command-Line Flags for srvrinit Utility

| Parameter | Command Line Flag | Description | Required? |
|---|---|---|---|
| 16K Table Space | /ks | For DB2/UDB installations only, the name of the appropriate 16K Table space should be provided. For more information, see the Siebel Server installation guide for the operating system you are using. | No |
| 32 K Table Space | /ls | For DB2 installations only, the name of the appropriate 32K Table space should be provided. For more information, see the Siebel Server installation guide for the operating system you are using. | No |
| N/A | /comm | Communication parameters for modem connection. | No |
| Parent AppServer Name | /dockconnstring | Name of the Siebel server at the parent node where the database extract was performed. | Yes |
| Regional Server Name | /n | Name of the Regional Node. | Yes |
| File System | /filesystem | File system directory on the regional node. | Yes |
| Help | /?<br>/help | Help on usage. | No |
| Index Space | /is | For DB2/UDB installations only. Space on the Regional Database where indexes should be created. | No |

Table 28. Parameters and Command-Line Flags for srvrinit Utility

| Parameter | Command Line Flag | Description | Required? |
|---|---|---|---|
| Initialize on new database | /init | Indicates whether the regional database is empty or not. Use Y if this is the initialization of the regional server on new database, N if it is an upgrade of the existing regional database with new dbxtract. Default value is N.<br><br>Parameter should be set to Y during a major upgrade—old transactions can be invalid for the new schema and cause failure. | No |
| N/A | /l | Language code, default is enu. | No |
| ODBC Data Source | /d | Name of ODBC data source to connect to the Regional Node. | Yes |
| User Password | /p | Password to authenticate the Regional Database against the Headquarters Server. The password you specify will be used when the users are set up on the Regional Database. | Yes |
| Repository Name | /reposname | Name of repository, usually "Siebel Repository." | Yes |
| Run in Parallel/ Number of Threads | /para | Indicates srvrinit will run in parallel and the number of threads. | No |
| Schema Qualifier | /sq | For DB2/390 and AS/400 environments only, the name used to qualify all database objects created that are required by Siebel. | Yes |
| Server Root Directory | /homedir | Server root directory, on the Regional node where *SIEBEL_SERVER_ROOT is the root directory of the regional Siebel Server installation.* | Yes |
| Table Grouping File | /tg | For DB2/390 installations only. Full path to the storage control file, which is a file that contains database object declarations. | Yes |

Table 28. Parameters and Command-Line Flags for srvrinit Utility

| Parameter | Command Line Flag | Description | Required? |
|---|---|---|---|
| Table Owner Password/Privileged User Password | /tp | Tableowner's password on the Regional Database.<br><br>For DB2/390 and AS/400 environments, this is the same as the Privileged User Password. This is the password for Privileged User ID. | Yes |
| Table Owner/ Privileged User ID | /t | Tableowner's logon on the Regional Database. Must have privileges to create database objects.<br><br>For DB2/390 and AS/400 environments, this is the same as the Privileged User ID. This is a user account that has the necessary database authority and privileges to create, access, and modify Siebel database objects as well as native database objects and operations required to implement the Siebel application. These environments have rigid controls on user identification—accounts must correspond to a real person. | Yes |
| Table Space | /ts | For DB2/UDB installations only. Space on the Regional Database where tables should be created. | No |
| User Name | /u | User name to authenticate the Regional Database against the Headquarters Server. You can use any database user associated with the Regional Database during the setup, but the user must have security admin privileges in order to create logins within the master database (such as sadmin). | Yes |
| N/A | /v | Verbose mode, may be set to Y or N. Default is N. | No |

After completion of initialization, enable the required server components such as Siebel Remote and the appropriate object managers. For information regarding how to do this for Remote Server components, see "Starting and Stopping Siebel Remote Server Components" on page 126. For information regarding how to do this for object managers and other components, see *Siebel System Administration Guide*.

Also, administrators need to install the Siebel Web Server Extension (SWSE). For information regarding how to do this, see the Siebel Server installation guide for the operating system you are using. This is essential for administrators to use the Web client to support users on Regional Nodes.

The SWSE is not required if administrators use the dedicated Web client connected to support users on Regional Nodes.

**NOTE:** Administrators must create users (database accounts or others) for all users who will be accessing the Regional Node because srvrinit will no longer create user accounts.

# Configuring Regional Node for Mobile Web Clients

If your Regional Server has Mobile Web Clients, you must start and configure Siebel Remote Server components on the Regional Server. Perform these steps in the Siebel Server Manager.

### *To configure Regional Node to support Mobile Web Clients*

1. Use the Generate New Database component to generate database templates on the Regional Server.

   For information on generating a new database template, see "Generating a New Database Template" on page 67.

2. Run Database Extract to extract Mobile Web Clients.

   For information on running a Database Extract, see "Creating Mobile Web Client User Accounts and Privileges" on page 100.

3. Start the Transaction Processor, Transaction Merger, and Transaction Router components.

   For more information, see "Starting and Stopping Siebel Remote Server Components" on page 126.

# Starting Replication Agent

Beginning with Siebel version 7.5, Replication Agent starts 0 tasks (default setting) when Siebel Server starts. If you want to start Replication Agent tasks automatically when Siebel Server starts, see *Siebel System Administration Guide* for information on how to change the Default Processes parameter to 1.

The procedures below describe how to start Replication Agent while Siebel Server is running.

### *To start Replication Agent – GUI*

1. If the default task is 0, reconfigure the Replication Agent component using the following Component Reconfiguration steps.
   a. Before reconfiguring server components, verify that any external resources remain available that will be accessed by current tasks running on the existing component configuration. Also verify that any external resources are available that will be defined in the new component configuration.

      Examples of external resources include the SRF and CFG files that are defined in the component configuration.

   b. From the application-level menu, choose Navigate > Site Map > Administration - Server Configuration > Enterprises.

   c. Click the Component Definitions view tab.

   d. In the Component Definitions list, select the component definition you want to reconfigure (Replication Agent, in this case).

   e. Click the menu button and then click Start Reconfiguration.

      The State field changes to display the value Reconfiguring.

   f. In the Component Parameters list, change the Value field of parameters that you want to reconfigure for the component—in this case, change the value of Default Tasks to 1.

      You can also change the values of fixed parameters, but you cannot change whether parameters are fixed.

   g. After parameter values have been reconfigured, commit the new configuration by clicking the menu button for the Component Definitions list and clicking Commit Reconfiguration.

      The new parameter values will be merged at the enterprise level. To cancel the reconfiguration before it has been committed, click the menu button and then Cancel Reconfiguration.

2. If the default task is 1, then, from the application-level menu, choose Navigate > Site Map > Administration - Server Management > Components.
3. Start Replication Agent.

For more information about this topic, see *Siebel System Administration Guide*.

### *To start Replication Agent – srvrmgr command line*

- From the srvrmgr command line, enter the following command all on one line:

```
start task for comp repagent server <server_name> with HQ=<DockConnString>,
sleeptime=<#sec>,
```

  Sleeptime is set to 60 sec by default, so it is not necessary to use.

# Setting Up Additional Application Servers on Regional Nodes

This section describes the procedures to add Application Servers on the Regional node. The first procedure will add an additional Applications Server. Srvrinit can only be invoked once and only one App Server receives the dictionary files. These files are placed in the App Server bin directory. The second App Server is not functional because it does not have a dictionary datafile. The regional server does not have repository rows, so the dictionary information is retrieved from this data file.

### *To install additional Application Servers on the Regional node*

- For the necessary steps to do this, see the Siebel Server installation guide for the operating system you are using.

As stated above, the additional server requires a copy of the dictionary file to become operational. There are several options to copy a dictionary file from one App Server another on a Regional node.

### *To copy the data file to a new Applications Server bin directory from another*

- Copy the necessary dicdata.dat from the initial Application server where srvrinit was run.

  **NOTE:** The dictionary data files are character-set-specific, so you need codepage dictionary caches.

# 11 Administering Siebel Replication Manager

This chapter describes registering users on the regional node in addition to managing synchronization and server processes. It also includes a discussion about backup and recovery of data.

To administer Siebel Replication Manager you must implement four main phases:

- Registering Users on a Regional Node
- Managing synchronization
- Managing server processes
- Performing backup and recovery of data

Administering the four phases of Siebel Replication Manager consists of the following tasks, shown in Table 29.

Table 29. Tasks for Administering Siebel Replication Manager

| Task | Where Performed |
|---|---|
| Managing Mobile Users on a Regional Node on page 189 | HQ Node |
| Adding Connected Users to a Regional Node on page 190 | HQ Node |
| Managing Synchronization on page 191 | Regional Node |
| Changing the Routing Group for a Regional Server on page 194 | HQ Node |
| Monitoring Regional Application Servers on page 195 | Regional Node |
| Managing Security and Authentication on page 195 | Regional Node |
| Performing Backup and Recovery of Data on page 195 | Regional Node |
| Managing Backlog in the Transaction Log Table on page 196 | HQ Node |
| Deactivating and Reactivating a Regional Node on page 196 | Parent Node |

# Managing Mobile Users on a Regional Node

If you need to add Mobile Web Clients on the regional database where they work, use the following procedure.

