



Siebel Store-and-Forward Messaging Guide for Mobile Web Client

Siebel Innovation Pack 2013
Version 8.1/8.2
September 2013

ORACLE®

Copyright © 2005, 2013 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Contents

Chapter 1: What's New in This Release

Chapter 2: Overview of Siebel Store-and-Forward Messaging

About Siebel Mobile Products	7
Overview of Siebel Store-and-Forward Messaging Functionality	10
Overview of Store-and-Forward Messaging for Mobile Web Client Architecture	11
About IBM WebSphere MQe	15
Overview of Siebel Store-and-Forward Messaging Administration	16
About Applications and Store-and-Forward Messaging for Mobile Web Client	17
About Data Flow in Store-and-Forward Messaging for Mobile Web Client	17

Chapter 3: Installation and Configuration of Siebel Store-and-Forward Messaging

Prerequisites for Siebel Store-and-Forward Messaging Installation	21
Process of Installing Siebel Store-and-Forward Messaging	21
Enabling Component Groups for Store-and-Forward Messaging	21
Adding Responsibilities for Store-and-Forward Messaging	23
About Configuring Store-and-Forward Messaging Integration Objects	23
Configuring the Data Message Timeout Parameter for Store-and-Forward Messaging	24
Configuring Maximum Attachment Size	25
Optional Step: Configuring Store-and-Forward Messaging Workflows	25
Activating Workflows for Store-and-Forward Messaging	26
Process of Deploying IBM MQe Queue Managers	26
Installing the MQe Servlet	27
Enabling the SMQReceiver	29
Optional Step: Creating Additional SMQ Subsystems	30
Deploying SMQReceivers on UNIX Systems	31
Setting Up Multiple Queue Managers	33
Process of Installing and Configuring Mobile Web Client Applications	33
Changing the Queue a Mobile Web Client Uses	33
Setting Up a Network Connection for Mobile Web Client	34

Registering a Mobile Web Client for Store-and-Forward Messaging	34
Troubleshooting Store-and-Forward Messaging Setup and Configuration	35
Siebel Store-and-Forward Messaging Server-Side Troubleshooting Checklist	35
Siebel Store-and-Forward Messaging Server-Side Common Error Messages	38
Siebel Store-and-Forward Messaging Client-Side Troubleshooting Checklist	39

Chapter 4: Administration of Siebel Store-and-Forward Messaging

Overview of Siebel Store-and-Forward Messaging Administration	43
Reviewing and Deleting Store-and-Forward Messaging History	43
About Inspecting and Clearing Store-and-Forward Messaging Queues	47
Inspecting and Clearing Store-and-Forward Messaging Queues	48

Chapter 5: Siebel Store-and-Forward Messaging Sample Workflows

About Store-and-Forward Messaging Workflows	51
Dispatch Board - Generic Workflow	52
SAF Activity Workflow	53
Transaction Delete Message Workflows	54
About the Transaction Delete Message Workflows	54
SMQ PreDelete FS Invoice Record	55
SMQ Delete FS Invoice Record	56
SMQ Outbound Delete in Action Business Object	56
Configuring Transaction Delete Workflows for Different Business Components and Business Objects	57

Index

1

What's New in This Release

What's New in Siebel Store-and-Forward Messaging Guide for Mobile Web Client, Version 8.1/8.2

No new features have been added to this guide for this release. This guide has been updated to reflect only product name changes.

What's New in Siebel Store-and-Forward Messaging Guide for Mobile Web Client, Version 8.1

No new features have been added to this guide for this release. This guide has been updated to reflect only product name changes.

This chapter provides an overview of Store-and-Forward Messaging functionality. The following topics are included in this chapter:

- ["About Siebel Mobile Products" on page 7](#)
- ["Overview of Siebel Store-and-Forward Messaging Functionality" on page 10](#)
- ["Overview of Store-and-Forward Messaging for Mobile Web Client Architecture" on page 11](#)
- ["About IBM WebSphere MQe" on page 15](#)
- ["Overview of Siebel Store-and-Forward Messaging Administration" on page 16](#)
- ["About Applications and Store-and-Forward Messaging for Mobile Web Client" on page 17](#)
- ["About Data Flow in Store-and-Forward Messaging for Mobile Web Client" on page 17](#)

About Siebel Mobile Products

Several mobile applications are included in Oracle's Siebel product line. Each serves a different purpose and runs on different devices. Often, these applications complement each other. The mobile applications are as follows:

Siebel Mobile Web Client and Siebel Remote. A full-function disconnected client and supporting server-side software, respectively. These applications provide intelligent transaction routing, data synchronization, and Store-and-Forward Messaging capabilities.

Siebel Handheld. A full-function disconnected client with data filtering, data synchronization, and Store-and-Forward Messaging capabilities.

Siebel Wireless. Siebel applications delivered to client browsers that connect to a Siebel Server through a wireless network.

