SIEBEL 7
eBusiness

SIEBEL REMOTE ADMINISTRATION GUIDE

MIDMARKET EDITION

VERSION 7.5

12-BD5L5E

SEPTEMBER 2002

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404
Copyright © 2002 Siebel Systems, Inc.
All rights reserved.
Printed in the United States of America

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

The full text search capabilities of Siebel eBusiness Applications include technology used under license from Hummingbird Ltd. and are the copyright of Hummingbird Ltd. and/or its licensors.

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

Supportsoft™ is a registered trademark of Supportsoft, Inc. Other product names, designations, logos, and symbols may be trademarks or registered trademarks of their respective owners.

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

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

Contents

Introduction

How This Guide Is Organized 10

Revision History 10

Chapter 1. Siebel Remote Concepts

Siebel Remote Architecture 13

- Elements in Siebel Remote Architecture 13
- Siebel Remote Flow Diagram 16

Siebel Remote Data Store Components 23

- Siebel Database Server 23
- Siebel File System 24
- Local Database and File System 25

Siebel Remote Client Software 26

Siebel Remote Server Components 27

- Creating Siebel Server Directories for Mobile Web Clients 27
- Generate New Database 28
- Database Extract 29
- Synchronization Manager 29
- Transaction Processor 30
- Transaction Router 30
- Transaction Merger 31

Using Siebel Remote 32

- Registering a Mobile User 32
- Generating a Database 32
- Extracting a Mobile Web Client 33

Initializing the Local Database 33
Synchronizing a Mobile Web Client 36
How Changes Are Propagated to and from a Mobile Web Client 38
Synchronizing a Mobile Web Client Machine 44
Security and Authentication 45
Authentication During Synchronization with Server Database 45
Authentication During Normal Operations 46
Locking and Concurrency 47
Conflict Detection and Resolution 48
Update Conflicts 49
Duplicate Conflicts 50
Delete Conflicts 51
Merge Conflicts 51

Chapter 2. Implementing Siebel Remote Server

Setting Siebel Remote System Preferences 54
MRG: System Conflict Resolution 57
MRG: Docking Timestamp Source 57
Docking: Transaction Logging 58
Docking: Get All PostnCon rows 59
CSM Logging 59
MRG: Txns Per Commit 59
MRG: User Friendly Notification 60
LOGMGR: Vis Rules Per Statement 60
DBX: Vis Rules Per Statement 1 60
DBX: Vis Rules Per Statement N 61
Enterprise DB Server Code Page 61
Disabling Local Access to All Views 62
Starting Siebel Remote Server Components 63
Transaction Processor 65
Transaction Router 67

Transaction Merger 69
Synchronization Manager 70
Enabling Windows NT Rights on Siebel Remote Server 72
Changing the Local Database Administrator Password 73
Generating a New Database Template 74
Creating a New Empty Database File 74
Running the Generate New Database Component 74
Distributing Database Templates to Siebel Remote Servers 77

Chapter 3. Setting Up Mobile Web Clients

Setting Up Mobile Web Client Hardware and Software 80
Enabling Network Connectivity 80
Establishing Autodial Preferences 80
Setting Synchronization Preferences 81
Synchronization Parameters 82
Using Autosynchronization 83
Enabling Encryption for Synchronization Networking 86
Using a Different Data Source 87
Registering a Mobile Web Client 88
Routing Rules and Dock Objects 91
Routing Rules 91
Dock Objects 92
Using Routing Models 93
Field Sales Representative Standard 94
Field Sales Manager Standard 95
Field Technician 95
Senior Field Engineer 96
Field Engineer 100
Consultant 102
Analyst 104

Minimal Data 105
Mobile Partner User 106
Mobile Client – Standard 106
Mobile Client – Extract Only 106
Executive Management 106
Limiting Views Available to Mobile Web Clients 107
Using the Responsibilities View 108
Using the Views View 109
Creating Mobile Web Client User Accounts and Privileges 109

Chapter 4. Extracting Databases for Mobile Web Clients

Database Extraction for a Mobile Web Client 112
Database Extraction for Multiple Users 114
Example of Extracting Databases for Multiple Users 115
Performing a Database Extract to a CD Directory 116
Initializing a Mobile Web Client Database 121
Enabling the Stand-Alone Synchronizer 123
Viewing Session Details on Mobile Web Clients 125

Chapter 5. Administering Siebel Remote

Starting and Stopping Siebel Remote Server Components 128
Transaction Processor 129
Transaction Router 130
Transaction Merger 131
Monitoring Siebel Remote Operations 132
Monitoring Mobile Web Client Status 132
Monitoring Transaction Logs 140
Monitoring Siebel Remote Server Operation Status 143
Data Synchronization 144
Managing Synchronization Frequency 144
Auto Synchronization 145

Sending Messages to Mobile Users 146
Refreshing a Client Database 147
Deactivating and Reactivating a Mobile Web Client 148
Deleting a Mobile Web Client 149
Changing Routing Models 150
Adding New Mobile Users 151
How to Set Client-Side Logging 151
Event Levels 151
Log File Naming 153
Archiving 153
Log File Location 154
Event Tracing for Locking 155
Handling Failure and Recovery 157
Siebel Remote Transmission Failure 157
Siebel Remote Server Failure 157
Siebel Remote Server Media Failure 157
File Server Media Failure 158
Database Server Failure 158
Server Database Records Truncated or Changed 159
Database Server Media Failure 159
Siebel Client Database Failure 161
Recovery from Client Initialization Failure 162
Restoring the File System After Recovery from a Previous Image 162

Chapter 6. Upgrading Mobile Web Clients

Repository Upgrade with Siebel Anywhere 164
Repository Upgrade Without Siebel Anywhere 166
Major Upgrades for Mobile Web Client 167
Apply Patches to the Mobile Web Client 168

Chapter 7. Troubleshooting Remote

Setting Up Remote Server and Remote Clients 170
Synchronization Manager 172
SynchMgr Situation 172
FAQ: What to Do with an Initialization or Synchronization Problem 174
FAQ: Long Initialization/Synchronization Times 174
FAQ: Cannot Connect to Server During Synchronization 175
What to Do When Transaction Merger Fails 178
Users Who Cannot See Records When Connected Locally 180
How to Handle a Large Transaction Backlog 182

Appendix A. Client-Side Merge Issues on the Server

Definition of the Problem 186
Solution 188

Appendix B. Docking Object Changes

Routing Definition Differences in Docking Objects 192
Visibility Level Changes for Docking Objects 199
New Regional Docking Objects 202

Index

Introduction

This guide provides information necessary to implement, configure, and administer Siebel Remote, MidMarket Edition.

NOTE: All Siebel MidMarket product names include the phrase MidMarket Edition to distinguish this product from other Siebel eBusiness Applications. However, in the interest of brevity, after the first mention of a MidMarket product in this document, the product name will be given in abbreviated form. For example, after Siebel Call Center, MidMarket Edition, has been mentioned once, it will be referred to simply as Siebel Call Center. Such reference to a product using an abbreviated form should be understood as a specific reference to the associated Siebel MidMarket Edition product, and not any other Siebel Systems offering. When contacting Siebel Systems for technical support, sales, or other issues, note the full name of the product to ensure its proper identification and handling.

Although job titles and duties at your company may differ from those listed in the following table, the audience for this guide consists primarily of employees in these categories:

Database Administrators	Persons who administer the database system, including data loading, system monitoring, backup and recovery, space allocation and sizing, and user account management.
Siebel Application Administrators	Persons responsible for planning, setting up, and maintaining Siebel applications.
Siebel System Administrators	Persons responsible for the whole system, including installing, maintaining, and upgrading Siebel applications.

The guide assumes users have an understanding of their system and network.

How This Guide Is Organized

This guide includes two areas: Siebel Remote and Troubleshooting. Also, there are two appendices: Client-Side Merge Issues on the Server and Docking Object Differences.

The first area discusses concepts, implementation, and administration of Remote. The second area provides troubleshooting information using scenarios and FAQs to help administrators resolve problems.

The first appendix discusses repair of orphaned server records that may appear from time to time when you invoke the merge functionality from a client. The second appendix describes the differences in the Docking Objects between Siebel 7.5 and Siebel 7.0.3.

Revision History

Siebel Remote Administration Guide, MidMarket Edition, Version 7.5

Siebel Remote Concepts 1

This chapter provides an overview of Siebel Remote. This includes architecture, components, software, setup, security, and resolution of data conflicts between the two users.

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.

New or Enhanced Capabilities

Siebel 7.5 includes the following enhanced or new capabilities for Remote:

- Reduction of enterprise data routed to mobile Web clients. This helps to optimize the size of the local databases.
- Filtering workflow processes from the local database that apply only at the regional level. This helps optimize the size for the local databases.
- Enhanced tracing capability that helps with system diagnostics.

- Standardized Status reports added to increase availability of status and diagnostic information.
- Autosynchronization to improve the frequency of synchronization. This reduces the outbox backlog and therefore shortens the average synchronization time for mobile Web clients.
- Enhanced routing model support with new local access management for views.
- Language filtering for multilingual deployment.

NOTE: This guide assumes that you have successfully installed your Siebel application and completed the implementation steps described in *Applications Administration Guide, MidMarket Edition*. Replication Manager is not supported in the MidMarket version of the Siebel application.

Siebel Remote Architecture

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

- "Elements in Siebel Remote Architecture"
- "Siebel Remote Flow Diagram" on page 16

Elements in Siebel Remote Architecture

Figure 1 illustrates major elements in the Siebel Remote architecture.

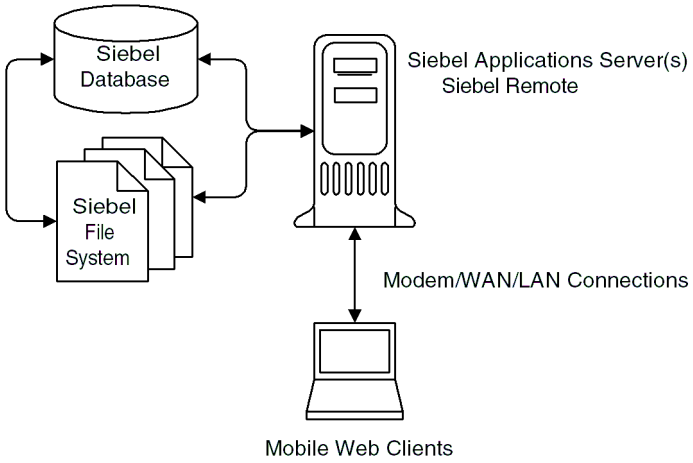

Figure 1. Siebel Remote Hardware Architecture

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 24.

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 Territory 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 is 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, or WAN. 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) that normally operates as a stand-alone unit, with a portion of the server database downloaded to a local database. This technique provides efficient access to the user's specific data. Mobile Web clients periodically connect to, and communicate with, a Siebel Server using a modem across public telephone lines, LAN or a WAN.

The Siebel Mobile Web Client also runs inside a standard Web browser. However, layers of the Siebel eBusiness Applications architecture, including the local database, reside on the user's personal computer and execute business logic locally. This client requires installation and periodically connects to a server to synchronize the server and the local databases.

Siebel Remote Flow Diagram

Figure 2 illustrates the Siebel Remote flow process. Numbers in the diagram relate to explanations in Table 1 on page 17. The purpose of this diagram and table is to provide a general overview of the process.

After Table 1 on page 17, there are smaller diagrams for the two parts of the Remote flow process—data downflow (Figure 3 on page 21) and data upflow (Figure 4 on page 22).

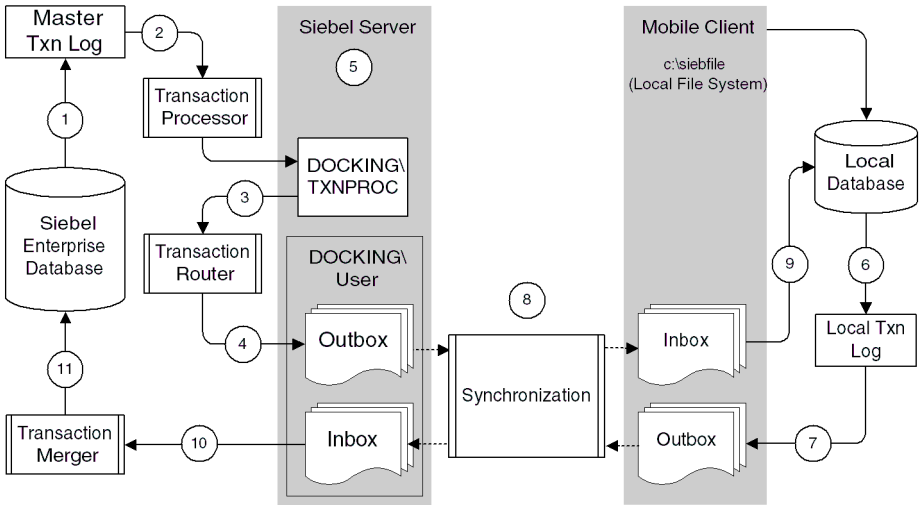

Figure 2. Siebel Remote Flow Diagram

Table 1. Steps in the Siebel Remote Flow Diagram

Step	Explanation of the Diagram
1	■ Every action in the Siebel database is considered a transaction. ■ These include adds, deletes, updates, merges, and so on. ■ A copy of each transaction is stored in the Master Transaction Log (S_DOCK_TXN_LOG). ■ 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*. ■ 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. ■ 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. ■ 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	■ The first time Remote Administrator 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. ■ It creates an inbox and an outbox. ■ The outbox stores any future extracts as well as DX files from Transaction Router. ■ 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. ■ 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 1. 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. ■ This server contains the Docking Directories for remote users to transfer the files involved with the synchronization process. These files include: ■ New database templates (from the Generate New Database task) ■ Database extracts (used with the templates to initialize the local DB) ■ DX files (.dx) ■ TOC files (.toc) ■ Visibility data for the Remote clients (visdata.dbf and dobjinst.dbf)
6	■ Remote clients process transactions in their local database while in disconnected mode. ■ 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. ■ Synchronization Manager must be running and will authenticate the Remote client, based upon the type of authentication in the Component Parameters. ■ The synchronization process includes handling communication between the mobile Web clients and the file system. ■ 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. ■ 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. ■ The Remote client begins applying the DX files to the local database when it completes the file exchange with the server.