### *To add mobile users to a Regional Database*

■ See "Adding Mobile Users to the Regional Node" on page 176.

If you need to delete Mobile Web Clients on the regional database where they work, use the following procedure.

**NOTE:** Mobile users should be deactivated before deleting them. For more information about ending mobile users, see "Deactivating and Reactivating a Mobile Web Client" on page 142.

### *To delete mobile users from a Regional Database*

1. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote.
2. In the link bar, select Mobile Clients.
3. In the Parent Server list, select the Server Name representing the regional database where the Mobile Web Client works.
4. In the Mobile Clients list, select the record representing the Mobile Web Client that you want to delete from the Regional Database.
5. Click the menu button and then Delete Record.
6. In the dialog box, confirm that you want to, or do not want to, delete this record.

   Repeat Step 4 through Step 6 for each Mobile Web Client you want to delete from the Regional Database.

# Adding Connected Users to a Regional Node

This section describes how to add users to the Regional Node. Before adding users to a Regional Node they must be added to the Parent Node. See *Applications Administration Guide* for adding users to the system.

### *To add connected users to a Regional node*

1. Make sure the user is a connected user at the HQ node.
2. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote.
3. In the link bar, select Replication Servers.
4. On the Parent Server form, select the parent server of the appropriate regional node.

   This may be HQ or a regional node that is also a parent node.
5. In the Regional Databases list, select the Regional Database you desire.
6. On the Users field, click the select button.
7. In the Database Users Selection Dialog box, click New and select the user you want, and then click Add.

8 Repeat Step 7 until you have added all necessary users, then click OK.

9 Create user access accounts on the regional node (db accounts or other external Directory Services such as LDAP). For more information about this topic, see *Security Guide for Siebel eBusiness Applications*.

Wait for a few hours (depending on your network and hardware configurations) to make sure that all new user data is replicated to the regional node.

# Managing Synchronization

The Siebel Server component Replication Agent, which operates on the Regional Siebel Server, synchronizes the Regional Database with its parent database.

Replication Agent must be started in order to synchronize the Regional Database. You can control Replication Agent with either the GUI or the command-line version of the Siebel Server Manager.

For information about starting Replication Agent from the GUI, see "Starting Replication Agent" on page 186. The following procedures describe how to start and stop Replication agent using srvrmgr.

### *To start Replication Agent – srvrmgr command line*

- From the srvrmgr command line, enter the following command all on one line:

```
start task for comp repagent server <server_name> with <parameter1>= <value1>,
<parameter2=<value2>,...
```

Values are from "Parameters for Replication Agent" on page 192.

### *To stop Replication Agent - srvrmgr command line*

- From the srvrmgr command line, enter:

```
stop task for comp repagent server <server_name>
```

## Resolving Synchronization Conflicts

There are times when Replication Manager will encounter data conflicts or corrupted transactions. Resolution rules may help to resolve these.

For further information on conflict resolution rules, refer to "Conflict Detection and Resolution" on page 44. There is currently no log file available.

## Replication Agent Parameters

Table 30 lists the parameters for Replication Agent. Replication Agent can also use generic parameters described in *Siebel System Administration Guide*. Also, refer to the same guide for more information about managing server components with Siebel Server Manager.

Table 30. Parameters for Replication Agent

| Parameter | Meaning | Required/ Optional | Default Value and Usage Notes |
|---|---|---|---|
| HQ | The Dock Connect String of the parent Siebel Server. | Required | The format is the same as for the Dock Connect String:<br>*machine_name:network_protocol:sync_port:service:encryption*<br>where *machine_name* identifies the parent Siebel Server and *sync_port* identifies the synchronization port number.<br>Available choices are:<br>*network_protocol:* TCPIP<br>*sync_port:* 40400, Any<br>*service:* SMI<br>*encryption:* None, MSCRYPTO, RSA<br>Default for Sync Port: 40400<br>Default for Encryption: None |
| Comm Param | The colon-separated list of communications parameters used to connect to the parent Siebel Server. | Optional | This parameter is not used. |
| SleepTime | The period (in seconds) for which Replication Agent should sleep between synchronization sessions. | Optional | Default: 60 |
| MaxWrite | Used to limit the size of a DX file—that is, to break DX files to be sent to the parent Siebel Server into smaller, more manageable pieces.<br>Replication Manager closes the existing DX file and starts writing to a new DX file when the current DX file reaches the maximum number of operations specified with this parameter. | Optional | Default: 10000 |

Table 30. Parameters for Replication Agent

| Parameter | Meaning | Required/ Optional | Default Value and Usage Notes |
|---|---|---|---|
| SendTxns | Sends DX files from the Regional Server to the parent Siebel Server.<br><br>If this parameter is set to FALSE, Replication Manager does not send DX files from the Regional Server to the parent Siebel Server. | Optional | Default: TRUE |
| RecvTxns | Receives DX files sent by the parent Siebel Server to the Regional Server.<br><br>If this parameter is set to FALSE, Replication Manager does not receive DX files sent by the parent Siebel Server to the Regional Server. | Optional | Default: TRUE |
| RecvFiles | Receives file attachments sent by the parent Siebel Server to the Regional Server.<br><br>If this parameter is set to FALSE, Replication Manager does not receive file attachments sent by the parent Siebel Server to the Regional Server. | Optional | Default: TRUE |
| Iterations | Specifies the number of times that Replication Agent will connect to the parent node.<br><br>Typically, this parameter is set to zero (0), and the Replication Agent loops forever (or until stopped by the server administrator).<br><br>If you set this parameter to 1 and start Replication Agent, this causes a single synchronization to occur. | Optional | Default: 0 |
| MaxCommitTxns | Specifies how many transactions will be applied before they are committed to the database. | Optional | Default: 100 |

The Replication Agent runs as a service-mode task, which means that after it is started, it runs continuously.

# Changing the Routing Group for a Regional Server

Each regional server has a routing group associated with it. The group determines the data that will be extracted to and transactions that will be synchronized with the regional server.

There may be times when you need to change the routing group for a regional server. The procedure below describes how to do this.

**CAUTION:** Changing the routing group from "Standard" to "Full Copy" requires a reextract of the regional database. Also any Mobile Web Clients associated with the Regional Node require a reextract as well.

### *To change the routing group for a regional server*

1. If the Regional Node has Mobile Web Clients, synchronize these with the Regional Node.
2. Send updates in the regional node to the HQ or parent node.
3. Stop the activities on the regional node.

   This means no more updating.
4. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote > Replication Servers.
5. In the Parent Server form, select the appropriate node as the parent.

   Usually, this is HQ.
6. In the Regional Databases list, select the appropriate Server Name.
7. Click the select button in the Routing Group field.
8. In the Routing Model Selection dialog box, select the new Routing Group you want and click OK.

   Fill in any of the other fields on the form as appropriate.
9. Perform a database extract of the regional database, then initialize the regional database.

   For extracting the regional database, see "Extracting the Regional Database" on page 177.

   For initializing the regional database, see "Initializing the Regional Database" on page 180.
10. Perform database extract of Mobile Web Clients on the new regional node, and resume normal operations on the regional node.

    For extracting mobile databases, see "Creating Mobile Web Client User Accounts and Privileges" on page 100.

**NOTE:** All views are supported only on Regional Nodes that have a Routing Group of type REGIONAL SERVER - FULL COPY.

# Monitoring Regional Application Servers

Administrators should use the Administration - Siebel Remote and Administration - Server Management views to monitor server components and data movement from regional nodes to the parent node.

Administrators should also monitor disk space availability.

# Managing Security and Authentication

Replication Manager uses the administrator's user name and password to log on to the Parent node. During this process, it authenticates against its parent node, usually the HQ node. It does this according to the Authentication Method set in the Synchronization Manager Component of the parent node. For more information about the Authentication Method parameter, see Table 9 on page 64.

It is recommended that the administrator user name and password be the same on the HQ database and the regional databases. The Regional Application Servers and srvrinit use the administrator database login.

### *To change the administrator password for Replication Manager*

1. Change the password according to the Authentication Method.

   For example, if the Authentication Method is set to Database, changing the password in the database on the Parent node requires changing the same password on the Regional Node. For information about this topic, see *Security Guide for Siebel eBusiness Applications*.

2. Change the password by using the Siebel Application Server on the Regional node.

   a. From the application-level menu, choose Navigate > Site Map > Administration - Server Configuration > Enterprises > Parameters.

   b. In the Enterprise Parameters list, enter an appropriate value for the Password parameter.

   c. Stop the Siebel application services.

      For more information on how to stop and start Siebel services, see *Siebel System Administration Guide*.

   d. Restart the Siebel application services for the changes to take effect.

# Performing Backup and Recovery of Data

Database administrators should perform regular backups and maintenance tasks on the Headquarters Database and on the Regional Databases. If a failure occurs on a Regional Database and it cannot be recovered to the exact point of failure, you must reextract the Regional Database on the Database Server at its parent node.

If the Regional node can be recovered to the exact point of failure, the S_DOCK_STATUS values of the HQ and regional nodes will be in sync. After recovery to the point of failure, the following transactions will be synchronized: (1) those that were not synchronized at the time of failure and (2) those created on the HQ and Regional nodes after the time of failure.