Figure 1 illustrates the hardware platforms on which these products run.

NOTE: In Figure 1, the term *form factors* refers to the general capabilities, screen shapes, and sizes of the various devices shown in the illustration.

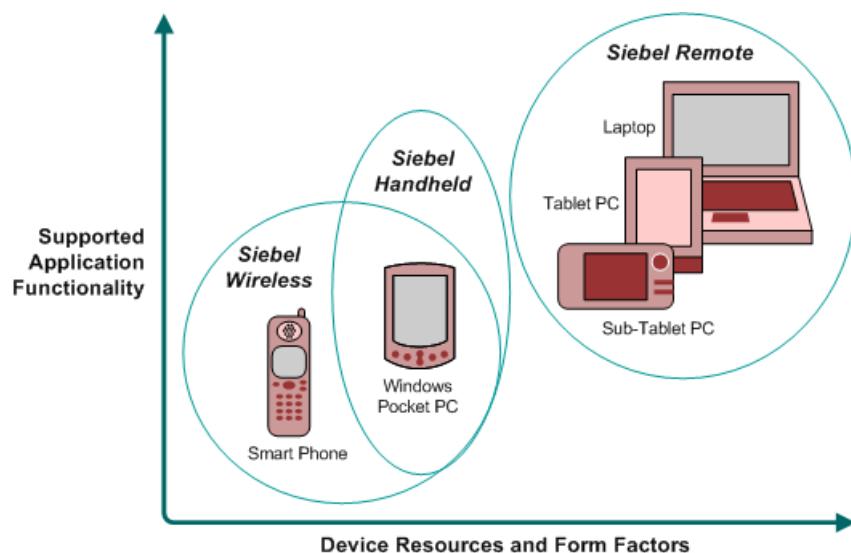


Figure 1. Siebel Mobile Product Platforms

Table 1 describes the Siebel mobile products' software components and supported hardware devices.

Table 1. Siebel Mobile Product Components and Supported Devices

Product Group	Server Components	Client Component	Supported Client Devices
Siebel Remote	Siebel Remote Store-and-Forward Messaging	Siebel Mobile Web Client	Desktop PCs Laptop PCs Tablet PCs
Siebel Handheld	Handheld Synchronization Object Manager Store-and-Forward Messaging	Siebel Handheld Client	Windows-based Mobile Devices
Siebel Wireless	Wireless Object Manager	Client Browser	Any mobile device running a supported client browser, including wireless devices and handheld devices

Table 2 describes the Siebel mobile products' supported Siebel applications and reference sources on your Siebel Bookshelf for information about these products.

Table 2. Siebel Mobile Product Supported Applications and References

Product Group	Supported Applications	Reference Sources
Siebel Remote	Almost all Siebel Business Applications and Siebel Industry Applications (Provides the same look and feel as those applications.)	<i>Siebel Remote and Replication Manager Administration Guide</i> <i>Siebel Store-and-Forward Messaging Guide for Mobile Web Client</i>
Siebel Handheld	Siebel Sales Siebel Field Service Siebel Consumer Goods Siebel eMedical Siebel ePharma	<i>Siebel Sales Handheld Guide</i> <i>Siebel Service Handheld Guide</i> <i>Siebel Consumer Goods Handheld Guide</i> <i>Siebel Medical Handheld Guide</i> <i>Siebel Pharma Handheld Guide</i> <i>Siebel Store-and-Forward Messaging Guide for Handheld</i>
Siebel Wireless	Siebel Sales Siebel Field Service Siebel Partner Relationship Management Siebel Self-Service Applications	<i>Siebel Wireless Administration Guide</i>

Figure 2 shows the synchronization infrastructure used by Siebel Handheld Clients and Siebel Mobile Web Clients.