Table 1. Steps in the Siebel Remote Flow Diagram

Step	Explanation of the Diagram
10	■ Transaction Merger, a component on the server side, pulls the DX files from the inbox within the Docking Directory.
11	■ Transaction Merger applies the changes to the server. ■ It also identifies conflicts. ■ 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. ■ Conflicts will be communicated to the Remote user during the next synchronization. ■ Transaction Merger deletes the DX files from the inbox within the Docking Directory.

Table 2 lists notes on the Siebel Remote flow process (Figure 2 on page 16).

Table 2. Siebel Remote Flow Diagram Notes

Topic	Comments
Remote Configuration File	■ The DockConnString is the logical Siebel server name. This represents the server where a mobile client was extracted. It should be populated when the client is first initialized. ■ 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. ■ 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. ■ 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; -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. ■ In Siebel 7.5, a new feature allows users to 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 more details about this topic, see *Siebel Mobile Web Client Implementation Guide*.

Table 2. Siebel Remote Flow Diagram Notes

Topic	Comments
Authentication	A Synchronization Manager component parameter—you identify the type of authentication used during synchronization. Options include: ■ **Siebel.** Synchronization Manager will compare the password used to login to the local db to the Sync Password stored in a Siebel table within the server database. ■ **Siebel/db password.** Synchronization Manager will compare the password used to login to the local db to the mobile Web client's user name and password to connect to the server database. ■ **Operating system password (Win 2000).** Synchronization Manager will compare the password used to login to the local db to the operating system password for the user on the application server.
Synchronization Frequency	Siebel Systems recommends that mobile users synchronize against the server at least daily. There is an automatic mechanism to help enforce frequent synchronization. Users can invoke autosynchronization on their laptops. For more detail about this topic, see "Using Autosynchronization" on page 83.

Figure 3 illustrates the Remote downflow of data. See Table 1 on page 17 for explanation of the numbered steps.

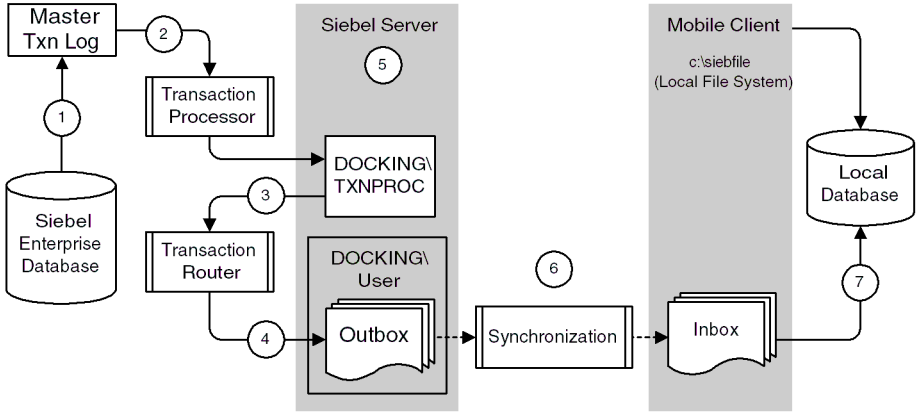

Figure 3. Remote Down Flow Diagram

Figure 4 illustrates the Remote upflow of data transactions. See Table 1 on page 17 for explanation of the numbered steps.

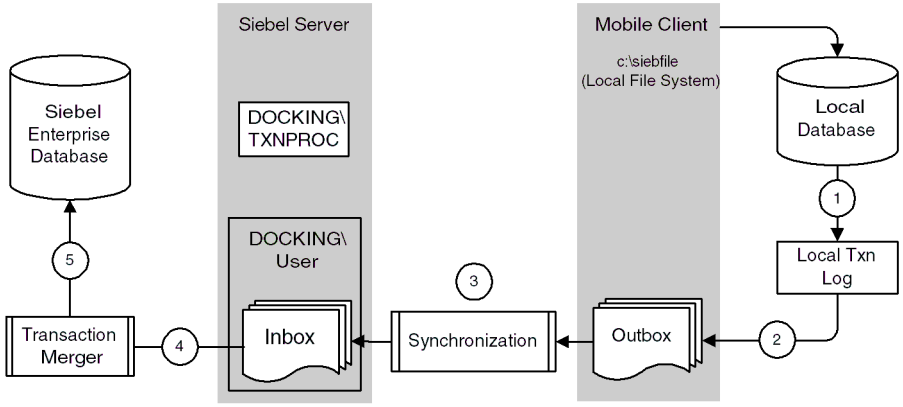

Figure 4. Remote Up Flow Diagram

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 24
- "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
- Master transaction log table that stores every net-change since the last database extraction

CAUTION: To avoid transaction problems between the database and the mobile Web client, make sure the same encoding method is used for both the database and the client. For example, do not use a multi-byte character client when running against a single-byte database.

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: 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. 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 the local database must be deleted before the these clients can be reinitialized.

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 Server Administration Guide, MidMarket Edition*.

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 Auto-Synchronization functionality also uses this method to perform synchronization when it starts a synchronization session

The Siebel Remote client uses the TCP/IP protocol to communicate with the Siebel Remote server over a modem using dial-up networking, a local area network or a wide area network.

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. Do not rename or delete DX files.

NOTE: Transactions are not lost after a client reextract.

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"
- "Generate New Database" on page 28
- "Database Extract" on page 29
- "Synchronization Manager" on page 29
- "Transaction Processor" on page 30
- "Transaction Router" on page 30
- "Transaction Merger" on page 31

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 Tech 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 password (if 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.

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 7.5 some Dock Objects were added, and some were dropped from those included in Siebel 7.0.3. 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.

- "Registering a Mobile User"
- "Generating a Database"
- "Extracting a Mobile Web Client" on page 33
- "Initializing the Local Database" on page 33
- "Synchronizing a Mobile Web Client" on page 36
- "How Changes Are Propagated to and from a Mobile Web Client" on page 38
- "Synchronizing a Mobile Web Client Machine" on page 44

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, MidMarket Edition*.

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. Generate a new 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 74.

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.

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's name, 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 started from the Siebel Mobile Web Client, Siebel Remote uses the username and password entered by the user in the Siebel Mobile Web Client.
- **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 password.
- **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 they are not 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.

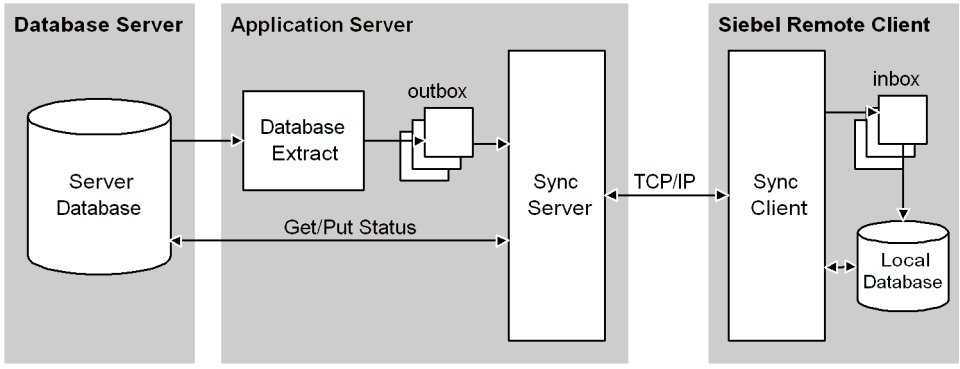

Figure 5. Initialize Local Database

For additional details on initializing the mobile Web client database, see "Initializing a Mobile Web Client Database" on page 121.

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 Client Installation and Administration Guide*.

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.

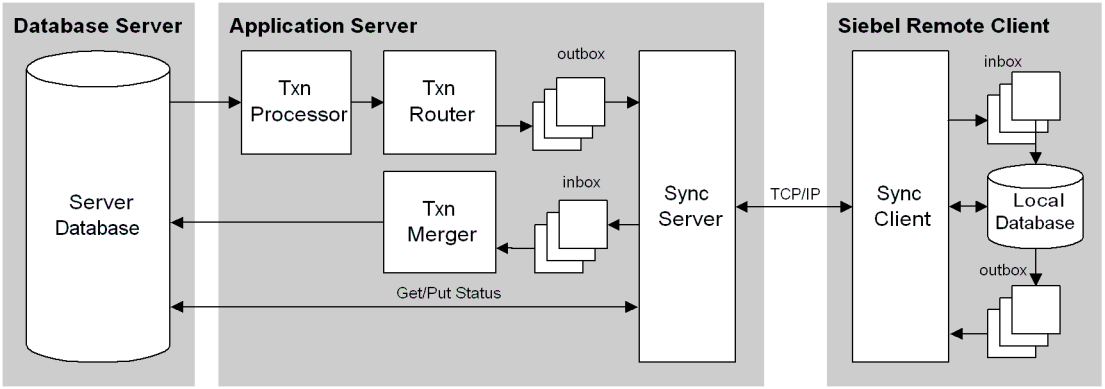

Figure 6. Synchronizing a Mobile Web Client (Routing and Merging)

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 another 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 password.

- **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 "Initializing the Local Database" on page 33. 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.** After disconnecting, 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.
- **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.

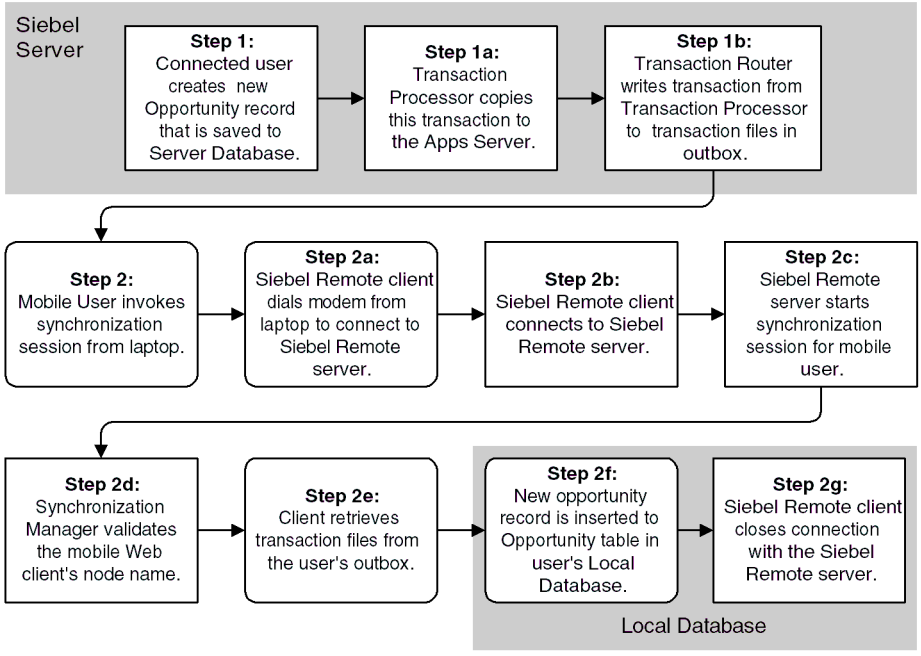

Figure 7. Process Flow for Changes by Connected Users

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.