If transactions are lost on the HQ node, the child nodes and subsequent child nodes must be reextracted. You must also reextract the mobile users who synchronize with the Regional Siebel Server whose Regional Database failed.

# Managing Backlog in the Transaction Log Table

It is best practice to make sure the transaction processors that are not required are end dated.

It is very important to make sure old transaction processors have end dates. For example, this may happen when a transaction processor is started and then not used on a server, postupgrade. Or, this may happen when a transaction processor is started and then the server is uninstalled.

### *To end date an old transaction processor*

1. From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote > Processor Status.
2. In the Transaction Processors list, select the old transaction processor.
3. In the End Date field, enter an end date.

   For more information about on this topic, see "How to Handle a Large Transaction Backlog" on page 213.

# Deactivating and Reactivating a Regional Node

This section describes the procedure to deactivate and reactivate a Regional Node.

**CAUTION:** Make sure to do a full backup of the HQ Node and the Regional Nodes before deactivating a regional node.

### *To deactivate a regional node*

1. Synchronize all remaining transactions of remote users assigned to the Regional Node (if any).
2. Make sure all transactions have been applied to the Regional Node (for example, make sure that Txn Merger on the Regional Node has processed the .dx files).
3. Check that the transactions have been synchronized with the HQ node.
4. Move every Remote user attached to your Regional Node to either the HQ Node or another Regional Node (if any).

5 Shut down the Transaction Processor, Router, Merger and RepAgent on the Regional Node.

6 Shut down the Transaction Processor and Router on the HQ Node.

7 On the HQ Node, set the Effective End Date for the Regional Node in Administration - Siebel Remote > Replication Servers.

8 Restart the Transaction Processor and Router on the HQ Node.

9 Point the Regional Node connected users to HQ by modifying ODBC sources and the CFG file.

10 Delete the docking folder for the Regional Node from the Siebel Server\DOCKING directory on the HQ Node as it is no longer needed.

### *To reactivate a Regional node*

1 Shut down the Transaction Processor and Router on the HQ Node.

2 On the HQ Node, set the Effective End Date to NULL in Administration - Siebel Remote > Replication Servers for this Regional Node, on HQ.

3 Reextract and reinitialize the regional database.

# 12 Upgrading Regional Nodes

This chapter describes upgrading regional nodes with and without the use of Siebel Anywhere. It also discusses schema upgrades in the DB2 regional environment.

## Upgrading the Repository for a Regional Node

A Siebel repository contains the object definitions of the Siebel application and database schema. Within any one hierarchy, every instance of Siebel Replication Manager and Siebel Remote must have the same database schema and SRF file.

**NOTE:** Although the Language can vary for each instance within any one hierarchy, the SRF file must be compiled from the same repository. For example, the SRF file used in a Replication environment can be a different language from the SRF file used in the HQ environment if both are compiled from the same repository. Similarly, the Language used by one Remote client can be different from another Remote client within the same hierarchy if each is compiled from the same repository.

When object definitions are changed—for example, to change application behavior or to add extension columns—it is necessary to distribute the changes to the Headquarters, Regional and local databases (this is known as a minor upgrade).

The application definition is compiled into an SRF file using Siebel Tools. Schema object changes are represented by a set of DDL (Data Definition Language) operations.

There are two ways to roll out repository changes to a regional environment:

- If you have Siebel Anywhere, you can use it to apply schema changes to regional and mobile users. For detailed steps, see "Repository Upgrade with Siebel Anywhere" on page 199.
- If you do not have Siebel Anywhere, you must reextract and reinitialize the regional and mobile nodes after applying the repository changes and synchronizing with the physical schema. For detailed steps, see "Repository Upgrade Without Siebel Anywhere" on page 200.

**NOTE:** It is recommended that you thoroughly test the repository changes and upgrade steps in a separate test environment before migrating them to production. The following steps assume that an SRF file has been compiled and is available for distribution.

## Repository Upgrade with Siebel Anywhere

This section describes upgrading a repository by using Siebel Anywhere.

***To prepare to upgrade a repository***

1. Have the mobile users synchronize with the Regional Database.

   **NOTE:** After synchronizing, mobile users should not make any changes to their local database until after the upgrade. Changes made after this point will be made with the old repository, schema, and SRF file, and so may cause problems when they are applied to the upgraded regional server.

2. Wait until every transaction is applied to the Regional Node.
3. Run Replication Agent to synchronize the Regional Node with the Parent Node.

   This is usually the Headquarters Node.

   **NOTE:** No more changes should be made to the Regional Node until after the upgrade. Changes made after this point will be made with the old repository, schema, and SRF file, and so may cause problems when they are applied to the Parent Node.

4. Wait until the transactions are applied to the Parent Database.
5. Disconnect the users and stop the server components as appropriate. Note that not all upgrades will require this step.
6. Upgrade the HQ Node.

   For more information on installing a new repository, see *Siebel Installation Guide* for the operating system you are using.

7. Restart server components and make the system available to users.

***To build the upgrade kit for the database schema***

- Refer to *Siebel Anywhere Administration Guide* for the necessary procedures.

***To distribute the upgrade kits and finish the upgrade***

- Refer to *Siebel Anywhere Administration Guide* for the necessary procedures.

# Repository Upgrade Without Siebel Anywhere

This section describes upgrading a repository without the Siebel Anywhere option.

***To upgrade a repository without Siebel Anywhere***

1. If the Regional Node has mobile users, instruct the mobile users to synchronize with the Regional Node and to stop working on their local databases until further notice.

   **NOTE:** After synchronizing, mobile users should not make any changes to their local databases until after their databases have been reinitialized. Changes made after this point will be lost.

2. Wait until the transactions have been applied to the Regional Node.

3 Run Replication Agent to synchronize the Regional Node with the Parent Node. This is usually the Headquarters Node.

   **NOTE:** No more changes should be made to the Regional Node until after it has been reinitialized. Changes made after this point will be lost.

4 Wait until the transactions are applied to the Parent Node.

5 If appropriate, disconnect the connected users on the Parent Node.

6 Stop the server components as appropriate.

7 Apply the upgraded repository to the Headquarters Node and synchronize the repository with the physical schema.

8 Allow connected users at the Headquarters Node to reconnect and continue working using the new SRF file.

9 Use Generate New Database to generate a new database template on the Parent Node.

10 On the Parent Node, perform a database extract of the Regional Node.

11 Initialize the Regional Node.

12 Allow connected users at the Regional Node to reconnect and to resume work using the new SRF file.

13 Generate a new database template on the Regional Node.

14 Extract each Mobile Web Client on the Regional Node.

15 Allow mobile users to reinitialize their local databases and to resume work using the new SRF file.

# 13 Troubleshooting Remote and Replication Manager

This chapter includes reference information, FAQs, and scenarios for Remote and Replication Manager to help resolve problems. The material discusses components, administrative tasks, and specific situations.



## Setting Up Remote Server and Remote Clients

This quick reference sequence provides an overview of the steps to set up Siebel Remote on both the server and the Mobile Web Client. The purpose is to provide a "quick look" so you can review or learn the material.

The following sequence of steps outlines the general flow to set up Siebel Remote Server and Remote Clients for an out-of-the-box installation of your Siebel application.

1. Make sure employees are set up; from the application-level menu, choose Navigate > Site Map > Administration - User > Employees.

   The employee must have at least one position and responsibility. The fields cannot be blank.

2. If your deployment includes regional nodes, define the appropriate employees as users on their corresponding regional nodes.

   For details on how to do this, see "Registering a Regional Node" on page 175.

3. Set up users as Mobile Web Clients: from the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote > Mobile Clients.

   a. Add Mobile Web Clients here.

   b. At a minimum, fill in the fields Mobile Client, User Login Name and Routing Model.

      For example: HALACON, HALACON, MOBILE CLIENT - STANDARD

4. Start a "Generate New Database" process. Wait for it to finish successfully.

5. If the Siebel Server has not had Mobile users set up on it before, start a "Database Extract Process" for the Mobile Web Clients.

   a. Enter * in the Client Name field in the Parameter Overrides view.

   b. Wait for this task to finish successfully.

6. If the transaction processor is not currently running, start a transaction processor process.

   For details to start a transaction processor component, see "Starting Siebel Remote Server Components" on page 58.

7. If the transaction router process is not currently running, start a transaction router process.

   For details to start a transaction router component, see "Starting Siebel Remote Server Components" on page 58.

8. If the transaction merger process is not currently running, start a transaction merger process.

   For details to start a transaction merger component, see "Starting Siebel Remote Server Components" on page 58.

9. Make sure that you set up the Mobile Web Client hardware and software.

10. Make sure the appropriate CFG files for your Mobile Web Clients are correctly set up.

    Especially verify the contents of the section titled [Local]. Values of CHANGE_ME are indicators that these values probably require changing.

11. Run a Mobile Web Client session on a mobile user's PC and attempt to connect to the local database.

    It will not exist yet and will require initialization of the local database.

    If this step is successful, it means the mobile user is set up and initialized correctly.

# About Merge Conflicts Related to Assignment Manager

If the LogTxnChgOnly (Log transaction on change only) parameter for Assignment Manager is set to True, your Siebel Remote implementation may log an unusually high number of merge conflicts. You can safely ignore many of these merge conflicts. The following paragraphs describe why this phenomenon occurs, and provide an example of a sequence of events that produces such merge conflicts. For information about how you can distinguish between harmless merge conflicts and merge conflicts that do concern your data, see "Distinguishing Between Harmless and Meaningful Merge Conflicts" on page 205.

## Why LogTxnChgOnly Affects the Quantity of Merge Conflicts

When LogTxnChgOnly is set to True, Assignment Manager does not log transactions for changes that only affect the ASGN_DT field for a record. The ASGN_DT field records the most recent date and time that Assignment Manager assigned that record. This field is not normally visible in Siebel applications.

Since transactions are not logged when only the value of the ASGN_DT field changes, these changes are not sent to Mobile Web Clients. Not sending these changes causes a discrepancy between the version of the record stored on the server and the versions stored in Mobile Web Client databases. The discrepancy causes no immediate problem, since it does not affect any of the visible data fields for the record. Allowing such harmless discrepancies can vastly reduce the amount of data that must be transferred to a Mobile Web Client during synchronization.

However, if Assignment Manager updates *visible* data fields in the record at a later time, a transaction *is* logged. In this case, the discrepancy in ASGN_DT field values is detected during the Mobile Web Client's next synchronization attempt, and is reported as a merge conflict.

### Example of a Sequence of Events that Produces a Harmless Merge Conflict

The following paragraphs describe an example of a sequence of events that produces a harmless merge conflict. For clarity, the value of the ASGN_DT field is shown here as a date only, although the field actually includes both date and time.

1. The server runs Assignment Manager and changes the value of several fields in record X, including setting the value of field ASGN_DT to 2003-10-29. Because values are changed in one or more visible fields, a transaction is logged.
2. A Mobile Web Client synchronizes and receives the updated values for all fields in record X, including the value of 2003-10-29 for the ASGN_DT field.
3. The server runs Assignment Manager and changes the value of the ASGN_DT field to 2003-10-30, but does not change values in any visible fields, so no transaction is logged. The change to the value of the ASGN_DT field is not transmitted to the Mobile Web Client.
4. The server runs Assignment Manager and changes the value of several fields in record X, including setting the value of field ASGN_DT to 2003-10-31. Because values are changed in one or more visible fields, a transaction is logged.
5. The Mobile Web Client synchronizes and receives the updated values for all fields in record X, including the value of 2003-10-31 for the ASGN_DT field. This causes a conflict, because the transaction updates the value of the ASGN_DT field from 2003-10-30 to 2003-10-31, but the current value of ASGN_DT in the local database is 2003-10-29. The old value in the transaction does not match the current value in the local database, so a conflict is reported.

## Distinguishing Between Harmless and Meaningful Merge Conflicts

The following procedure describes how to distinguish between harmless merge conflicts caused by changes to ASGN_DT fields, and conflicts that do involve your data. For an explanation of the cause of merge conflicts involving ASGN_DT fields, see "About Merge Conflicts Related to Assignment Manager" on page 204.

### *To distinguish between harmless and meaningful merge conflicts*

1. Using the Mobile Web Client, from the application-level menu, choose Navigate > Site Map > User Preferences > Remote Status.

2 In the Remote Status list, select a synchronization session record that interests you.

3 In the Session Actions list, select a record that has the following value in the Item Name field:

Apply database changes

4 In the Session Action Details list, inspect the messages in the Item Details field.

Table 31 describes how to interpret these messages.

Table 31. Interpretation Guidelines for Synchronization Conflict Messages

| Item Details Message Characteristics | Guideline |
|---|---|
| Identifies who updated the record. For example, "Updated by HKIM."<br>Identifies a visible data field. For example, "Field: Product Under Warranty Flag, New Value: Y, Old Value: N" | Meaningful. Inspect server and client values to determine whether conflict was resolved appropriately. |
| Does not identify who updated the record. For example, "Updated by ?"<br>Identifies an Assignment Date field. For example, "Field: Assignment Date, New Value: 2003-11-04 11:09:18.000000, Old Value: 2003-10-25 10:10:02.000000" | Safe to ignore. |

# Troubleshooting Synchronization Manager

This section describes how to analyze selected problems that can be encountered with Synchronization Manager, plus possible causes and workarounds.

## SynchMgr Situation

The following error was encountered in SynchMgr_xxx.trc/log files:

- DCK-00123: Error opening file (null) for read

```
[ERR33] (drl.cpp 5(206) err=1700123 sys=1400022) DCK-00123: Error opening file
d:\siebfile\S_DOC_PPSL_0-CQNE_0-S9.saf for read
```

Possible Causes:

- Unable to access the File System directory.

- File Attachments do not exist in the File System.

  If this occurs with only one particular Mobile Web Client, make sure the System DSN is set up correctly.

  Verify that the attachments are available in the file system. Siebel eBusiness Applications come with a set of default templates. Make sure you have copied the files from the <dbsrvr>\files to the Siebel File System.

- DCK-00164: Error connecting to datasource (null) ((null))

  Possible Causes:

  - Siebel Gateway and Services were started while database server was shut down.

  Workaround:

  - From the application-level menu, choose Navigate > Site Map > Administration - Server Management > Components, shut down the Synchronization Manager and then restart it. Refresh the applet to make sure that Synchronization Manager has a state of running.

    For more details about Synchronization Manager, see *Siebel System Administration Guide* and "Starting Siebel Remote Server Components" on page 58. SynchMgr_xxx contains the following error message: (syncsrvr.cpp 22(692) err=1700213 sys=0) DCK-00213: Another Synch Server is already servicing this node.

  Possible Causes:

  - Interrupted synchronization sessions. If a client synchronization session stops or disconnects abnormally, the Siebel Synch. Manager may still remain running.
  - If a user connects through a dial-up line via TCP/IP and the line is disconnected, then the TCP/IP session stays active for a certain time. The Synchronization Manager task cannot close until the TCP/IP session is finally released.

  Workaround:

  - Configure the TCP/IP timeout on the Applications Server. Contact your System Administrator for information about the TCP/IP keep-alive functionality.

- SyncMgr_xxx contains the following error message: "DCK-00214: Directory (null) does not exist"

  Possible Causes:

  - Docking directories of Mobile Web Clients have been deleted.

  Solution:

  Reextracting the mobile users will re-create these docking directories and client should be able to download the latest snapshot files and then synchronize with the server again.

## FAQ: What to Do with an Initialization or Synchronization Problem

1 Look in the <SiebelClientInstall>\log directory for files:

- upgwiz*.log and syncthrd*.log (in case of an initialization problem)
- syncthrd*.log (in case of a synchronization problem)

2 Find the error message in the LOG file.

3 With error messages, search this troubleshooting chapter and SupportWeb.

4 If this does not help resolve the problem, follow "How to Set Client-Side Logging" on page 144 to obtain additional information in the LOG file.

   Does the additional information help to resolve the problem (troubleshooting chapter and SuppportWeb)?

5 If not, log a Service Request and send the *.log information with the additional trace information (Step 4) to support@siebel.com.

# FAQ: Long Initialization/Synchronization Times

When you receive the CSSSISDockFgetACKMsg msg error during initialization or synchronization there are basically two cases to differentiate:

1 You are working in a network environment.

   In this case the error might have occurred due to very heavy network traffic.

2 You are working over a phone line.

   In this case one possible reason could be an unstable telephone line.

   - If possible, you should try different telephone lines.
   - Another reason could also be a bad modem, either on the client side or the server side.
   - In either case, work with your IT department to troubleshoot the problem.

# FAQ: Cannot Connect to Server During Synchronization

As the text of the error message indicates, there is a problem with the connection between the Siebel Remote client and the Siebel server.

If this error occurs, do the following:

- Ping the Siebel server from the client by running the following command from the DOS window on the client machine:

  ```
  ping <server machine name>
  ```

  If the server cannot be reached by name, try the IP address. The expected result is that the ping is able to resolve the host name to an IP address and connect to the machine.

- Before synchronizing again, refer to "How to Set Client-Side Logging" on page 144 to increase the level of client-side tracing. It is located in the \"log" directory of the client. It is named SyncThrd*.log. Send this file to Siebel Technical Support for review.

  If the connection is established on the server then there will also be a Synchronization Manager trace file. It is named SyncMgr*.log located in the Siebel server log directory.

  Check the DockConnString in the [Local] section of the client's CFG file and confirm that it is set to the host name of the Siebel server with which this client will synchronize. Send the client's CFG file to Siebel Technical Support for review as well.

  It is recommended that you specify the DockConnString in the following format:

  - Siebel server machine name: network protocol: sync port #: server: encryption.
  - Siebel server machine name, assuming the default static port number is 40400.

  Remote users use the DockConnString parameter in the CFG file to synchronize to the server. For details regarding the syntax of the DockConnString parameter, see "Synchronization Parameters in Configuration Files" on page 75.

- If the Synchronization Manager component is not running, check the following:

  - %Siebel_server%\log\SynchMgr* log file.
  - If the failure message is Login Failed, then verify that Synchronization Manager uses the correct user name and password when connecting to the database.

    This should be the system administrator user (SADMIN) and not the database tableowner (SIEBEL or dbo). The system administrator's password in the database must match the password registered in the Gateway Server. If the SADMIN password changed in the database but not in the Gateway Server, users will not be able to log in to Server Manager views. To register a new password with the gateway, verify that the Gateway Server service is running and complete one of the options below from the %Siebel_Server%\bin command prompt.