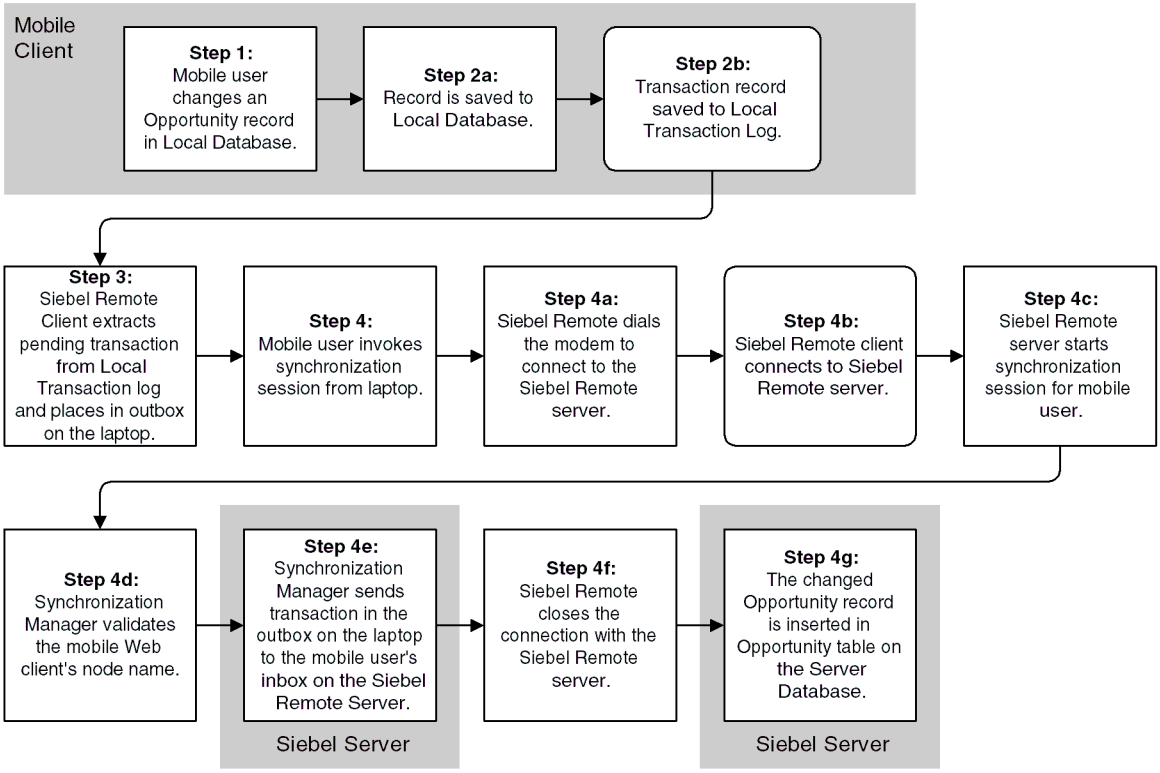

Figure 8. Process Flow for Changes by Mobile Users

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.

Synchronizing a Mobile Web Client Machine

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.

To synchronize a mobile Web client

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.

   See *Siebel Web Client Administration Guide, MidMarket Edition* for options.
4. Click Synchronize.

Security and Authentication

This section discusses security and authentication for mobile users.

■ "Authentication During Synchronization with Server Database"

■ "Authentication During Normal Operations" on page 46

Authentication During Synchronization with Server 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. 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. Verifies that the mobile Web client has connected to the correct Siebel Remote server. If the mobile Web client connects to the wrong Siebel Remote server, the Synchronization Manager reconnects the mobile Web client to another Siebel Remote server and updates the client's local configuration information.

Password. Validates the mobile Web client's password by using one of the following authentication methods, represented by the parameter name. The Siebel administrator specifies the password authentication method in the Siebel Server Component Parameters for Synchronization Manager, and in the appropriate CFG file located at the mobile client.

- **None.** Does not validate the mobile Web client's password. This is the default.
- **Database.** Uses the mobile Web client's user name and password to connect to the server database.

NOTE: You cannot use the Database authentication model if you have enabled password encryption. This would require the mobile user to use the encrypted password to log into the local database. For more information about password encryption, see *Siebel Server Administration Guide, MidMarket Edition*.

- **Siebel.** Verifies that the password is the same password stored in a table in the server database.
- **AppServer.** Verifies that the password is the operating system password for the user on the application server.

Authentication During Normal Operations

Another situation where mobile users may use password authentication occurs when they are connecting to the local database, which is the normal operation for mobile Web clients. The authentication process checks the user password set during client initialization against the Local database.

CAUTION: The Mobile web client does not support third-party or customer proprietary authentication methods.

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 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.

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.

Duplicate Conflicts

Although a user may successfully add a database row to a client database (or the server database), the added transaction may duplicate a new entry in another database residing elsewhere and not yet processed by the server. Siebel Remote determines that a 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 a system internal flag indicating that the row is a duplicate. The mobile user can determine whether a duplicate conflict has occurred.

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. To resolve a duplicate 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 54 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.

Implementing Siebel Remote Server 2

This chapter describes actions necessary to prepare the Remote server for use in the production environment. Table 3 includes these tasks.

Table 3. Tasks for Initial Siebel Remote Implementation

Task	Where Performed
Setting Siebel Remote System Preferences on page 54	Siebel client application
Disabling Local Access to All Views on page 62	Siebel client application
Starting Siebel Remote Server Components on page 63	Siebel Server Manager
Enabling Windows NT Rights on Siebel Remote Server on page 72	Siebel Remote server
Changing the Local Database Administrator Password on page 73	Siebel Remote server
Generating a New Database Template on page 74	Siebel Server Manager

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, please refer to the topic about controlling access to information in *Siebel Tools Reference, MidMarket Edition*.

To set Siebel Remote system preferences

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

Table 4 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 4. Siebel Remote System Preferences

System Preference	Default Value	Other Values
MRG: System Conflict Resolution	Server Wins	Client Wins
MRG: Docking Timestamp Source	Client Transaction Time	Server Database Merge Time
Docking: Transaction Logging	TRUE	FALSE

Table 4. Siebel Remote System Preferences

System Preference	Default Value	Other Values
Docking: Get All PostnCon rows	FALSE: Get only S_POSTN_CON rows on primary contact team member, the mobile Web client (or subordinate) position for a contact who is visible to the mobile Web client.	TRUE: Get all the S_POSTN_CON rows for a contact who is visible to the mobile Web client.
CSM Logging	FALSE	TRUE
MRG: Txns Per Commit	50	10 for DB2 10 for Oracle 1 for Microsoft SQL Server
MRG: User Friendly Notification	Conflicts	TRUE FALSE
LOGMGR: Vis Rules Per Statement	50	20 for DB2 20 for Microsoft SQL Server 20 for Oracle
DBX: Vis Rules Per Statement 1	20 for DB2 20 for Microsoft SQL Server 20 for Oracle	

Table 4. Siebel Remote System Preferences

System Preference	Default Value	Other Values
DBX: Vis Rules Per Statement N	1 for DB2 1 for Microsoft SQL Server 1 for Oracle	
Enterprise DB Server Code Page	Value is set by the database installer.	■ utf-8 ■ utf-16 (includes MS SQL and DB2 only) ■ cp1252 ■ cp932 The following are for development and migration only: ■ cp847 ■ cp936 ■ cp949 ■ cp950 ■ cp1250 ■ cp1251 ■ cp1253 ■ cp1254 ■ cp1255 ■ cp1256 ■ cp1257 ■ cp1258

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 (the default) - Updates from a server will take precedence and overwrite those already on the mobile Web client.

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

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.

Docking: Transaction Logging

This preference enables or disables docking transaction logging:

- Enable transaction logging by setting the value to TRUE.
- Disable transaction logging by setting the value to FALSE.
- The default is TRUE. This value is case sensitive.
- If you change this value from TRUE to FALSE and then change it back to TRUE while mobile users are active, you must reextract and reinitialize the local databases.
- Set this value to TRUE when performing a database extract. Transaction logging must be enabled during extraction to log changes to the local database.

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

NOTE: To disable transaction logging, restart the Siebel server.

When using the Territory 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.

Docking: Get All PostnCon rows

Tells the Transaction Router and Database Extract components whether or not to get all records in the S_POSTN_CON table for a contact. This parameter is designed to improve Siebel Remote performance. The default value is FALSE, which means "Only get the row for the mobile user's position."

Use this parameter with caution, because sometimes S_POSTN_CON table can become quite huge on a local database. If you do change this parameter, you need to reextract mobile users. Also, you need to restart transaction routers.

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.

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 4 on page 54.

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.

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 4 on page 54.

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

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 4 on page 54.

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 4 on page 54.

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

Enterprise DB Server Code Page

Siebel 7.5 supports non-Unicode code pages 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 4 on page 54.

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, select View > Site Map > Application Administration > 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 any of the responsibilities assigned to this user.

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:

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

  Values are from "Start-Up Parameters for Transaction Processor" on page 65.

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 Server Administration Guide, MidMarket Edition*.

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, navigate to Site Map > Server Administration > Servers > Server Components tab, and start the component. A new task should be started.

To start Transaction Router – srvrmgr command line

- From the srvrmgr command line, enter:

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

  Values are from "Start-Up Parameters for the Transaction Router" on page 67.

To start Transaction Router – 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 at least one task when the Siebel Server is started if the component is enabled. If the component is not started, navigate to Site Map > Server Administration > Servers > Server Components tab, and start the component. A new task should be started.

To start Transaction Merger – srvrmgr command line

- From the srvrmgr command line, enter:

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

  Values are from "Start-Up Parameters for Transaction Merger" on page 69.

To configure Transaction Merger – 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 at least one task when the Siebel Server is started if the component is enabled. If the component is not started, navigate to Site Map > Server Administration > Servers > Server Components tab, and start the component. A new task should be started.

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 Server Administration Guide, MidMarket Edition*.

Transaction Processor

Table 5 shows the start-up parameters for the Transaction Processor.

Table 5. 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.

Table 5. Start-Up Parameters for Transaction Processor

Name	Alias	Required/Optional	Comments and Default Value
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.
TS Cache Size	TSCacheSize	Optional	Cache size for dobjinst.dbf (visibility database) in kilobytes. The default is 4096 kilobytes. 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 6 shows selected parameters for the Transaction Router.

Table 6. 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.
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. 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. Please do not modify this parameter without approval from Expert Services or Technical Support.

Table 6. Start-Up Parameters for the Transaction Router

Name	Alias	Required/Optional	Default Value and Usage Notes
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. 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.
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 7 shows selected parameters for the Transaction Merger.

Table 7. 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: ■ STOP. Stops Transaction Merger. ■ IGNORE. Causes Transaction Merger to sleep if all nodes were ignored in an iteration. ■ 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 8 shows the start-up parameters for the Synchronization Manager.

Table 8. 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. ■ **None.** Does not validate the mobile Web client's password. ■ **Database.** Uses the mobile Web client's user name and password to connect mobile Web clients to the server database. ■ **Siebel.** Validates the mobile Web client's password against the password stored in the mobile Web client's screen. ■ **AppServer.** Validates the password against the password on the Siebel Server operating system. The mobile Web client's NT password cannot contain lowercase letters.
Domain Name	NTDomain	Optional	If you are using AppServer authentication, this parameter specifies the name of the Windows NT domain for the user name and password. To use Windows NT password authentication, you must enable the Windows NT 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.
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.

Table 8. Selected Synchronization Manager Start-up Parameters

Name	Alias	Required/Optional	Default Value and Usage Notes
Synchronization Port Number	SyncPort	Optional	The TCP/IP port number dedicated to the Synchronization Manager. Default value is 40400. Use the Siebel Server Administration screens to override the default value by specifying a value and restarting the Siebel Server. 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 mobile Web clients.
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.

Enabling Windows NT Rights on Siebel Remote Server

The Administrator will need to enable Windows NT rights for the sadmin user on the Application servers. This section describes how to do this. See *Siebel Server Administration Guide, MidMarket Edition* for more information.

This procedure is generally required for any Siebel server, not just Siebel Remote.

To enable the Windows NT user rights on the Siebel Remote server

1. From the application-level menu, choose Start > Programs > Administrative Tools > User Manager.
2. Choose Policies > User Rights.
3. Select the Show Advanced User Rights check box and then Act as part of Operating system (the Grant To field shows the current owner).
4. Click Add.
5. In the Add Users and Groups dialog box, click Show users.
6. Scroll to the appropriate name in the Names list, select the name, and click Add.

   You only need to add the Act as part of Operating system user right to the user who will start the Siebel Remote server. You can add rights for domain users, such as HQ\Smith, or you can add local users.
7. Click OK and exit User Manager.
8. Log out and reboot to activate the new rights.

Changing the Local Database Administrator Password

In some cases, you may want to change the local DBA password on mobile Web clients. To accomplish this task, you should change the DBA password in the database template file before generating the new database template.

The following procedure shows how to accomplish this task in the SQL Anywhere environment.

To change the local DBA password on mobile Web clients

1. In a DOS window on the server machine, run the Interactive SQL utility: dbisqlc.exe.
   a. At the DOS prompt, change to the \bin subdirectory within the Siebel server root directory:

      `cd \siebel_server_root\bin`

   b. Run dbisqlc.exe by entering:

      `dbisqlc -c "UID=DBA;PWD=SQL;DBF=siebel\dbtempl\my_templ.dbf"`

2. In the Command window, enter:

   `grant connect to user_id identified by new_password`

   For example, to set a new password of `SQL2` for the user `DBA`, enter:

   ```
   grant connect to DBA identified by SQL2
   ```

   NOTE: You must use uppercase for every password in SQL Anywhere.

3. Click Execute.
4. Run the Generate New Database component using the new DBA password.

   For information on running the Generate New Database component, see "Generating a New Database Template" on page 74.

5. Run a Database Extract for mobile Web clients, and notify mobile users to initialize their databases.

   For information about initializing a local database, see "Initializing a Mobile Web Client Database" on page 121.

Generating a New Database Template

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 columns or extension indexes to the default Siebel schema. In this case, run the Database Template Utility each time you click the Apply button in the Siebel Schema screen.

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.