    **Using Server Manager**, enter:

    ```
    srvrmgr /g <GatewayName> /e <EnterpriseName> /u <username> /p <password>
    ```

    /g: specifies the host name of the machine running the gateway

    /e: enterprise server name

    /u: Siebel Administration username

    /p: Siebel Administration password

    When the Server Manager prompt appears, type:

    ```
    srvrmgr> change ent param Password=NewSADMINPassword
    ```

    Type `exit`.

    For more information about Server Manager, see *Siebel System Administration Guide*.

    **Using srvrcfg**, enter:

```
srvrcfg /g <GatewayName> /e <EnterpriseName> /m enterprise /w
Password=NewSADMINPassword
```

Stop and restart the Siebel server service to invoke the change.

**NOTE:** Use caution when changing the password—an incorrect entry will cause errors throughout your system.

If the password of the database account is unchanged, the password of the system administrator account can be changed in the Server Manager views.

### *To view the current Synchronization Manager port number*

1. From the application-level menu, choose Navigate > Site Map > Administration - Server Configuration > Servers.
2. In the Siebel Servers list, select a server that runs Synchronization Manager, and then click the Components view tab.
3. In the Components list, use standard query techniques to select the record for the Synchronization Manager component, and then click the Parameters subview tab.
4. In the Component Parameters list, use standard query techniques to locate the record for the Static Port Number parameter.

   The port number in use is in the Value field.

### *To change the current Synchronization Manager port number*

1. Follow all steps in the previous procedure, "To view the current Synchronization Manager port number."
2. Enter the new value for the Static Port Number in the Value on Restart field.
3. Restart the server.
4. Test the connection by copying a CFG file, from a user that is able to connect or synchronize, onto the client machine where the behavior is occurring.

   Observe the results and compare the two CFG files for any noticeable differences.

# What to Do When Transaction Merger Fails

This section describes a temporary workaround to restart Transaction Merger until the root cause has been found.

### *To work around a Transaction Merger's failure*

1 Check out the log file (TxnMerge*.log) to find the Mobile Web Client and the *.dx-file this is happening for. In the example following, the Mobile Web Client is SADMIN and the *.dx is 00000009.dx.

**CAUTION:** Renaming or deleting the DX files in siebsrvr\Docking\Client\Inbox directory is not allowed. If you rename or delete DX files you will lose the transactions and will have to reextract the Mobile Web Client.

```
[TRC35] >>> Processing Client: SADMIN

[TRC35]   File: c:\Sea704\Siebsrvr\docking\SADMIN\inbox\00000009.dx

[TRC33] 2000-10-06 12:09:51 Client: SADMIN, File:
c:\Sea704\Siebsrvr\docking\SADMIN\inbox\00000009.dx.

[DBG33] 2000-10-06 12:09:51 Message: Generated SQL statement:,

Additional Message: SQLExecute: INSERT INTO dbo.S_OPTY_PROD_X (ATTRIB_01, ATTRIB_02,
ATTRIB_03, ATTRIB_04, ATTRIB_05, ATTRIB_06, ATTRIB_07, ATTRIB_08, ATTRIB_09,
ATTRIB_10, ATTRIB_11, ATTRIB_12, CONFLICT_ID, CREATED, CREATED_BY, LAST_UPD,
LAST_UPD_BY, MODIFICATION_NUM, PAR_ROW_ID, ROW_ID)

VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
```

**NOTE:** In the log file, following the line with the series of question marks, you can find the actual values, which are bound to each by the question mark.

For example:

```
INSERT INTO dbo.S_EMPLOYEE(NAME, AGE, SEX)
VALUES(?,?,?)
NAME:    Bill
AGE:     40
SEX:      M
```

```
[DBG33] 2000-10-06 12:09:51 Message: Error: An ODBC error occurred,

Additional Message: Function: DICInsRowExecStmt; ODBC operation: SQLExecute
```

2 If the error is specific to one Mobile Web Client, rename the INBOX directory for this Mobile Web Client (that is, from C:\Sea704\SiebSrvr\Docking\Sadmin\Inbox to C:\Sea704\SiebSrvr\Docking\Sadmin\Inbox_Old) and restart Transaction Merger.

3 If Transaction Merger runs after that, this is only affecting one Mobile Web Client. If Transaction Merger fails again, you can expect that the failure applies to the Mobile Web Clients.

4 Open a Service Request on the SupportWeb and send the related *.dx-file and the trace file for further analysis.

**NOTE:** This is a temporary workaround because the specific Mobile Web Client will not be able to synchronize.

# Users Who Cannot See Records When Connected Locally

There are many reasons why mobile users may not see a record through the user interface when connected to their local database. To troubleshoot such an issue, the instructions in this FAQ may help you determine the reasons for the behavior.

The following paragraphs describe several situations and tips to solve the problems.

From the application-level menu, choose Navigate > Site Map > Administration - Siebel Remote > Mobile Clients view. Make sure the Mobile Client has the appropriate routing model. In addition, make sure the Receiving Transactions check box is checked. If the routing model is set correctly but Receiving Transactions is not checked, search the database extract log file for errors. It is an indication that database extraction is not successful.

From the application-level menu, choose Navigate > Site Map > Administration - Application > System Preferences view; verify that the 'Docking: Transaction Logging' parameter value is set to True. If this parameter is set to False, you should set it to True, after which you must reextract the Mobile Web Clients.

Make sure the transaction processor and router are running and also check the transaction processor(txnproc_xxx.log) and router(txnroute_xxx.log) log files for errors.

For limited visibility records, make sure that the record in question is visible in one of the "My… "or "My Team's" views when the same user is connected directly to the server. For Organization, Opportunity, Contact and Service Request Dock Object, these records are also routed to the mobile users if the records are available through drilldown from a My or My Team's view. A list of limited and Enterprise visible objects can be found by using Siebel Tools to query the repository.

Use the visutl.exe utility to check if the mobile user has visibility to the records. Review the log file generated by this utility. If visutl.exe reports that the record is not visible, this means that the record does not reside on the mobile user's local database. You can find a logical explanation for what makes certain records visible to the mobile users by logging into Siebel Tools. Using the Flat tab, navigate to the Dock Object Table in the Object Explorer. Once you have done this, go back to the Types tab, and choose Dock Object >Dock Object Visibility Rule object type. In the Dock Object Visibility Rule object List Editor window, right scroll to the Comments field. You will find the explanation for each rule that is associated to that dock object. Refer to the FAQ question, "What is visutl and how do I run it?"

If the visutl.exe reports that the record is visible, log on using isql55 to verify if the record resides on the local database. Refer to FAQ, "How to log on a local database using isql55."

Once you have logged on successfully, run the following query:

```
Select * from SIEBEL.<TABLENAME> where ROW_ID = '<Rowid of the non-visible record>'
```

If the record resides on the local database but is not visible through the user interface, and if you are using a custom SRF file, try to log in to the Siebel application using the standard SRF file. If the records are visible, something in your configuration is filtering out the records. Investigate your configuration.

In addition, you may also start up your Siebel Application with the /S option to spool out the SQL that is being run on the view that does not show the record. Look at the query/joins that are being run to see what may be filtering out the records that should be visible on the User Interface. Refer to the Technical Note, "Using the /S Option to Examine the SQL generated by Siebel Enterprise Applications."

If the record does not reside on the local database, this indicates a potential problem with the Siebel Remote processes. Confirm that the user has synchronized successfully with the remote server. Check the Synthrd_xxx.log files for any errors. If transaction, router and synchronizing are fine, reextract the user and see if the record is visible after the reextraction.

If you are using a collating sequence other than 1252, read Alert, "The SQL Anywhere collating sequence treats characters and their accented or special counterparts as duplicates."

If you still need assistance, send Technical Support the visutl.log located in the current directory where you run the visutl.exe for analysis. In addition, you may perform the following test:

Create a couple of transactions in which you want the Mobile Web Client to have visibility, then start the transaction processor. Next, you start the transaction router with the following event levels set: GenericLog=4, Trace=4, SqlparseandExecute=4. Send the transaction processor and router trace file and the DX files in the <SiebelRoot>\docking\<client>\outbox folder to support@siebel.com for analysis.

# How to Handle a Large Transaction Backlog

This section discusses how to handle a large transaction backlog in the transaction log table.

With the System Preference Docking:Transaction Logging set to TRUE, your Siebel application will record transactions to the transaction log table (S_DOCK_TXN_LOG). The Transaction Processor (txnproc) is responsible for deleting entries from this table—after all txnprocs in the system have copied them to the Application server TXNPROC directory. Enterprise visible data will be routed to the active Mobile Web Clients.

The backlog is the number of transactions in S_DOCK_TXN_LOG, or large number of DX files in TXNPROC directory. For example, to see the backlog in the S_DOCK_TXN_LOG select count (TXN_ID) from S_DOCK_TXN_LOG. Alternatively you may wish to know the oldest transaction, for example select min(CREATED) from S_DOCK_TXN_LOG. However, a backlog of 1000 transactions is not usually considered a problem.