Creating a New Empty Database File

Beginning with Siebel 7.5, Siebel applications will only support a Unicode local database. Thus, only the Unicode database template file sse_utf8.dbf is shipped with the application. With this database template, every collating sequence can be supported. The sorting sequence for the application is configured using a parameter SortCollation in the CFG file. If the parameter is not specified, the application will sort using the Unicode multilingual collation order. No other character set is supported for the local database.

Running the Generate New Database Component

You should always use an SQL Anywhere transaction log file when creating the database template. The SQL Anywhere transaction log significantly reduces the time required to create the database template.

Use the Siebel Server Manager to run the Generate New Database component.

To generate a new database template (GenNewDb)

1. From the application-level menu, choose View > Site Map > Server Administration > Enterprise Operations.
2. Click the Component Requests tab.
3. Click New.
4. In the Component /Job field, select Generate New Database.
5. Specify the server name in the Server field.
6. In the Component Request Parameters list, click New.
7. In the Name field, select SQL Anywhere Database from the pick list.

   The Value field typically defaults to sse_utf8.dbf and appears automatically.
8. Modify the values of other parameters as necessary by repeating Step 6 and Step 7.

   Table 9 on page 76 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.
9. In the Component Requests form, click the menu button and then click Submit request.

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

Table 9 shows the start-up parameters for generating new database components.

Table 9. Selective 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. Default is sse_utf8.dbf. Set the DbfFile parameter to the name of the empty database template file you created. The empty database template file must be located in the siebel\dbtempl directory.
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.
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.
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. If the value is set to TRUE, it means the schema is read from the DDL file. 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.

Table 9. Selective Parameters for the Generate New Database Component

Name	Alias	Required/Optional	Default Value and Usage Notes
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 utility. This utility 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.

Setting Up Mobile Web Clients 3

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 10. Tasks for Enabling Mobile Web Clients

Task	Where Performed
Setting Up Mobile Web Client Hardware and Software on page 80	Mobile Web client
Enabling Network Connectivity on page 80	Mobile Web client
Establishing Autodial Preferences on page 80	Mobile Web client
Setting Synchronization Preferences on page 81	Mobile Web client
Using a Different Data Source on page 87	Mobile Web Client
Registering a Mobile Web Client on page 88	Siebel Server Manager
Using Routing Models on page 93	Siebel Server Manager
Limiting Views Available to Mobile Web Clients on page 107	Siebel Server Manager
Creating Mobile Web Client User Accounts and Privileges on page 109	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 Web Client Administration Guide, MidMarket Edition*.

Enabling Network Connectivity

Install the necessary hardware and software on the mobile Web client to allow the this 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 Web Client Administration Guide, MidMarket Edition*.

Establishing Autodial Preferences

If using Microsoft dial-up networking, mobile users 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 first define your phone book entries using Microsoft Dial-Up Networking for Windows 2000, or NT.

Setting Synchronization Preferences

This section describes synchronization preferences. The Siebel Remote client reads configuration parameters in the Siebel configuration file (default is siebel.cfg) to specify the location of the Siebel Server directories, the Siebel File Server directories, and the Siebel Database installation. If a different CFG file is used, the shortcut created by the Installer should be changed.

Before using Siebel Remote, you must set the values for the configuration parameters. The Siebel installation utility creates a siebel.cfg 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 7.5 includes an autosynchronization option to help maintain the frequency of synchronization. Frequent synchronization by mobile users can improve the performance of the system. After initializing their local database, mobile users enable or disable autosynchronization from User Preferences > DB Synchronization. For more information, see "Using Autosynchronization" on page 83.

Synchronization Parameters

The configuration parameters that Siebel Remote uses for synchronization include:

- **ClientRootDir.** Name of the Siebel client installation directory.
- **DockConnString.** Logical network address of the Siebel Server to which you want to connect for synchronization. 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 this is the environment, the port for synchronizing with your server must be opened on your firewall. Also make sure your Internet Service Provider (ISP) does not block communication to this particular port. This parameter has the following format:

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

  where:

  - *logical 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.
  - *sync port #* = the TCP/IP port number dedicated to the Synchronization Manager. The default value is 40400 if nothing is specified.

    NOTE: You can use the Siebel Server Administration screens to override the default value by specifying a value for the Synchronization Manager (SynchMgr) component's SyncPort parameter and 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, the CFG file must also be updated.

- *service* = the TCP/IP service you are requesting. SMI is the only valid value and is the default value.
- *encryption* = the encryption package you are using. The encryption facility must match the type used by the server. Both MSCRYPTO and RSA are supported.

Examples of valid values are:

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

SIEBAPP1:TCPIP: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.
- **TableOwner.** Name of the database account on the local database where the Siebel schema is installed. Default is SIEBEL.
- **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.

Using Autosynchronization

The purpose of autosynchronization 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 Autosynchronization Agent runs in the background at scheduled times to perform automatic synchronization when connected to the network.

Follow the procedure below to invoke autosynchronization and the available options. One of the options is a synchronization reminder that prompts the user to synchronize if a specified period passes without a synchronization session.

To enable autosynchronization

1. Verify that the local database is initialized.
2. Verify that the Siebel Autosync 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 Auto Synchronization agent to start when the computer boots up.
3. Log in to the local database, and navigate to User Preferences > DB Synchronization.
4. Under Auto-Synchronization, define your options using the information below.

Field	Description
Enable Auto-Synchronization check box	Required to enable autosynchronization.
User Confirmation check box	Optional.
Maximum Retries drop-down list	Defaults to zero, but 2 or 3 is recommended.
Maximum Network Latency drop-down list	Sets a network latency that will prevent the autosynchronization agent from invoking a synchronization session. Required. The administrator establishes this policy. For example, at 56K the threshold may be 200 to 300 milliseconds or higher.

5. Under Synchronization Frequency, select a frequency from the Synchronization Frequency drop-down list, then use the other fields to further define the frequency.

6 Synchronization Frequency drop-down list values:

Option	Description
Default	Default is Empty. Same behavior as if autosynchronization is disabled.
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 Start up	Perform synchronization after the Mobile Web Client is started. If no network connection is available, try again the next time the client is started.
Hourly	Perform the synchronization every hour at the specified time increment chosen in the *Hourly at* picklist. 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. 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.

7 In the Synchronization Reminder form, select the Enable Synchronization Reminder check box and select a minimum number of days from the Min Days Between Sync Sessions drop-down list to display the Synchronization Reminder dialog box when the mobile Web client is up and if the last synchronization was earlier than the “Minimum days between sync sessions.”

This will remind the user to perform synchronization. The reminder dialog will appear with the following text: “Perform database synchronization now?” The user can respond accordingly.

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 server parameter to MSCRYPTO or RSA. You can use the Server Manager command line or the Server Administration screen.
- For the client, encryption is the fifth parameter in the DockConnString. 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.

Using a Different Data Source

You can use a different data source from the one specified in the [Local] section of the CFG file. Use the following procedure to do this.

To use a different data source

1. In the [DataSources] section of the CFG file, add a statement for the new data source name.

   For example,

   ```
   NewDSN = NewDSN
   ```

2. Under the [Local] section, create a new section for the new data source.

   You can copy the [Local] section and rename the section. For example,

   ```
   [NewDSN]
   ```

3. Update the new section as appropriate and adjust the ConnectString parameter as necessary to read:

   ```
   ConnectString=ClientRootDir\NewDSN\sse_data.dbf  -q -m -x NONE -
   gp 4096 -c40m -ch60m
   ```

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. 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:

- Client Status view. Helps mobile Web clients resolve data merge conflicts on their local databases by showing conflict information after synchronization.
- Mobile User Summary view. Contains information about the mobile client.
- DB Synchronization view. Auto Synchronization view. Allows users to enable and configure autosynchronization after initializing their local database, mobile users enable or disable autosynchronization.

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

To register a new mobile Web client on a node

1. Navigate to Site Map > Siebel Remote Administration > Mobile Clients.
2. From the Parent Server list, select the node.
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 93.

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 data routing model assigned to this mobile user is Sales Representative, Analyst, Minimal Data, or Sales Manager, navigate to Applications Administration > Responsibilities > Responsibilities list, 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 on page 89. 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 107.

a In the Users list, add a new record.

b In the Add Users Selection 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 92

For additional information about this topic, see *Siebel Tools Reference, MidMarket Edition*.

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. There are about 1,500 active routing rules in Siebel 7.5. 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.

You can see these routing rules in *Siebel Tools Reference, MidMarket Edition*. However, you will not be able to update or modify these rules. Any such changes will 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.

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 107.

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.

Field Sales Representative Standard

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

The views associated with this model include the following:

- Accounts
- Activities / Calendar
- Activity Plans
- Contacts
- Competitors
- Correspondence
- Currencies
- Employees
- Expense Reports
- Opportunities

- Periods
- Sales Methods
- Sales Stages

Field Sales Manager Standard

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 listed above for the Sales Representative Standard in addition to the manager's Team's Accounts, Contacts, and Opportunities.

Field Technician

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. The views associated with this model include the following:

- Account
- Action
- Activities
- Asset Mgmt - Asset
- Contact
- Employee
- Employee Skill
- Employee Skill Item
- Entitlement Account
- Internal Product
- List of Values

- Order Entry-Orders
- PS Project Team
- Salutation (eApps)
- Service Agreement
- Service Request
- Session Action
- Session Detail
- Session Log
- SI Account External
- SI Employee External
- SRF Screen
- SRF Screen View
- Time Sheet Daily Hours
- User Preferences

Senior Field Engineer

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 data, access to solution knowledge bases, basic sales functionality, and customer and asset data. The views associated with this model include those in the following list:

- Account
- Account External Product
- Account Note
- Account Private Note

- Business Address
- Action
- Activities
- Asset Mgmt - Asset
- Calendar Access
- Contact
- Contact Note
- Contact Private Note
- Employee
- Employee Skill
- Employee Skill Item
- Entitlement Account
- Expense
- Expense Item
- Expense Approver list
- Expense Unassoc Item
- Internal Product
- List Of Values
- Opportunity
- Order Entry - Orders
- Order Entry - Line Items
- Order Entry - Line Item Actions
- Personal Payment Profile
- Price List

- PS Rate List
- PS Position Type
- PS Project Team
- Response
- Salutation (eApps)
- Service Agreement
- Service Agreement Metric
- Service Request
- Service Request Attachment
- Session Action
- Session Detail
- Session Log
- SI Account External
- SI Employee External
- Shift Hour
- Solution SR/PD
- SRF Screen
- SRF Screen View
- Time Sheet
- Time Sheet Daily Line
- Time Sheet Unassoc Daily Hours
- Time Sheet Approver
- Time Sheet Daily Hours
- User Preferences

- Work Type
- Contact(All)
- Action Attachment
- Messaging
- Related Service Request
- Account Attachment
- Quote
- Project
- Position
- Order Entry Attachment
- Asset Mgmt - Asset Relationship
- Asset Mgmt - Asset Transaction
- Asset Shared Note
- Asset Private Note
- Asset Mgmt - Asset (sub-components)
- Product Defect
- Asset Mgmt - Asset Attachment
- Contact Attachment
- Contact Relationship
- Solution
- SR Resolution Item
- Related Solution
- Sales Tool
- Key Feature

- Sales Tool by Product
- Product Comparison
- Component Product
- Quote Item
- Service Region
- Zip Code

Field Engineer

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. Debriefing is the reporting of material usage, time, and expenses to service managers after completing an activity.

The views associated with this model include:

- Account
- Account External Product
- Account Note
- Account Private Note
- Business Address
- Action
- Activities
- Asset Mgmt - Asset
- Calendar Access
- Contact
- Contact Note

- Contact Private Note
- Employee
- Employee Skill
- Employee Skill Item
- Entitlement Account
- Expense
- Expense Item
- Expense Approver list
- Expense Unassoc Item
- Internal Product
- List Of Values
- Order Entry - Orders
- Order Entry - Line Items
- Order Entry - Line Item Actions
- Price List
- PS Rate List
- PS Position Type
- PS Project Team
- Response
- Salutation (eApps)
- Service Agreement
- Service Agreement Metric
- Service Request
- Session Action

- Session Detail
- Session Log
- SI Account External
- SI Employee External
- Shift Hour
- Solution SR/PD
- SRF Screen
- SRF Screen View
- Time Sheet
- Time Sheet Daily Line
- Time Sheet Unassoc Daily Hours
- Time Sheet Approver
- Time Sheet Daily Hours
- User Preferences
- Work Type

Consultant

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 Siebel projects functionality. Using this routing model, the data will be limited to time sheets, expense reports, limited project information, calendar and employee skills.The views associated with this model include the following:

- Action
- Action (Busy Free Time)
- Calendar Access

- Change Position
- Employee
- Employee Skill
- Employee Skill Item
- Expense
- Expense Approver
- Expense Hotel Item
- Expense Item
- Expense Mileage Item
- Expense Unassoc Item
- Opportunity
- Project
- Project Status
- Project Status Item
- PS Project Risk
- Salutation (eApps)
- SI Account External
- Time Sheet
- Time Sheet Approver
- Time Sheet Daily Hours
- Time Sheet Daily Line
- Time Sheet Unassoc Daily Hours

Analyst

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. The views associated with this model include the following:

- Contacts
- Contact entitlements
- Agreements for Accounts
- Agreements for Contacts
- Orders for an account
- Assets
- Quotes for an account
- Activities for orders
- Activities for quotes
- Activities for assets
- Activities for opportunities
- Payments
- Service Requests
- Customer Surveys
- My Opportunities
- Price Lists
- Expense Reports
- Expenses

If on the Project Team:

- ProjectItems
- TeamWorkbook
- Time & Expenses
- TimeSheets
- Expenses

Minimal Data

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.