If you are experiencing a large number of rows in S_DOCK_TXN_LOG, or large number of DX files under TXNPROC directory, follow these steps:

1 Check that Transaction Processor and Transaction Router or Routers are running.

   In the Server Tasks view (Administration - Server Management > Tasks), use standard query techniques to find tasks for the Transaction Processor and Transaction Router components, and then check the Task State and Status for each such record. The Task State should be Running for both the Transaction Processor and the Transaction Router.

   At least one Transaction Processor, Router, and Merger are required per Remote server. Multiple Routers and Mergers can be run on one Remote server. Multiple Routers are often recommended.

2 Check that Transaction Processor is processing Transactions.

   Check by looking at the Administration - Siebel Remote > Processor Status view. Here you will find information about the last transaction and last file, created by Transaction Processor in the TXNPROC directory on your Siebel Server. Under normal circumstances and if there are not any problems, these keep increasing.

3 Check for old TP entries.

   There may be old transaction processor entries in the S_NODE table that are no longer in use; either they have been left active after an upgrade or perhaps they are associated to a Siebel server that is no longer in use. It is a best practice to make sure transaction processor entries that are not required are end dated.

4 Check for Transaction Processors without end dates.

   It is very important to make sure old transaction processors have end dates. For example, this may happen when a transaction processor is started and then not used on a server, postupgrade, or a transaction processor is started and then the server is uninstalled.

   To end date an old transaction processor, see "Managing Backlog in the Transaction Log Table."

5 Changes in positional Hierarchy, Territory realignments, or large EIM loads can also create a large number of Transactions.

   The higher in the hierarchy you change, add, or delete one or more positions, the more transactions will be created.

   When using EIM, the parameter LOG TRANSACTIONS TO FILE defaults to TRUE. This means EIM will log transactions under the File_System\eim folder with only one marker logged into the S_DOCK_TXN_LOG table. If the parameter is set to FALSE, EIM will use set-based transaction logging to reduce database contention for S_DOCK_TXN log by only recording one txn per EIM set in S_DOCK_TXN_LOG.

   It can take some time for the Transaction Processor and the Router to work through the created backlog and route the changes to your mobile users. Multiple routers or reextracting the mobile users will speed up processing.

6 If you are still having a problem, contact Technical Support and provide Transaction Processor and Router log files with the following trace flags set:

   - Change the following log events (in srvrmgr):
     - evtloglvl sql=4
     - sqlParseandExecute=4
     - genericlog=4
   - Run the Siebel Remote component with SQL Flag=2, Trace Flag=1

In addition to the steps above, you should examine the indexes on S_DOCK_TXN_LOG and S_DOCK_TXN_SET.

The P1 index is on an ID column that counts upwards. Lower IDs are deleted. This can lead to many index leaf rows pointing to rows that are no longer there.

For this reason, you should rebuild the indexes in the S_DOCK_TXN_LOG and S_DOCK_TXN_SET tables regularly.

# A Client-Side Merge Issues on the Server

This appendix describes how to repair orphaned records on the server created by the merge process occurring while child records are being created on Mobile Web Clients. The solution involves tracking the merge process and then, using a workflow process, reassociating the orphaned child records with their appropriate parents.

## Definition of the Problem

The combination of actions below will create orphaned child records on the server.

- Merging on the server that deletes a parent record.
- Before invoking a synchronization session that would merge the same records on the local database, the user creates or associates child records to the parent that was deleted on the server.
- When synchronization does occur, the child record associated with the deleted parent is orphaned on the server.

The following example illustrates the sequence of events that may occur to create the client-side merge problem. Keep in mind that a client-side merge refers to actions initiated from the user interface rather than another type of merge such as an EIM merge.

To begin, both the Mobile Web Client and the server have equivalent records such as:

- Account A with Activity 1 as a child record
- Account B with Activity 2 as a child record

The client-side merge problem arises *if* the following sequence of activities takes place:

1. Account A and Account B are merged on the server into Account B. As a result, the following happens on the server:
   - Account A is deleted.
   - Account B has both Activity 1 and Activity 2 as children. This happens because the object manager issues a multirow update statement for each possible child table of the Account A table.
   - The above actions generate transactions that are placed in the outbox for the Mobile Web Client.
2. Before the transactions in Step 1 are routed to the Mobile Web Client, the following occurs:
   - The Mobile Web Client inserts Activity 3 as a child record of Account A and synchronizes.
3. The Mobile Web Client synchronizes and the following occurs:
   - Activity 3 is routed up to the server. An Activity 3 is created with parent Account A on the server. This activity is an orphaned record because its parent does not exist on the server.

- The merge operations on the server described in Step 1 are sent down to the Mobile Web Client. Activity 3's parent is reset to Account B on this Mobile Web Client. Consequently, the client-side merge does not create an orphaned record on the Mobile Web Client.

# Solution

This section describes the solution and how to repair the client-side merge problem.

The solution is to provide the ability to repair orphaned data on the server. To do this you will use the following:

- Log all merge transactions on the server
- Use a batch process to reassociate orphaned child records with the appropriate parents using the log
- A view that shows the content of the log for administration and system management

For organizations with the client-side merge problem, use the following procedures to repair the orphaned child records.

### *To turn on CSM Logging*

1. From the application-level menu, choose Navigate > Site Map > Administration - Application screen.
2. In the link bar, select System Preferences.
3. Select System Preference Name=CSM Logging.
4. Set the System Preference Value=TRUE.
5. Restart the Siebel Server.

   This allows system preferences to take effect.
6. Restart the Siebel Remote components.

   See "Starting and Stopping Siebel Remote Server Components" on page 126.

The next step should be to create a workflow process that calls the appropriate business service. Create a workflow process that calls the CleanUp business service. To control the size of this log, create another workflow process to remove old merge transactions.

The following two procedures describe how to create these workflows. Two workflows should be created for each method of CSM Log Service.

### *To repair orphaned child records*

This is also called the CleanUp () function.

1. Create a workflow with three steps: Start, CleanUp, and End.

   See *Siebel Business Process Designer Administration Guide* for details on how to create workflows.

2 For the CleanUp step, set the following:

   a Set Business Service to CSM Log Service.

   b Set Method to CleanUp.

   c There is no input argument or output argument for this method.

### *To delete entries in the CSM log*

This is also called the RemoveOldEntry () function.

1 Create a workflow with three steps: Start, RemoveOldEntry, and End.

   See *Siebel Business Process Designer Administration Guide* for details on how to create workflows.

2 For the RemoveOldEntry step, set the following:

   a Set Business Service to CSM Log Service.

   b Set Method to RemoveOldEntry.

3 Create one input argument and set Input Argument=TransactionId.

4 Find the value of this input argument in the CSM Log View.

   This workflow deletes records in the CSM Log table that have a transaction ID smaller than the input argument.

### *To run a workflow from the command line*

1 From a DOS (or shell) prompt, change the current directory to:

   ```
   <Siebel Installation directory>\siebsrvr\bin
   ```

2 Enter:

   ```
   srvrmgr /g <gateway> /e <enterprise> /s <siebsrvr> /u <username> /p <password>
   ```

3 To run wfprocmgr, enter:

   ```
   start task comp wfprocmgr with ProcessName="<Workflow Name>"
   ```

Administrators determine how frequently to repair orphaned child records and delete CSM log entries. Typically, the frequency should be determined by how often Merging is used to clean up duplicate entries.

# B Docking Object Changes

This appendix contains a table that provides an overview of the differences between the docking objects in Siebel 7.7 and Siebel 7.5.

## Routing Definition Differences in Docking Objects

Table 32 compares the routing definition differences between Siebel 7.5 and Siebel 7.7 docking objects, and lists new ones for Siebel 7.7. The table provides a comparison of pairs of docking objects between the two versions, or identifies new docking objects for Siebel 7.7. To read Table 32 use the following:

- If the Version field of a row reads 7.5, compare it with the 7.7 row immediately below it.
- If the Version field of a row reads 7.7 and the Version field of the row immediately above it also reads 7.7, a docking object is new in Siebel 7.7.