The views associated with this model include the following:

- My Contacts More Info View
- My Contacts Addresses View
- My Daily Calendar
- My Monthly Calendar
- My Weekly Calendar
- My Accounts More Info View
- My Opportunities More Info View
- All Employees View

Mobile Partner User

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

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

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

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.

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 can limit views for mobile users with the following out-of-the box routing models: Sales Representative, Analyst, Minimal Data, and Sales Manager.

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 Applications Administration screen. For more information about this topic, see "Using the Responsibilities View" on page 108.

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.

Responsibilities with the Routing Model suffix include all the views that are *not* available to the user assigned to the corresponding routing model. This prevents the application from displaying views for a mobile user where there is no data.

Another mobile user, with or without the same responsibilities, may have a different set of views turned off. Therefore, administrators can turn off one set of views for one mobile user and a different set for another mobile user. By limiting the views on a user-by-user basis, administrators can further optimize the size of local databases.

NOTE: A particular user may have two or more responsibilities that include the same view. If Local Access = False (unchecked) 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.

The corresponding 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 Applications Administration 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: Assign users to routing models in the Remote Administration > Mobile Clients view.

- 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. Navigate to Applications Administration > Responsibilities > Responsibilities list, 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 Step 3 and Step 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 Applications Administration 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 above.

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, you need to create accounts for each client depending on the authentication method. This section describes the passwords for the three authenticating methods available.

- If you use AppServer authentication, you must create a user account for each mobile Web client on the server. Examples include:
  - If the Remote server is running in the NT environment, create an NT user and password for each mobile Web client you defined.

    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 invalid characters (/ \ : * ? “ < > |) in a DOS file naming scheme.

  - 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 logon and password for each mobile Web client. After the local database password is set up at the time of initialization, users cannot change it.

  CAUTION: If the user changes the password on the server (in the connected mode), that user will have two different passwords: one for the server and one for the local database. This will cause database password authentication to fail.

- If you use the Siebel authentication method, you must set the password for each mobile Web client on the Mobile Clients view.

In these cases, you must provide the password to mobile users so they can use it to synchronize with the Siebel Server. If you do not use authentication, then you do not need to create any of these accounts or passwords.

To create an NT user and password for a mobile Web client

- See "To enable the Windows NT user rights on the Siebel Remote server" on page 72 and follow the instructions.

Extracting Databases for Mobile Web Clients 4

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 11 shows these tasks. You must repeat each of these steps, for every mobile Web client.

Table 11. Tasks for Enabling Mobile Web Clients

Task	Where Performed
Database Extraction for a Mobile Web Client on page 112	Server
Initializing a Mobile Web Client Database on page 121	Mobile Web client
Enabling the Stand-Alone Synchronizer on page 123	Mobile Web client
Viewing Session Details on Mobile Web Clients on page 125	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 Siebel 7.5 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.

Multiple instances of the database extract components can be run simultaneously. Before running a database extract for a client, you must make sure that your organization's reporting hierarchies are updated. Use the Position Administration view from the Application Administration screen 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, MidMarket Edition*.

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 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 View > Site Map > Server Administration > Enterprise Operations.
2. Click the Component Requests tab.
3. In the Component Request form, click New.
4. In the Component/Job field, select Database Extract from the picklist.

5 In the Component Request Parameters list, 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 Component Requests form, click the menu button and then click Submit request.

   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.

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. Create the database template by navigating to Site Map > Server Administration > Enterprise Operations > Component Requests tab.
3. Create a new component request for Generate New Database.
4. Submit the component request.

   The task of creating a database template takes a while. To check the progress of this task, go to the Components View and check the information about the status of this task.

   Upon successful creation of a database template, you will see the Components View.

5 In order to extract databases for multiple users, create a text file called ExtractUserList.txt containing the user IDs of the new mobile users.

6 Navigate to Enterprise Operations > Component Requests tab and define a component request for Database Extract.

7 Enter the Component Request Parameters and submit the component request.

Once the task of creating database extract is completed, go to the directory d:\sea14010\siebsrvr\docking to see folders for AMARTIN, CCHENG, PSINGH and RMARLOW.

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.

When mobile Web clients initialize from a CD-ROM, a parameter in the mobile Web client's configuration file instructs the database initialization process to check the CD-ROM directory rather than the Siebel Remote server for files to download. This parameter is `FileSystem` and is located in the Local section of the client configuration file. See "Setting Siebel Remote System Preferences" on page 54 for more information.

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.

Before mobile Web clients can initialize from a CD-ROM or other media device, you need to add the `FileSystem` parameter to client CFG files. The `FileSystem` parameter directs the mobile Web client to the CD-ROM or other media device directory during the initialization process. For information on setting the `FileSystem` configuration parameter, see "Setting Synchronization Preferences" on page 81.

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 113.
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 12 on page 118 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.

Table 12 lists the parameters and default values for the database extract component.

Table 12. 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 enu_1252.
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.
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.

Table 12. Database Extract Parameters

Name	Alias	Required/Optional	Default Value and Usage Notes
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. 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. 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 12. 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. 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. The Transaction Router, Transaction Merger, and DB Extract processes on the old server stop servicing the mobile Web client 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 116.

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 123.

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 13 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 13. Stand-Alone Synchronizer Command-Line Options

Option	Definition	Required/Optional	Default Value and Usage Notes
/n	Client name	Required	Name of mobile Web client. The value must be entered using uppercase letters.
/u	User name	Required	Database connection user name. 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.
/c	Configuration file	Required	The default is siebel.cfg.
/d	Data source	Optional	The default is local.
/a	Autostart mode	Optional	Available modes are Y or N.
/v	Verbose mode	Optional	Available modes are Y or N.[1]
/i	Iterations	Optional	Sets the number of iterations.
/l	Language	Optional	Language to use for the docking session.
/sleep	Sleep time	Optional	Number of seconds to sleep between iterations.

Table 13. Stand-Alone Synchronizer Command-Line Options

Option	Definition	Required/Optional	Default Value and Usage Notes
/comm	Communication parameters	Optional	Used for modem connections.
/RecvFiles	Receive files	Optional	Determines whether mobile Web client will download files from server. 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.
/RecvPubFiles	Receive published files	Optional	Determines whether mobile Web client will download published files. Available modes are Y or 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.
/? /help	Help	Optional	Provides online help for usage.

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.

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 54. 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 Siebel Remote Administration 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.

5 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 14 addresses these tasks.

Table 14. Tasks for Administering Siebel Remote

Task	Where Performed	When Performed
Starting and Stopping Siebel Remote Server Components on page 128	Siebel Server Manager	Database start-up, Siebel Server start-up
Monitoring Siebel Remote Operations on page 132	Siebel Server Manager	Daily
Data Synchronization on page 144	Client Machine	Daily or more often
Sending Messages to Mobile Users on page 146	Siebel Remote server	As needed
Refreshing a Client Database on page 147	Siebel Server Manager	As needed
Deactivating and Reactivating a Mobile Web Client on page 148	Siebel Server Manager	As needed
Deleting a Mobile Web Client on page 149	Siebel Server Manager	As needed
Changing Routing Models on page 150	Siebel Server Manager	As needed
Adding New Mobile Users on page 151	Siebel Server Manager	As needed
How to Set Client-Side Logging on page 151	Siebel Server Manager	As needed
Event Tracing for Locking on page 155	Siebel Server Manager	As needed
Handling Failure and Recovery on page 157	Various	As needed