Table 32. Docking Comparison - Docking Object Differences

| Version | Docking Object | Visibility Level | Primary Table | DO Active? |
|---|---|---|---|---|
| 7.7 | ActivityWarehouseHierarchy | Private | S_EVT_ACT_WHIR | Y |
| 7.7 | ApprovalLevel | Enterprise | S_FN_APPR | Y |
| 7.5 | AssignGroup | Limited | S_ASGN_GRP | Y |
| 7.7 | AssignGroup | Enterprise | S_ASGN_GRP | Y |
| 7.5 | Audit Log | Private | S_AUDIT_ITEM | Y |
| 7.7 | Audit Log | Limited | S_AUDIT_ITEM | Y |
| 7.5 | Audit Trail | Private | S_AUDIT_TRAIL | Y |
| 7.7 | Audit Trail | Limited | S_AUDIT_TRAIL | Y |
| 7.7 | Barcode | Limited | S_BARCODE | Y |
| 7.7 | Barcode Enable | Limited | S_BCODE_ENABLE | Y |
| 7.5 | BusComp Audit | Limited | S_AUDIT_BUSCOMP | Y |
| 7.7 | BusComp Audit | Enterprise | S_AUDIT_BUSCOMP | Y |
| 7.7 | CFGUIGrpItem | Enterprise | S_CFG_UIGRP_IT | Y |
| 7.7 | CM MessageStatusTracking | Private | S_CM_MSG_STAT | Y |
| 7.5 | CalendarSharing | Limited | S_EMP_APPT | Y |
| 7.7 | CalendarSharing | Limited | S_EMP_APPT | N |

Table 32. Docking Comparison - Docking Object Differences

| Version | Docking Object | Visibility Level | Primary Table | DO Active? |
|---|---|---|---|---|
| 7.7 | CampListOutput | Private | S_CAMP_LSTOUTPT | Y |
| 7.7 | CommInboundEvent | Private | S_CM_INBND_EVT | Y |
| 7.7 | CompPlan Employee | Private | S_CP_EMP_PER | Y |
| 7.7 | CompatibilityMatrix | Limited | S_PRODCOMP_MTRX | Y |
| 7.5 | ContactProduct | Limited | S_CON_PRDINT | Y |
| 7.7 | ContactProduct | Limited | S_CON_PRDINT | N |
| 7.7 | ContactSyncSchd | Private | S_CON_SYNC_SCHD | Y |
| 7.7 | ContainerTxn | Private | S_IC_ACHV_TXN | Y |
| 7.7 | DMAP | Enterprise | S_BUSOBJ_DMAP | Y |
| 7.7 | DataValidation | Private | S_VALDN_RL_SET | Y |
| 7.7 | DispatchBoardLog | Private | S_DSP_BRD_LOG | Y |
| 7.7 | DockRequestRecord | Private | S_DOCK_REQ_REC | Y |
| 7.7 | DockTimeFilter | Enterprise | S_DOCK_TM_FLTR | Y |
| 7.7 | DynamicMatrix | Limited | S_DYN_PMTRX | Y |
| 7.5 | EAIBatch | Private | S_EAI_BATCH | Y |
| 7.7 | EAIBatch | Private | S_EAI_BATCH | N |
| 7.7 | EMTDataSet | Private | S_EMT_DATA_SET | Y |
| 7.7 | EMTObject | Private | S_EMT_OBJECT | Y |
| 7.7 | EMTSession | Private | S_EMT_SESSION | Y |
| 7.7 | EMTSessnItm | Private | S_EMT_SESSN_ITM | Y |
| 7.7 | EligibilityMatrix | Limited | S_PRODELIG_MTRX | Y |
| 7.7 | EmailResponseTime | Private | S_EM_RESP_TM | Y |
| 7.7 | Employee Position Period | Private | S_EMP_POSTN_PD | Y |
| 7.7 | ExcelIntegration | Limited | S_XL_VIEW_MAP | Y |
| 7.7 | Handheld Admin Application | Limited | S_MOBILE_APPL | Y |
| 7.7 | Handheld Admin Buscomp Filter | Limited | S_HH_BCFILTER | Y |
| 7.7 | Handheld Admin Busobj Filter | Limited | S_HH_BOFILTER | Y |
| 7.7 | Handheld Admin Setting | Limited | S_HH_SETTING | Y |
| 7.7 | Handheld Admin Summary | Limited | S_HH_SYNC_SUM | Y |

Table 32. Docking Comparison - Docking Object Differences

| Version | Docking Object | Visibility Level | Primary Table | DO Active? |
|---|---|---|---|---|
| 7.7 | Handheld Conflict Admin | Limited | S_HH_TXN_ERR | Y |
| 7.7 | ICSrvrTask | Private | S_IC_SRVR_REQ | Y |
| 7.7 | ICUserProperty | Private | S_IC_USER_PROP | Y |
| 7.7 | ISSCfgRuleNode | Private | S_ISS_CFRL_NODE | Y |
| 7.7 | ISSCfgServer | Private | S_ISS_CFG_SRVR | Y |
| 7.7 | ISSCfgValdtn | Private | S_ISS_CFG_VALDN | Y |
| 7.7 | ISSCfgValdtnScenario | Private | S_ISS_CFG_VALSC | Y |
| 7.7 | InvoicableCharge | Limited | S_INVOICE_CHRG | Y |
| 7.7 | InvoiceConsolidationPlan | Private | S_INVCNSLD_PLN | Y |
| 7.7 | InvoiceSignature | Private | S_INVOICE_SIGN | Y |
| 7.7 | JWS | Private | S_ISS_JWS | Y |
| 7.7 | LessonBasedURL | Private | S_CRSE_LSN_URL | Y |
| 7.7 | MSGControlTable | Private | S_MSG_CONTROL | Y |
| 7.5 | MSProjectIntegration | Enterprise | S_MPP_MAP | Y |
| 7.7 | MSProjectIntegration | Limited | S_MPP_MAP | Y |
| 7.7 | MaketingBudgetReq | Limited | S_MKTG_BDGT_REQ | Y |
| 7.7 | MarketingGoalTmpl | Private | S_SRC_GOAL_TMPL | Y |
| 7.7 | MarketingServerTask | Private | S_MKTG_SRVR_TSK | Y |
| 7.5 | MergeLog | Limited | S_MERGE_LOG | Y |
| 7.7 | MergeLog | Private | S_MERGE_LOG | Y |
| 7.7 | OMTempTable | Private | S_OM_TEMP | Y |
| 7.5 | OnLink Category | Private | S_ONL_CATEGORY | Y |
| 7.7 | OnLink Category | Limited | S_ONL_CATEGORY | Y |
| 7.5 | OnLink Object | Private | S_ONL_OBJECT | Y |
| 7.7 | OnLink Object | Limited | S_ONL_OBJECT | Y |
| 7.7 | OutlookIntegration | Limited | S_MC_OBJ_TYPE | Y |
| 7.7 | PSPFactorType | Enterprise | S_PSP_FCTR_TYPE | Y |
| 7.7 | PSPValueMap | Enterprise | S_PSP_VAL_MAP | Y |
| 7.7 | PerformanceFormula | Limited | S_IC_PFMS_FRMLA | Y |
| 7.7 | PersonResponsibility | Limited | S_PER_RESP | Y |

Table 32. Docking Comparison - Docking Object Differences

| Version | Docking Object | Visibility Level | Primary Table | DO Active? |
|---|---|---|---|---|
| 7.7 | PostnCon | Limited | S_POSTN_CON | Y |
| 7.5 | ProductDefectProductLine | Limited | S_PRDFCT_PRODLN | Y |
| 7.7 | ProductDefectProductLine | Enterprise | S_PRDFCT_PRODLN | Y |
| 7.7 | Quick Fill | Limited | S_QF_TMPL | Y |
| 7.7 | RecurringSchedule | Enterprise | S_RCRNG_SCHED | Y |
| 7.7 | Region | Limited | S_REGION | Y |
| 7.7 | RetroProcess | Private | S_IC_RETRO_PROC | Y |
| 7.7 | SMQ Administration | Limited | S_SMQ_MSG_OBJ | Y |
| 7.7 | SMQ History | Limited | S_SMQ_HISTORY | Y |
| 7.7 | Server Domain | Limited | S_SERVER_DOMAIN | Y |
| 7.7 | Server Domain Engine | Private | S_SD_ENG_IDEN | Y |
| 7.7 | Server Domain List of Value | Limited | S_SD_PIM_LIC | Y |
| 7.7 | Server Domain PIM RowId Map | Private | S_SD_PIM_ROW | Y |
| 7.7 | Server Domain Profile | Limited | S_SD_PIMSV_CFG | Y |
| 7.7 | Server Domain Recurrence | Private | S_SD_REPEAT | Y |
| 7.7 | Server Domain SEBL RowId Map | Private | S_SD_SEBL_ROW | Y |
| 7.7 | Server Domain Server Profile | Private | S_SD_SV_PROFILE | Y |
| 7.7 | Server Domain Translation Map | Limited | S_SD_TRANSL_MAP | Y |
| 7.7 | Server Domain User Map | Private | S_SD_USER_MAP | Y |
| 7.7 | ServerTaskMsg | Private | S_SRVR_TASK_MSG | Y |
| 7.5 | Service Region | Enterprise | S_SRV_REGN | Y |
| 7.7 | Service Region | Limited | S_SRV_REGN | Y |
| 7.7 | ServiceRegionBusinessUnit | Limited | S_SRV_REGN_BU | Y |
| 7.7 | ShippingCalcMatrix | Enterprise | S_SHIPCALC_MTRX | Y |
| 7.7 | Sort Character | Private | S_SORT_CHAR | Y |
| 7.7 | StandardEntlMatrix | Limited | S_STDENTL_PMTRX | Y |
| 7.7 | StandardProdMatrix | Limited | S_STDPROD_PMTRX | Y |

Table 32. Docking Comparison - Docking Object Differences

| Version | Docking Object | Visibility Level | Primary Table | DO Active? |
|---|---|---|---|---|
| 7.7 | StandardTrngMatrix | Limited | S_STDTRNG_PMTRX | Y |
| 7.5 | Task Based Process | Limited | S_TASK_PROCESS | Y |
| 7.7 | Task Based Process | Limited | S_TASK_DEFN | Y |
| 7.7 | VOD | Limited | S_VOD | Y |
| 7.7 | VODCacheSync | Private | S_VOD_CACHESYNC | Y |
| 7.7 | ValidationMessage | Private | S_ISS_VALDN_MSG | Y |
| 7.7 | Workflow Batch | Private | S_WFA_BATCH_JOB | Y |
| 7.7 | Workflow Definition | Limited | S_WFA_DPLOY_DEF | Y |
| 7.7 | Workflow Definition Log | Private | S_WFA_DEFN_LOG | Y |
| 7.7 | Workflow Instance Log | Private | S_WFA_INST_LOG | Y |
| 7.7 | Workflow Instance Step Log | Private | S_WFA_INSTP_LOG | Y |
| 7.7 | Workflow Runtime | Private | S_WFA_INSTANCE | Y |
| 7.7 | Workflow StepProp Log | Private | S_WFA_STPRP_LOG | Y |
| 7.7 | ePortalTemplate | Private | S_PAGE_ITEM_SET | Y |

# C Routing Models for Financial Services

This appendix describes the routing models in Siebel Financial Services that are different from the routing models in Siebel Remote and Replication Manager.

## Using Routing Models in Siebel Financial Services

This section describes the routing models available out-of-the box in Siebel Financial Services. Each model determines what data will be extracted to, and what follow-on transactions will be routed to, Mobile Web Clients belonging to that model.

Each mobile user is associated with one routing model. A Routing Model includes a set of routing rules. The union of the routing rules determines whether a record will be routed to a Mobile Web Client. By careful application of specific routing models, local database sizes can be reduced, and, as a result, so are synchronization times and transaction application times.

**CAUTION:** Make sure Routing Models are consistent with the responsibilities and positions of the mobile users. The responsibilities and position of an employee determine the access that person has to the Server database. Balancing the data routing model with a user's access helps to optimize the size of that user's local database.

Siebel Financial Services includes the following, pre-defined, financial-services-specific routing models as indicated Table 33 on page 227.

Table 33. Pre-defined Siebel Financial Services Routing Models

| Routing Model | Siebel Responsibility | Type |
|---|---|---|
| Institutional Finance Sales Manager | Institutional Finance Sales Manager | Specialized |
| Institutional Finance Analyst | Institutional Finance Analyst | Specialized |
| Credit Originator | Credit Originator | Specialized |
| Insurance Sales Agent | Insurance Sales Agent | Specialized |
| Insurance Sales Manager | Insurance Sales Manager | Specialized |
| Claims Adjuster | Claims Adjuster | Specialized |
| Claims Manager | Claims Manager | Specialized |

Each of these models should be used with a specific Siebel Responsibility as shown above. For customers that have a business requirement to reduce the amount of data replicated to mobile users, use one of the specialized routing models may be appropriate. Before deploying Siebel Remote with any of these specialized routing models, it is strongly recommended that you discuss this with Siebel Technical Support.

## Analyzing the Use of Routing Models

Before using the specialized Siebel Routing Models, assess and consider the impact of the following:

- Analyze the mobile users' usage patterns. Determine what data the users need and what they do not need when using Siebel Financial Services in the local mode.
- Compare the usage pattern with the definition of each routing model. Each model was designed for a specific class of mobile users.
- Perform a review of your Siebel installation and consider the following:
  - Did you modify any of the views assigned to the Siebel Responsibilities?

    If an MVG field were added to an existing view, a mobile user could set the value of the MVG field to a "No Match" row-id by modifying any fields in the row or selecting the MVG field. The reason is that no data for the MVG was downloaded to Mobile Web Client. This action implicitly creates an update transaction to be replicated to the server during the next synchronization session. Database integrity is compromised when the transaction is applied on the server.
  - Did you modify any of the Siebel Responsibilities by adding new views that are accessible in local mode?

    Generally, views that contain data from limited visibility objects should have routing rules to support them. If a view is added that does not have corresponding routing rules, then no data will be replicated for that specific view. If such a view was added to a Siebel Responsibility and the implementation uses Client Wins for update conflict resolution, the following undesirable situation could occur—this should be avoided.

    - A mobile user creates a new record using the newly added view on the Mobile Web Client.
    - The mobile user synchronizes with the server.
    - A Web client user makes changes to the same record on the server.
    - The mobile user synchronizes and does not see the updates by the Web client to the same record.
    - The mobile user makes changes to the record in the Mobile Web Client.
    - The mobile user synchronizes and overwrites the content of the record on the server.

    In this manner, a mobile user could unintentionally overwrite updates made by another user to the same record.

- Exposing a custom foreign key in a Siebel view and business component.

  A custom foreign key may not have any Siebel Remote routing support, because routing rules were either deactivated out-of-the-box or no rule was created. As a result, no data will be routed to support the foreign key. If any base record containing this foreign key is touched or updated by a mobile user in their local database, it may inadvertently reset the reference to null and in turn, replicate this change to the server.

- If the answer to any of the above questions was yes, do not use any of the specialized routing models in Table 33 on page 227.
- If the answer to all the above questions was no, conduct a field test with a small group of representative mobile users. Use the appropriate routing models and assess the impact of the specialized routing models on the Transaction Router. During the test, observe the following:
  - Local database size and the amount of data being replicated—should be reduced.
  - Mobile users can still perform business tasks—no data is missing that prevent the users from completing a critical business process.
  - Synchronization times—should be shorter.
  - Every server transaction should be routed to the Mobile Web Client except for those explicitly excluded in the routing models—if not, the users responsibilities may not match the routing model.
  - Every view that has data on the server should show the same data on the mobile client—if not, there is a strong possibility the wrong views were included in the users' responsibilities.
- If you have acceptable results from the field test, you are ready to deploy the routing models to the appropriate mobile users.

# Descriptions of Financial Services Routing Models

The descriptions on the following pages include the routing models available out-of-the-box in Siebel Financial Services. For more information about other Siebel eBusiness Applications routing models and application of these, see "Using Routing Models" on page 89.

## Insurance Sales Agent

The Insurance Sales Agent model should be used for insurance field sales agents in your organization. This routing model should also be used for other remote users such as health insurance sales agents and site-visit coordinators. Users assigned the Insurance Sales Agent routing model receive a database extract and follow-on transactions dealing with information relevant to their positions.

The business components associated with this model include the following:

- Activities
- Applications
- Calendar Access

- Companies
- Contacts
- Events
- Facilities
- Households
- Opportunities
- Policies
- Providers
- Claims
- Service Requests

## Insurance Sales Manager

This routing model should be used for insurance sales managers in your organization. Users assigned the Insurance Sales Manager routing model will receive database extracts and follow-on transactions dealing with information relevant to a manager of insurance sales agents. All records routed to any of a manager's direct reports will also be routed to the manager.

The business components associated with this model include those listed above for the Insurance Sales Agent in addition to the manager's Team's data associated with those business components.

## Claims Adjuster

The Claims Adjuster model should be used for insurance field claims adjusters in your organization. The business components associated with this model include the following:

- Households
- Contacts
- Policies
- Service Providers
- Claims
- Claim Elements
- Service Requests
- Activities
- Calendar Access

## Claims Manager

This routing model should be used for insurance claims managers in your organization. Users assigned the Claims Manager routing model will receive database extracts and follow-on transactions dealing with information relevant to a manager of claims adjusters. All records routed to any of a Claims Manager's direct reports will also be routed to the Claims Manager.

The business components associated with this model include those listed above for the Claims Adjuster in addition to the manager's Team's data associated with those business components.

## Credit Originator

This routing model should be used for individuals in your organization responsible for originating mortgages or other consumer loans and for managers of these individuals. Users assigned the Credit Originator routing model will receive database extracts and follow-on transactions dealing with information relevant to loan origination. Managers will see all records for themselves and for all of their direct reports. A manager who does not have any direct reports will not receive any team-based data.

The business components associated with this model include the following:

- Activities
- Applications
- Contacts
- Events
- Opportunities
- Products

## Institutional Finance Sales Manager

Users assigned the Institutional Finance Sales Manager routing model will receive database extracts and follow-on transactions dealing with information relevant to a manager of institutional finance sales. Managers see all records for themselves and all of their direct reports. A manager who does not have any direct reports will not receive any team-based data. This routing structure allows this model to be used for managers and nonmanagers alike, making for a simpler setup and administration process.

The business components associated with this model include the following:

- Activities
- Applications
- Calendar Access
- Call Reports
- Companies
- Contacts
- Employees

- Events
- Expense Reports
- Financial Accounts
- Holdings and Interests
- Households
- Literature
- Opportunities
- Products
- Securities
- Service Requests

**NOTE:** Literature routing is handled somewhat differently. Links to the literature are synched to the client but the actual literature attachments are only synched upon request.

## Institutional Finance Analyst

Users assigned the Institutional Finance Analyst routing model receive database extracts and follow-on transactions dealing with information relevant to a institutional finance analyst. Managers see all records for themselves and for all of their direct reports. A manager who does not have any direct reports will not receive any team-based data. This routing structure allows this model to be used for managers and non-managers alike, making for a simpler setup and administration process.

The business components associated with this model include the following:

- Activities
- Calendar Access
- Call Reports
- Companies
- Contacts
- Employees
- Events
- Expense Reports
- Holdings and Interests
- Literature
- Products
- Securities
- Service Requests

# Index

## N



## O

## S

## T

## U



## V



## W