Starting and Stopping Siebel Remote Server Components

Use 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 Server 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 with Client=<ClientName>, SQLFlags=2,
TraceFlags=7
```

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, Database Extract, Parallel Database Extract, and Generate New Database.

4. From the GUI, synchronize the components by navigating to View > Site Map > Server Administration > Enterprise Configuration > Batch Component Admin.
5. Click Synchronize.

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`, `dbxtract`, `pdbxtract`, or `gennewdb`, depending upon the situation.

For additional information on starting and stopping server components using the Server Manager, see *Siebel Server Administration Guide, MidMarket Edition*.

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 the *visutl* utility on a mobile Web client when Transaction Router is processing that node.

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.

Monitoring Mobile Web Client Status

You can use views in the Siebel Remote Administration screen to monitor the status and progress of mobile Web clients.

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

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, select View > Site Map > Siebel Remote Administration.
2. From the Show drop-down list, select Client Status.
3. In the Mobile Clients 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 (bytes)	Total size in bytes for extracted records
# of files in File System	Number of files in file system for this client
File System (bytes)	Size of files in file system, in bytes
Max transaction	Maximum transaction ID when this client was extracted
Initialized on Client	
Last Session	Time when the local database initialization was done
Seconds	Time it takes to initialize the local database
Free disk (bytes)	Free disk space available on the mobile client's laptop drive
Database (bytes)	The size in bytes of the local database
# of files in File System	Number of files in local file system
File System (bytes)	Total size in bytes of files in local file system
Product version	System information for the machine where local database resides

Field	Description
Current status	
Last session	Time when the last synchronization
Seconds	Duration of the last synchronization was done
Free disk (bytes)	Free disk space available on the mobile Web client's laptop drive
Database (bytes)	Current size of the local database
# of files in File System	Number of files in the local file system
File system (bytes)	Current size in bytes of files in the local file system
Total session	How many 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 15 lists and describes the fields on the Siebel Remote Client Diagnostics form.

Table 15. 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
Last trxn duration (sec)	Time it took to process transaction in the latest session

Table 15. Fields on Siebel Remote Client Diagnostics Form

Field	Description
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, select View > Site Map > Siebel Remote Administration.
2. From the Show drop-down list, 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 16 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 16. 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, select View > Site Map > Siebel Remote Administration.
2. From the Show drop-down list, select Upload Statistics.
3. In the Mobile Clients list, select the mobile Web client that you want to monitor.

Table 17 gives descriptions of the fields on the Siebel Remote Upload Statistics form.

Table 17. 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, and applied on the server.
Last Transaction	Last transaction created on the client, received on the server, and applied on the server.
Last Session Transaction Count	Number of transactions created on the client, received on the server, and applied on the server during the last synchronization session.
Total Transactions	Total transactions created on the client, received on the server, and applied on the server since the last database extract.
Last Session Operation Count	Number of operations created on the client, received on the server, and applied on the server during the last synchronization session.
Total Operations	Total number of operations created on the client, received on the server, and applied on the server since the last database extract.
Last Session (Bytes)	Size of transactions created on the client, received on the server, and applied on the server during the last synchronization session.

Table 17. Fields on Siebel Remote Upload Statistics Form

Field	Description
Total Transferred (Bytes)	Size of transactions created on the client, received on the server, and applied on the server since the last database extract.
Last Session Duration (Sec)	Duration of last synchronization session created on the client, received on the server, and applied on the server.
Total Seconds	Total duration of all synchronization sessions since the last database extract.
Last Session Attachment File Count	Number of attachment files created on the client, received on the server, and applied on the server during the last synchronization.
Total Attachment Files	Number of attachment files created on the client, received on the server, and applied on the server since the last database extract.
Last Session Att File Size (Bytes)	Size of attachment files created on the client, received on the server, and applied on the server during the last synchronization session.
Total Attachment (Bytes)	Size of files created on the client, received on the server, and applied on the server since the last database extract.

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. Navigate to the Siebel Remote Administration screen.
2. From the Show drop-down list, select Download Statistics.
3. In the Mobile Clients list, select the mobile Web client that you want to monitor.

Table 18 describes the fields on the Siebel Remote Download Statistics form.

Table 18. 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, and applied on the client.
Last File	Last transaction file created on the server, received on the client, and applied on the client.
Last Transaction	Last transaction created on the server, received on the client, and applied on the client.
Last Session Transaction Count	Number of transactions created on the server, received on the client, and applied on the client during the last synchronization session.
Total Transactions	Total transactions created on the server, received on the client, and applied on the client since the last database extract.
Last Session Operation Count	Number of operations created on the server, received on the client, and applied on the client during the last synchronization session.
Total Operations	Total number of operations created on the server, received on the client, and applied on the client since the last database extract.
Last Session (Bytes)	Size of transactions created on the server, received on the client, and applied on the client during the last synchronization session.
Total Transferred (Bytes)	Size of transactions created on the server, received on the client, and applied on the client since the last database extract.
Last Session Duration (Sec)	Duration of last synchronization session.
Total Seconds	Total duration of all synchronization sessions since the last database extract of the node.
Last Session Attachment File Count	Number of attachment files created on the server, received on the client, and applied on the client during the last synchronization.
Total Attachment Files	Size of files created on the server, received on the client, and applied on the client.

Table 18. Fields on Siebel Remote Download Statistics Form

Field	Description
Last Session Att File Size (Bytes)	Size of the file attachments created on the server, received on the client, and applied on the client during the last synchronization session.
Total Attachment (Bytes)	Size of all file attachments created on the server, received on the client, and applied on the client since the last database extract of this node.

Monitoring Transaction Logs

Use the Siebel Remote Administration views to monitor transaction logs.

To monitor transaction logs

1. From the application-level menu, select View > Site Map > Siebel Remote Administration.
2. From the Show drop-down list, 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	ROW_ID of the transaction
Transaction updated	Timestamp showing when the transaction was last updated

The fields in the detail form are described below.

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 columns hold transaction data

3 To monitor transaction logs by operation, select Transaction Log By Operation from the Show drop-down list.

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 Show drop-down list.

The Transaction Log By Node view displays the number of transactions for each node. A node is a mobile Web client. 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. 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 Show drop-down list.

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, statistics, and the Server Manager, see *Siebel Server Administration Guide, MidMarket Edition*.

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 16 and "Synchronizing a Mobile Web Client" on page 36.

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.

Auto Synchronization

Auto-Synchronization functionality performs database synchronization automatically. Once enabled and configured by the mobile user, the Auto-Synchronization 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 autosynchronization, see "Using Autosynchronization" on page 83.

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 54 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 109.
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 121.

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, select View > Site Map > Siebel Remote Administration.
2. From the Show drop-down list, select Mobile Clients.
3. In the Parent Server form, use the record navigation buttons 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 Save.

   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 Save.

   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, select View > Site Map > Siebel Remote Administration.
2. From the Show drop-down list, select Mobile Clients.
3. In the Parent Server form, use the record navigation buttons 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 Save.
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 27 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, select View > Site Map > Siebel Remote Administration.
   b. From the Show drop-down list, 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 109.
4. Prompt the mobile user to synchronize.

   The mobile user can download the new extract at the beginning of the synchronization session.

   It will reinitialize the mobile Web client database again. See "Initializing a Mobile Web Client Database" on page 121.
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 88.
2. Perform a database extract for the new mobile Web client. For details, see "Creating Mobile Web Client User Accounts and Privileges" on page 109.
3. Perform client initialization. For details, see "Database Extraction for a Mobile Web Client" on page 112.

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 Win 95 or 98

1. Using Notepad, or another text editor, open the autoexec.bat file in the C:\ directory.
2. Add the following line and save the file.

   ```
   set SIEBEL_LOG_EVENTS=<event level>
   ```

3. Reboot and 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

Siebel 7.5 includes enhanced tracing capability for Remote 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.

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 TNXPROC.

This functionality is for the Remote server components on the HQ or regional nodes, and is not available to the mobile user. These components include:

- Database Extract
- Transaction Router
- Parallel Database Extract
- Transaction Processor
- Transaction Merger
- Synchronization Manager

Tracing and logging of locks for the components includes the following objects on the application server:

- Inbox
- Outbox
- Dobjinst.dbf
- Visdata.dbf
- DX files

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. Navigate to Server Administration > Components, and select the component you need to trace.

   Choices include the components listed above.

2. Click the Component Event Configuration tab, and select the Event Type you want to trace.

   Use the Event Description field to help determine which Event Type to choose.

3. In the Component Event Configuration list, 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 Server Administration Guide, MidMarket Edition*.

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 preceding steps, 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 109.

      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 109.

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 121. 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 not necessary to perform laptop backups, because the information contained on the local database 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 109.

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.

Upgrading Mobile Web Clients 6

This chapter describes the upgrade tasks for a mobile Web client. Table 19 lists the required tasks for administering Siebel Remote. Each of these tasks is covered separately in this chapter.

Table 19. Upgrade Tasks for Mobile Web Client

Task	Where Performed	When Performed
Repository Upgrade with Siebel Anywhere on page 164	Server and client machine	Change to the application or change in the schema.
Repository Upgrade Without Siebel Anywhere on page 166	Server and client machine	Change to the application or change in the schema.
Major Upgrades for Mobile Web Client on page 167	Server and client machine	Major upgrade is available.
Apply Patches to the Mobile Web Client on page 168	Server and client machine	Patches available.

Repository Upgrade with Siebel Anywhere

This section describes upgrading a repository by using Siebel Anywhere. This includes procedures to prepare for the upgrade, build the upgrade kits for both the SRF file and database schema and then distribute the kits and finish the upgrade. It is assumed the new SRF file and schema extension files are available to the administrator.

To prepare to upgrade a repository

1. Have mobile users synchronize with the Server.

   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 therefore may cause problems when they are applied to the server.

2. Wait until transactions are applied to the Server.

   NOTE: No more changes should be made to the mobile Web client 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 database.

3. Wait until transactions are applied to the Server Database.
4. Disconnect the connected users on the Headquarters Node and shut down the server tasks except Transaction Processor and Transaction Router.

   NOTE: Connected users who remain connected and make changes may cause problems because they will be using the old SRF file, which will be out of step with the new repository and schema.

5. Transfer the new repository to the Server Database.

   For more information, see *Siebel Tools Reference, MidMarket Edition*.

To build the upgrade kit for the SRF file

- See *Siebel Anywhere Administration Guide, MidMarket Edition* for the necessary procedures.

To build the upgrade kit for the database schema

- See *Siebel Anywhere Administration Guide, MidMarket Edition* for the necessary procedures.

To distribute the upgrade kits and finish the upgrade

- See *Siebel Anywhere Administration Guide, MidMarket Edition* for the necessary procedures.

Repository Upgrade Without Siebel Anywhere

This section describes upgrading a repository without the Siebel Anywhere option. It is assumed the new SRF file is available to the administrator.

To upgrade a repository without Siebel Anywhere

1. Have mobile users synchronize with their server database.

   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 on the server databases, from the users, are complete.

   NOTE: No more changes should be made to the server databases until after it has been reinitialized. Changes made after this point will be lost.

3. Wait until the transactions are applied to the server databases.
4. Disconnect mobile users on the node.
5. Apply the upgraded repository to the headquarters database and synchronize the repository with the physical schema.

   For more information, see *Siebel Tools Reference, MidMarket Edition*.

6. Use Generate New Database to generate a new database template on the Headquarters Server.
7. Extract each mobile Web client on the Server.
8. Distribute the new SRF file to mobile users.

   Typically, this will be through email or a shared network drive.

9. Allow mobile users on the Headquarters Server to reinitialize their local databases and to resume work using the new SRF file.

Major Upgrades for Mobile Web Client

For major upgrades for a mobile Web client, perform the following tasks:

- **Ask all users to synchronize.** See "Data Synchronization" on page 144 for details.
- **Upgrade the server (both Database and Applications).** See *Siebel Server Installation Guide, MidMarket Edition*, and *Upgrade Guide for Microsoft Windows, MidMarket Edition* for details.
- **Create a Siebel Anywhere kit to upgrade the mobile Web client.** See *Siebel Anywhere Administration Guide, MidMarket Edition* for details.
- **Perform gennewdb.** "Generating a New Database Template" on page 74 for details.
- **Perform db extract.** See "Creating Mobile Web Client User Accounts and Privileges" on page 109 for details.
- **Initialize the mobile Web client.** See "Initializing a Mobile Web Client Database" on page 121 for details.

Apply Patches to the Mobile Web Client

Use Siebel Anywhere to distribute Siebel patches to mobile Web clients. See *Siebel Anywhere Administration Guide, MidMarket Edition* for more information about this topic.

7 Troubleshooting Remote

This chapter includes reference information, FAQs, and scenarios for Remote 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; navigate to Application Administration > Employees > Employees.

   The employee must have at least one position and responsibility. The fields cannot be blank.

2. Set up users as mobile Web clients: navigate to Remote Administration > 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

3. Start a "Generate New Database" process. Wait for it to finish successfully.

4. 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.

5. If the transaction process 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 63.

6 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 63.

7 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 63.

8 Make sure that you set up the mobile Web client hardware and software.

9 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.

10 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.

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:

  - Navigate to Server Administration and shut down the Synchronization Manager and then restart it again. Refresh the applet to make sure that Synchronization Manager has a state of running.

    For more details about Synchronization Manager, see *Siebel Server Administration Guide, MidMarket Edition* and "Starting Siebel Remote Server Components."

- 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 151 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 151 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 sync 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 synchronization 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" on page 82.

- 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 Server Administration Guide, MidMarket Edition*.

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 sync manager port number

1. From the application-level menu, select View > Site Map > Server Administration > Components.
2. From the Server Components list, click Query and run a query for the Synchronization Manager component.
3. From the Server Components list, select the Synchronization Manager component for the appropriate Siebel Server.
4. Click the Components Parameters tab, and click Query to run a query for the Static Port Number parameter.

   The port number in use is in the column Current Value.

To change the current sync manager port number

1. Navigate to the Component Parameters tab as in the previous procedure.
2. In the Component Parameter list, run a query for the Static Port Number parameter.
3. Enter the new value in the Value on Restart field.
4. Restart the server.
5. 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.

Navigate to the Remote Administration > 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.

Navigate to the Application Administration > 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 TechNote, "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 TechSupport 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 screen (Siebel Administration > Servers > Server Tasks), check the Task State and Status.

   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 Siebel Remote Administration > Transaction 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.

a From the application-level menu, select View > Site Map > Siebel Remote Administration screen > Processor Status.

b In the Transaction Processors list, select the old transaction processor.

c Click the menu button, then click Edit.

d In the Transaction Processors form, enter an end date in the End Date field.

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

7 Also run the following from the <Siebel_server_root>\bin directory, and supply the two TXT files to Siebel Tech Support:

```
Siebenv

txnutl /u SADMIN /p SADMIN /C <ODBC Datasource> /D <tableowner> /
w TxnOut1.txt /K LOWSCANMARK /L Y /Z N > TxnOut2.txt
```

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.

Client-Side Merge Issues on the Server A

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. Navigate to the Application Administration screen and from the Show drop-down list, click System Preferences.
2. Select System Preference Name = CSM Logging.
3. Set the System Preference Value = TRUE.
4. Restart the Siebel Server.

   This allows system preferences to take effect.
5. Restart the Siebel Remote components.

   See "Starting and Stopping Siebel Remote Server Components" on page 128.

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, MidMarket Edition* 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, MidMarket Edition* 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.

Solution

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.

Docking Object Changes B

This appendix includes three sections.

- The first section, "Routing Definition Differences in Docking Objects" on page 192, includes a table that provides an overview of the differences between the docking objects in Siebel 7.5 and Siebel 7.0.3.
- The second section, "Visibility Level Changes for Docking Objects" on page 199, includes four tables to identify the docking objects that changed visibility levels to help optimize mobile database size. For example, some docking objects changed visibility from Enterprise to Regional.
- The third section, "New Regional Docking Objects" on page 202, includes one table to identify the new regional docking objects that further optimize mobile database size.

Routing Definition Differences in Docking Objects

Table 20 compares the routing definition differences between Siebel 7.0.2, MidMarket Edition and Siebel 7.5, MidMarket Edition docking objects, and lists new ones for Siebel 7.5. The table provides a comparison of pairs of docking objects between the two versions, or identifies new docking objects for Siebel 7.5, MidMarket Edition. To read Table 20 use the following:

- If the Version field of a row reads 7.0.3, compare it with the 7.5 row immediately below it.
- If the Version field of a row reads 7.5 and the Version field of the row immediately above it also reads 7.5, a docking object is new in Siebel 7.5.

Table 20. Docking Comparison - Docking Object Differences

Version	Docking Object	Visibility Level	Primary Table	DO Active?
7.5	AppLicense	Private	S_APP_LICENSE	Y
7.0.3	AssignGroup	Enterprise	S_ASGN_GRP	Y
7.5	AssignGroup	Limited	S_ASGN_GRP	Y
7.5	AssignRuleGroup	Limited	S_ASGN_RULE_GRP	Y
7.0.3	BusComp Audit	Enterprise	S_AUDIT_BUSCOMP	Y
7.5	BusComp Audit	Limited	S_AUDIT_BUSCOMP	Y
7.5	CFGCache	Private	S_CFG_CACHEREQ	Y
7.5	CFGConstrain	Enterprise	S_CFG_CNSTR_TBL	Y
7.5	CFGRuleNode	Private	S_CFG_RULENODE	Y
7.0.3	CFGUIGrp	Enterprise	S_CFG_UIGROUP	Y
7.5	CFGUIGrp	Limited	S_CFG_UIGROUP	Y
7.0.3	CHAMP	Enterprise	S_CHMPINTV	Y
7.5	CHAMP	Limited	S_CHMPINTV	Y
7.0.3	CTI Cfg	Enterprise	S_CTI_CFG	Y
7.5	CTI Cfg	Limited	S_CTI_CFG	Y

Table 20. Docking Comparison - Docking Object Differences

Version	Docking Object	Visibility Level	Primary Table	DO Active?
7.5	CalendarResource	Enterprise	S_CAL_RSRC	Y
7.0.3	Call Script Page	Enterprise	S_CS_PAGE	Y
7.5	Call Script Page	Limited	S_CS_PAGE	Y
7.0.3	Call Script Path	Enterprise	S_CS_PATH	Y
7.5	Call Script Path	Limited	S_CS_PATH	Y
7.0.3	Call Script Question	Enterprise	S_CS_QUEST	Y
7.5	Call Script Question	Limited	S_CS_QUEST	Y
7.0.3	Call Script Style	Enterprise	S_CS_STYLE	Y
7.5	Call Script Style	Limited	S_CS_STYLE	Y
7.0.3	Catalog	Enterprise	S_CTLG	Y
7.5	Catalog	Limited	S_CTLG	Y
7.0.3	Catalog Category	Enterprise	S_CTLG_CAT	Y
7.5	Catalog Category	Limited	S_CTLG_CAT	Y
7.0.3	CommTemplate	Enterprise	S_COMM_TMPL	Y
7.5	CommTemplate	Private	S_COMM_TMPL	Y
7.5	CompPlan Budget	Private	S_CP_BDGT	Y
7.5	CompPlan Guideline	Private	S_CP_GDLN_SET	Y
7.5	CompPlan Job	Private	S_JOB	Y
7.5	CompPlan Job Family	Private	S_JOB_FAMILY	Y
7.5	CompPlan Plan	Private	S_CP_PLAN	Y
7.5	CompPlan Region	Private	S_CP_REGN	Y
7.5	CompPlan Salary Grade	Private	S_SALARY_GRADE	Y
7.5	CompPlan Salary Plan	Private	S_SALARY_PLAN	Y
7.5	CompensationCap	Enterprise	S_IC_CAP	Y

Table 20. Docking Comparison - Docking Object Differences

Version	Docking Object	Visibility Level	Primary Table	DO Active?
7.5	CompensationCapTxn	Private	S_IC_CAP_RUNTXN	Y
7.5	CompensationDrawTxn	Private	S_IC_DRAW_RNTXN	Y
7.5	CompensationPayTxn	Private	S_IC_PAY_RUNTXN	Y
7.5	CompensationPaymentRelease	Limited	S_IC_PAY_RLSE	Y
7.5	CompensationPlanCap	Limited	S_IC_PLAN_CAP	Y
7.5	CompensationSkew	Enterprise	S_IC_SKEW	Y
7.5	Competency	Private	S_CMPTNCY	Y
7.5	Content Set	Limited	S_CB_CNTNT_SET	Y
7.5	Content Type	Limited	S_CB_CNTNT_TYPE	Y
7.5	Curriculum Course Person	Private	S_CURR_CRSE_PER	Y
7.5	Curriculum Step Person	Private	S_CURR_STEP_PER	Y
7.0.3	DD Attribute Family	Limited	S_DD_ATTRFAM	Y
7.5	DD Attribute Family	Private	S_DD_ATTRFAM	Y
7.0.3	DD Data Object	Limited	S_DD_DATA_OBJ	Y
7.5	DD Data Object	Private	S_DD_DATA_OBJ	Y
7.0.3	DD DbConnector	Limited	S_DD_DB_CNCTR	Y
7.5	DD DbConnector	Private	S_DD_DB_CNCTR	Y
7.0.3	DD Dimension	Limited	S_DD_DIM	Y
7.5	DD Dimension	Private	S_DD_DIM	Y
7.0.3	DD Extract	Limited	S_DD_EXTRACT	Y
7.5	DD Extract	Private	S_DD_EXTRACT	Y
7.5	Deal	Limited	S_DEAL	Y
7.5	DecisionBroker	Enterprise	S_BRKR_DECISION	Y
7.0.3	DockRoute	Enterprise	S_DOCK_ROUTE	Y

Table 20. Docking Comparison - Docking Object Differences

Version	Docking Object	Visibility Level	Primary Table	DO Active?
7.5	DockRoute	Limited	S_DOCK_ROUTE	Y
7.5	DockSession	Private	S_DOCK_SESSION	Y
7.0.3	Docking Object	Enterprise	S_DOCK_OBJECT	Y
7.5	Docking Object	Limited	S_DOCK_OBJECT	Y
7.5	Dynamic Hierarchy	Private	S_DYN_HRCHY	Y
7.0.3	EAILookupMap	Enterprise	S_EAI_LOOKUPMAP	Y
7.5	EAILookupMap	Limited	S_EAI_LOOKUPMAP	Y
7.5	ETL Country Region	Private	S_ETL_CTRYREGN	Y
7.5	ETL Curr Run	Private	S_ETL_CURR_RUN	Y
7.5	ETL Day	Private	S_ETL_DAY	Y
7.5	ETL Industry	Private	S_ETL_INDUS_VER	Y
7.5	ETL String	Private	S_ETL_STRINGS	Y
7.5	ETL Time Day	Private	S_ETL_TIME_DAY	Y
7.5	ETL VI Image	Private	S_ETL_VI_IMAGE	Y
7.5	ETL VR Image	Private	S_ETL_VR_IMAGE	Y
7.5	EmployeeAward	Private	S_PER_AWARD	Y
7.5	EmployeeCompetency	Private	S_EMP_CMPTNCY	Y
7.5	EmployeeEduQualification	Private	S_PER_EDU_QUAL	Y
7.0.3	EmployeePerformanceRating	Enterprise	S_PERF_RATING	Y
7.5	EmployeePerformanceRating	Enterprise	S_PERF_RTNG_SCL	Y
7.0.3	EmployeePerformanceReview	Enterprise	S_PERF_RVW	Y
7.5	EmployeePerformanceReview	Enterprise	S_PERF_RVW_TMPL	Y
7.5	EmployeeProfCertification	Private	S_PER_PROF_CERT	Y
7.5	EmployeeProfMembership	Private	S_PER_PROF_MBR	Y

Table 20. Docking Comparison - Docking Object Differences

Version	Docking Object	Visibility Level	Primary Table	DO Active?
7.5	EmployeeReviewComponent	Private	S_RVW_COMP	Y
7.5	EmployeeWorkExperience	Private	S_PER_WRK_EXP	Y
7.5	Event Location	Limited	S_EVTLOC	Y
7.5	GroupNews	Limited	S_CONTENT_PATH	Y
7.0.3	IntegrationObjectMap	Enterprise	S_INT_OBJMAP	Y
7.5	IntegrationObjectMap	Limited	S_INT_OBJMAP	Y
7.0.3	InternalOrg	Enterprise	S_ORG_INT	N
7.5	Job Profile	Private	S_JOB_PROFILE	Y
7.5	LicenseUsage	Private	S_LICENSE_USAGE	Y
7.0.3	ListofValue	Enterprise	S_LST_OF_VAL	Y
7.5	ListofValue	Limited	S_LST_OF_VAL	Y
7.5	Localization	Limited	S_LOCALE	Y
7.0.3	MasterForecast	Enterprise	S_MSTR_FCST	Y
7.5	MasterForecast	Private	S_MSTR_FCST	Y
7.5	NamedUser	Private	S_NAMED_USER	Y
7.5	News Package	Enterprise	S_NEWS_PKG	Y
7.5	Nquire Account	Private	S_NQ_ACCT	Y
7.5	Nquire Job	Private	S_NQ_JOB	Y
7.5	OnLink Category	Private	S_ONL_CATEGORY	Y
7.0.3	OpptyForecast	Limited	S_OPTY_FCST	Y
7.5	OpptyForecast	Private	S_OPTY_FCST	Y
7.0.3	OpptyProdForecast	Limited	S_OPTYPRD_FCST	Y
7.5	OpptyProdForecast	Private	S_OPTYPRD_FCST	Y
7.0.3	OrgForecast	Limited	S_ORG_FCST	Y

Table 20. Docking Comparison - Docking Object Differences

Version	Docking Object	Visibility Level	Primary Table	DO Active?
7.5	OrgForecast	Private	S_ORG_FCST	Y
7.5	Partner Web Service Port	Private	S_PRTNR_WS_PORT	Y
7.0.3	PersonForecast	Limited	S_PER_FCST	Y
7.5	PersonForecast	Private	S_PER_FCST	Y
7.0.3	Position	Enterprise	S_POSTN	N
7.0.3	Product Business Unit	Limited	S_PROD_INT_BU	Y
7.5	ProductBusinessUnit	Limited	S_PROD_INT_BU	Y
7.5	ProductDefectProductLine	Limited	S_PRDFCT_PRODLN	Y
7.0.3	ProductForecast	Limited	S_PROD_FCST	Y
7.5	ProductForecast	Private	S_PROD_FCST	Y
7.5	Project Attachment Log	Private	S_PROJ_ATT_LOG	Y
7.5	ProxyAccessUser	Private	S_USER_PROXY	Y
7.0.3	Responsibility	Enterprise	S_RESP	Y
7.5	Responsibility	Limited	S_RESP	Y
7.5	Revenue Template	Enterprise	S_REVN_TMPL	Y
7.5	SA Application	Limited	S_SA_APPL	Y
7.5	SA ContentType	Limited	S_SA_CNTNT_TYPE	Y
7.5	SA NameValueDef	Limited	S_SA_NMVAL_DEF	Y
7.5	SalesHierarchy	Limited	S_SLS_HIER	Y
7.5	SalesHierarchyVersionRule	Private	S_SLSHR_VER_RL	Y
7.5	SalesTerritory	Limited	S_SLS_TERR	Y
7.5	SalesWarehouseHierarchy	Private	S_SLS_TERR_WHIR	Y
7.5	SystemKeyMapping	Private	S_SYS_KEYMAP	Y
7.5	Task Based Process	Limited	S_TASK_PROCESS	Y

Table 20. Docking Comparison - Docking Object Differences

Version	Docking Object	Visibility Level	Primary Table	DO Active?
7.5	TeamCompetency	Private	S_TEAM_CMPTNCY	Y
7.5	Test Run Question Category	Private	S_TSTRUN_CATSTS	Y
7.5	TestCore1	Private	T_MASTER	Y
7.5	TestCore2	Private	T_DETAIL	Y
7.0.3	Time Zone	Enterprise	S_TIMEZONE	Y
7.5	Time Zone	Limited	S_TIMEZONE	Y
7.5	Training Course	Private	S_PROD_INT	N
7.5	UniversalInboxItem	Limited	S_UNVBX_ITEM	Y
7.5	UniversalInboxItmTyp	Limited	S_UNVBX_ITMTYPE	Y
7.5	UsageSummary	Private	S_USAGE_SUMMARY	Y
7.5	User Role Task	Limited	S_USER_TASK	Y
7.5	Web Application	Private	S_WEB_APPL	Y
7.5	Web Service	Enterprise	S_WS_WEBSERVICE	Y
7.5	Web Service Port Type	Enterprise	S_WS_PORT_TYPE	Y
7.0.3	Workflow	Enterprise	S_WF_STEP	Y
7.5	Workflow	Limited	S_WF_STEP	Y
7.0.3	XAClass	Enterprise	S_XA_CLASS	Y
7.5	XAClass	Limited	S_XA_CLASS	Y
7.5	ePortalItem	Private	S_PRTL_ITEM	Y
7.5	ePortalPage	Private	S_PRTL_PAGE	Y

Visibility Level Changes for Docking Objects

The tables in this section identify the docking objects that changed visibility levels from Siebel 7.0.3 to Siebel 7.5, such as Enterprise to Regional. These changes help optimize mobile database size.

To read Table 21 through Table 25 on page 201, compare a pair of rows—the 7.03 Version with the 7.5 Version immediately below it. The docking objects listed in Table 21 through Table 25 are double entries in the Dock Object field, which allows comparison between the two versions.

Table 21. Docking Objects Changed from Enterprise to Regional

Version	Docking Object	Visibility Level
7.0.3	AssignGroup	Enterprise
7.5	AssignGroup	Regional
7.0.3	BusComp Audit	Enterprise
7.5	BusComp Audit	Regional
7.0.3	CHAMP	Enterprise
7.5	CHAMP	Regional
7.0.3	CTI Cfg	Enterprise
7.5	CTI Cfg	Regional
7.0.3	DockRoute	Enterprise
7.5	DockRoute	Regional
7.0.3	Docking Object	Enterprise
7.5	Docking Object	Regional
7.0.3	EAILookupMap	Enterprise
7.5	EAILookupMap	Regional
7.0.3	IntegrationObjectMap	Enterprise
7.5	IntegrationObjectMap	Regional

These docking objects pick up a routing value from the user interface. Their visibility levels can be set to enterprise, regional, or private depending upon the value entered through the UI.

Table 22. Docking Objects Changed from Enterprise to UI Based

Version	Docking Object	Visibility Level
7.0.3	Call Script Question	Enterprise
7.5	Call Script Question	User-Interface Based
7.0.3	ListofValue	Enterprise
7.5	ListofValue	User-Interface Based
7.0.3	Workflow	Enterprise
7.5	Workflow	User-Interface Based

Table 23. Docking Objects Changed from Enterprise to Private

Version	Docking Object	Visibility Level
7.0.3	CommTemplate	Enterprise
7.5	CommTemplate	Private

Table 24. Docking Objects Changed from Enterprise to Limited

Version	Docking Object	Visibility Level
7.0.3	Responsibility	Enterprise
7.5	Responsibility	Limited

Table 25. Docking Objects Changed from Limited to Regional

Version	Docking Object	Visibility Level
7.0.3	Comm	Limited
7.5	Comm	Regional
7.0.3	DD Filter	Limited
7.5	DD Filter	Regional
7.0.3	DD Hierarchical Attribute	Limited
7.5	DD Hierarchical Attribute	Regional
7.0.3	DD List Format	Limited
7.5	DD List Format	Regional
7.0.3	DD Measure	Limited
7.5	DD Measure	Regional
7.0.3	DD Measure-based Attribute	Limited
7.5	DD Measure-based Attribute	Regional
7.0.3	DD Sort Family	Limited
7.5	DD Sort Family	Regional
7.0.3	DD Source Code Format	Limited
7.5	DD Source Code Format	Regional
7.0.3	Note	Limited
7.5	Note	Regional
7.0.3	OrgSource	Limited
7.5	OrgSource	Regional
7.0.3	ProductDefect	Limited
7.5	ProductDefect	Regional
7.0.3	PrspCon	Limited
7.5	PrspCon	Regional

New Regional Docking Objects

New Regional docking objects listed below help prevent routing of regional data to mobile Web clients. This also helps to optimize the mobile database size.

Table 26. New Regional Docking Objects

Version	Docking Object	Visibility Level
7.5	AssignRuleGroup	Regional
7.5	Content Set	Regional
7.5	Content Type	Regional
7.5	GroupNews	Regional
7.5	ProductDefectProductLine	Regional
7.5	SA Application	Regional
7.5	SA ContentType	Regional
7.5	SA NameValueDef	Regional
7.5	Task Based Process	Regional
7.5	UniversalInboxItem	Regional
7.5	UniversalInboxItmTyp	Regional

Index

A

All List view, restriction 62
All Opportunity List view, restriction on 62
analyst routing model 104
attachment files, monitoring 136, 138
authentication
  mobile client passwords 109
  synchronization authentication, types of 21
Authentication parameter 70
autodial preferences 80
AutoStopDB parameter 20

C

cache size parameters 71
CDDir parameter 118
CD-ROM, extracting to 117
child records, repairing orphaned records 186
CleanFilesIter parameter 65
CleanTxnsIter parameter 65
Client parameter 118
Client Status view 132, 136
Client Wins rule 49
ClientDbType parameter 77
ClientRootDir parameter 82
ClientVersion parameter 119
commit interval 59
configuration file, setting preferences 81
conflict detection 48
conflict resolution
  about 19, 48
  deleting conflicts 51
  duplicate conflicts 50
  merging conflicts 51
  notification, system preference setting 60
  system preference settings 57
  system preferences 57
  updating conflicts 49
connected users
  downloading changes to local database, process 39
connection parameters, encryption 86
connection time evaluation 144
consultant routing model 102
CSM log
  activating 188
  deleting entries 189
CSM Logging 59
CSSSISDockFgetACKMsg 174

D

data
  conflicts between mobile users 48
  divergence 48
  storage, about 23
data integrity, corruption of 62
data routing models
  about 93
  analyst 104
  consistency with positions and responsibilities 94
  consultant 102
  field engineer 100
  field technician 95
  minimal data 105
  mobile client extract only 106
  mobile client standard 106

mobile partner user 106
sales manager standard 95
sales representative standard 94
senior field technician 96
database
See also local database
components, start-up parameters 76
database commit, interval 59
foreign key relationships 62
initialization 28
template generation at server setup 32
database extract
about 29
about running for mobile client 112
client database initialization of 120
concurrency rules 47
conflict resolution rules 49
extract parameters 118, 119
extracting to a CD-ROM directory 117
list of users, extracting 114
multiple extracts 112
reporting hierarchies, condition of 112
running 113
Target node error message 120
database files. *See* database templates
database server
about 13
access, about 14
hardware failure 158
media failure 159
RDBMS failure 158
types of data stored 23
database snapshot
Generate New Database component, running 75
Database Template Utility 28, 74
database templates
about 74
creating 74
distributing 77
regenerating, timing of 28
database volatility 145
DatFileSize parameter 118
DBA password, changing 73
DbaPwd parameter 76
DbfFile parameter 76
DbTmplFile parameter 118
DBX: Vis Rules Per Statement 1 60
DBX: Vis Rules Per Statement N 61
DCK-00123 error 172
DCK-00164 error 172
DCK-00213 error 173
DCK-00214 error 173
directories, application server 27
disk space requirements 144
distmpl, invoking 77
dock objects
defined 30
DockConnString machine 20
DockConnString parameter 82
docking
changing routing rules 30
database template, creating new 74
directory, about creating 17
hardware architecture 13
Docking: Get All PostnCon Rows 59
Docking: Transaction Logging 58
DockRepositoryName parameter 83
DockTxnsPerCommit parameter 83
Download Statistics view 138
duplicate conflicts 50
DX files
editing restriction (caution) 178
storage of 17
transaction exchange 26

E

ErrorMode parameter 69
ExtractRepos parameter 118
ExtractSince parameter 118

F

field engineer routing model 100
field technician routing model 95
file server, media failure 158
foreign key relationships, problems with 62

G

Generate New Database
about 28
database template, creating 74
Generate New Database component, running 75

H

hardware configuration, docking 13

I

IdDbRecreate parameter 68
IFaceTbls parameter 76
initialization
local database 28
problems, troubleshooting 174
troubleshooting problems 174
InitMethod parameter 118
ISpace parameter 76

L

Language parameter 118
Local Area Network (LAN)
connections 14
synchronization 13
local database
about 25
creating, view 33
database schema template, about 28
failure, recovering from 161
initialization diagram 35
initializing 33, 35
mobile user change process 41
performance considerations 25
preventing loss of transactions 113
process flow, downloading changes 39
refreshing 147
sizing considerations 29
Local Database Initialization program 28
local file system, about 25
LOGMGR: Vis Rules Per Statement 60

M

MaxCtxCache parameter 70, 71
MaxRead parameter 65, 67
MaxTasks parameter 70
MaxWrite parameter 67
media failures
database server 159
file server 158
Siebel Database Server 159
Siebel File Server 158
Siebel Remote Server 157
message-of-the-day 146
MinCtxCache parameter 70, 71
minimal data routing model 105
mobile clients
See also mobile users
authentication 45
autodial preferences, setting 80
client status, about monitoring 132
database, initializing 122
deactivating 148, 149
deleting 149
initialization failure, recovery 162
network connectivity 80
NT user, creating 110
ODBC code page settings 80
password, creating 110
productivity evaluation 145
reactivating 148
registering 88, 90
remote status, viewing 125
server directories, creating 27
synchronization requests, handling 29
synchronization start-up process 123, 124
task overview 79, 111
transaction log truncation, consequences of 24
mobile clients, database
authentication with database 45
initializing 121
local database, use of 25

mobile clients, database extract
about running 112
client database initialization 120
extract parameters 118, 119
extracting to a CD-ROM directory 117
list of users, extracting 114
multiple extracts 112
reporting hierarchies, condition of 112
running 113
mobile clients, routing models
about 93
analyst 104
consultant 102
field engineer 100
field technician 95
minimal data 105
mobile client extract only 106
mobile client standard 106
mobile partner user 106
sales manager standard 95
sales representative standard 94
senior field technician 96
mobile clients, synchronization
procedure 44
process, overview 36, 38
mobile clients, upgrading
patches, about 168
repository upgrade, using Siebel Anywhere 164
repository upgrade, without Siebel Anywhere 166
tasks, listed 167
mobile partner user routing model 106
mobile users
See also mobile clients
about 11
All List view, restriction 62
All Opportunity List view, restriction 62
change process 41
conflicting data resolution 48, 51
directories 27
local database, creating 33
message-of-the-day 146
registration prerequisite 32
routing model, changing 150
sending messages to 146
Siebel Remote client, starting 26
starting synchronization session 37, 38
synchronization frequency 144
unable to view records, troubleshooting 180
modem connections
about 14
servers 145
Move parameter 119
MRG: Docking Timestamp Source 57
MRG: System Conflict Resolution 57
MRG: Txns per Commit 59
MRG: User Friendly Notification 60, 125

N

NTDomain parameter 70

O

ODBC
code page settings 80
orphaned child records, about repairing 186

Q

quiet mode 20

R

RDBMS
failure 158
ReadClientsIter parameter 68
remote implementation
changing routing rules 30
database template, creating 74
remote server. *See* Siebel Remote Server
remote status, viewing 125
responsibilities, routing model consistency 94
routing models
about 93
analyst 104

changing for mobile user 150
consistency with positions and responsibilities 94
consultant 102
field engineer 100
field technician 95
minimal data 105
mobile client extract only 106
mobile client standard 106
mobile partner user 106
sales manager standard 95
sales representative standard 94
senior field technician 96
routing rules
about 30
number checked, system preference setting 60, 61
preconfigured routing rules 30

S

S_DOCK_TXN_LOG
flow diagram 17
troubleshooting 182
truncation, consequences of 24
sales manager routing model 95
sales representative routing model 94
Selective retrieval 13
senior field technician routing model 96
Server Manager, monitoring Siebel Remote server tasks 143
server modem connections 145
Server Wins rule 49
Session Manager 36
SetAppName parameter 67
Siebel Anywhere, repository upgrade 164
Siebel Client database
recovering from failure 161
refreshing 147
Siebel Database Server
downloading changes to local database, process 39
media failure 159
mobile user change process 41
Siebel database, about 13
Siebel File Server, troubleshooting media failure 158
Siebel File System
about 14
contents of 24
local file system, about 25
Siebel Mobile Web Client. See *mobile clients entries* 15
Siebel Remote
about 11
architecture diagram 13
conflicting data, resolution 48
data flow diagram 21, 22
hardware components 13, 15
process overview 16, 20
server tasks, monitoring 143
system preferences, setting 54
transmission failure 157
Windows NT user rights, enabling 72
Siebel Remote client
about 14, 26
communication protocol 26
setup overview 170
standalone synchronization 26
Siebel Remote Server
about 14
between synchronization sessions 38
components, starting/stopping 128
contents of 18
database template generation 32
databases, distributing 77
directories for mobile users 27
media failure 157
routing and merging diagram 36
server failure 157
setup overview 170
start-up parameters 63
Siebel Remote Synchronization Manager, security and authentication 45
Siebel Repository upgrades
using Siebel Anywhere 164
without Siebel Anywhere 166

Siebel Server
authentication process 45
SleepTime parameter
Transaction Merger startup parameter 69
Transaction Processor startup parameter 65
Transaction Router startup parameter 67
SQL
number routing rules checked, system preference setting 61
routing rules system preference setting 60
SQL Anywhere, suppressing window at startup 20
Stand-Alone Synchronizer 123, 124
synchronization
about repairing orphaned child records 186
authentication parameter 21
authentication process 45
background synchronization 26
between sessions 38
configuration parameters 82
connection parameter encryption 86
defined 11
diagnosing and restoring database synchronization 160, 161
encoding for transactions 23
frequency managing 144, 145
local transaction log, clearing 41
mobile clients, procedure 44
overview 36, 38
process overview 18
recommended frequency 21
Stand-alone 26
starting session, mobile user 37, 38
starting Siebel Remote client 26
transactions forwarding from Siebel Remote 25
troubleshooting 174
Synchronization Manager
authentication and security 45
concurrency rules 47
database connections 71
start-up parameters 70, 71
tasks, starting 29
SyncSince parameter 118
system preferences
CSM Logging 59
DBX: Vis Rules Per Statement 1 60
DBX: Vis Rules Per Statement N 61
Docking: Get All PostnCon Rows 59
Docking: Transaction Logging 58
LOGMGR: Vis Rules Per Statement 60
MRG: Docking Timestamp Source 57
MRG: System Conflict Resolution 57
MRG: Txns per Commit 59
MRG: User Friendly Notification 60, 125
setting 54

T

TableOwner parameter 83
Target node error message 120
timestamp system preference settings 57
trace file 178
transaction logs
log table, deleting entries 182, 184
modifying 41
monitoring 140
Transaction Merger
about 19, 31
concurrency rules 47
configuring 64
enabling logging, system preference 59
optimal number, determining 131
process diagram 36
starting 64
start-up parameters 69
troubleshooting 178
Transaction Processor
about 24, 30
concurrency rules 47
routing transactions 24
setup rules 129
starting 63
start-up parameters 65, 66
transaction log table, deleting entries from 182, 184

Transaction Router
about 24, 30
concurrency rules 47
configuring 67
optimal number, determining 130
parallel processes, about running 130
performance of 31
preference setting for S_POSTN_CON table 59
process diagram 36
setup rules 130
starting 64
start-up parameters 67
transactions
about processing 11
between synchronization sessions 38
encoding method caution 23
log file (S_DOCK_TXN_LOG) 17
processing, flow description 17
transaction log, truncation warning 24
transaction logging system preference setting 58
viewing transaction information 136, 138
transmission failures 157
troubleshooting
client initialization failure, recovery 162
CSSSISDockFgetACKMsg 174
DCK-00123 error 172
DCK-00164 error 172
DCK-00213 error 173
DCK-00214 error 173
DX file editing restriction 178
initialization problems 174
mobile users unable to see records 180
RDBMS failure 158
Siebel Database Server media failure 159
Siebel File Server media failure 158
Siebel Remote Server failure 157
Siebel Remote Server media failure 157
Siebel Remote transmission failure 157
synchronization problems 174
trace file 178
transaction log table 182, 184
transaction merger failure 178
TruncateTSTable parameter 119
TSBlockSize parameter 66, 120
TSCacheSize parameter 66, 67, 120
TSpace parameter 76
TSTableNum parameter 120
txnroute.exe file, startup parameters 67

U

update conflicts 49
Upload Statistics view 136
UseDdlFile parameter 76
UseTxnLog parameter 76

V

visibility
dock object classification 30
mobile client standard routing model 106
mobile users, restriction 62
troubleshooting problems with 180

W

WAN connections, about 14
WarehouseTbls parameter 77
Windows NT user rights, enabling 72
WriteCompressed parameter 65, 67