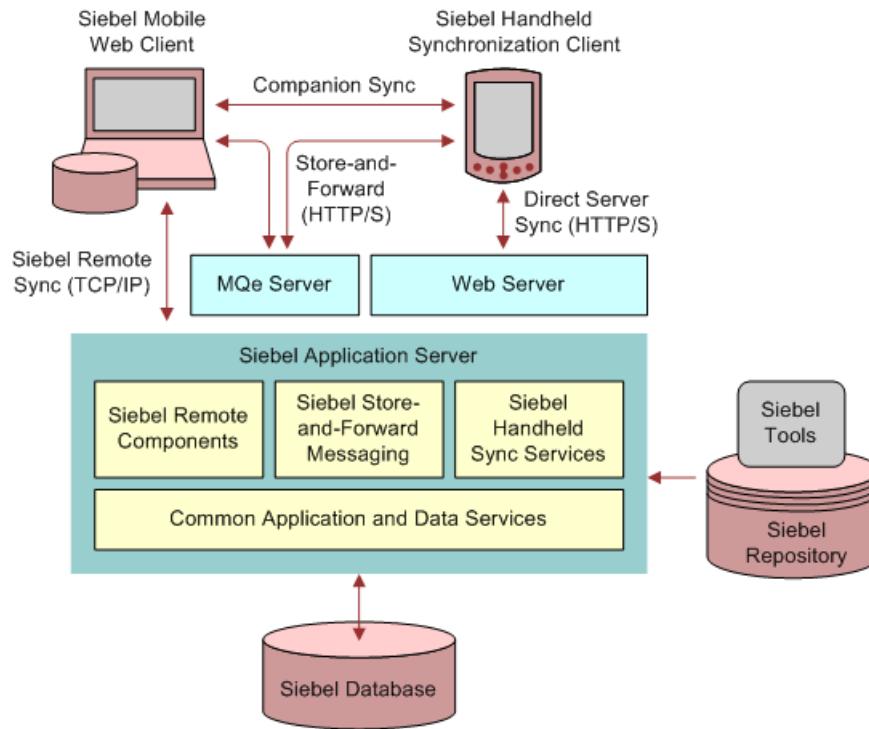


Figure 2. Synchronization Infrastructure for Siebel Handheld Client and Siebel Mobile Web Client.

Overview of Siebel Store-and-Forward Messaging Functionality

Siebel Store-and-Forward Messaging provides push-based information delivery between server-based Siebel Business Applications and Siebel Mobile Web Client applications. Using Siebel Store-and-Forward Messaging, a Siebel Business Application can dispatch information such as an activity and its associated data to a mobile user running the Mobile Web Client. The changes made by the mobile user in the Siebel Mobile Web Client are delivered to the Siebel Server through Store-and-Forward Messaging.

For instance, a mobile technician using the Siebel Service application on a Mobile Web Client together with Siebel Store-and-Forward Messaging can receive a dispatched activity, acknowledge the activity, update the activity's status, move parts, or debrief the assigned activity. During this process, Siebel Store-and-Forward Messaging enables the mobile user to communicate all information, in real or near-real time, with the back-end Siebel Server.

If the Mobile Web Client is connected to the Siebel Server, the information is delivered immediately. If the device is disconnected, the information is queued up in either the server side or the client side and is delivered as soon as the device is reconnected to the Siebel Server. Siebel Store-and-Forward Messaging uses the IBM WebSphere MQ Everyplace (MQe) queuing system to make sure data is delivered only once. The MQe queuing system is supported on wireless or wired networks that can provide TCP/IP connections.

The connection between the Mobile Web Client and the Siebel Server can be made in any of the following ways:

- By using conventional network wiring
- From a Mobile Web Client machine that supports wireless networking, by using a wireless connection in a Wi-Fi hot-spot location
- From a Mobile Web Client machine that is equipped with a cellular telephone modem, by using a wireless connection to the cellular phone network in a location where compatible cellular telephone coverage is provided

Siebel Store-and-Forward Messaging is available as an addition to Siebel Remote, the Siebel Mobile Web Client, and Siebel Business applications.

For more information about the Mobile Web Client, see *Siebel Remote and Replication Manager Administration Guide*.

Overview of Store-and-Forward Messaging for Mobile Web Client Architecture

The architecture of Store-and-Forward Messaging for Mobile Web Client consists of Siebel Server elements, Siebel Mobile Web Client elements, and IBM MQe queueing software, in the context of a Siebel Business Application deployment that includes Siebel Remote.

Table 3 specifies the kinds of data synchronization that are handled by Store-and-Forward Messaging, and the kinds of data synchronization that are handled by Siebel Remote.

Table 3. Synchronization Types Handled by Store-and-Forward Messaging and Siebel Remote

Direction of Synchronization	Handled by Store-and-Forward Messaging for Mobile Web Client	Handled by Siebel Remote
Mobile Web Client to Server	<ul style="list-style-type: none"> ■ All local database transactions. ■ File attachments smaller than the maximum attachment size set by the administrator. For more information about this parameter, see "Configuring Maximum Attachment Size" on page 25. 	<p>File attachments larger than the maximum attachment size set by the administrator. For more information about this parameter, see "Configuring Maximum Attachment Size" on page 25.</p>
Server to Mobile Web Client	<ul style="list-style-type: none"> ■ New Activities assigned by means of the Dispatch Board to a Mobile Web Client that has Store-and-Forward. ■ Any enterprise database transactions that update existing Activities that have already been sent to the Mobile Web Client using Store-and-Forward Messaging. 	<ul style="list-style-type: none"> ■ All enterprise database transactions that Siebel Remote normally routes to the Mobile Web Client. Any transactions that were previously sent by means of Store-and-Forward Messaging are provided for redundancy, but duplication of records is prevented automatically. ■ All file attachments, regardless of size.

Figure 3 shows the components that are used to transmit messages from the Siebel Server and receive those messages on the Mobile Web Client using Store-and-Forward Messaging.

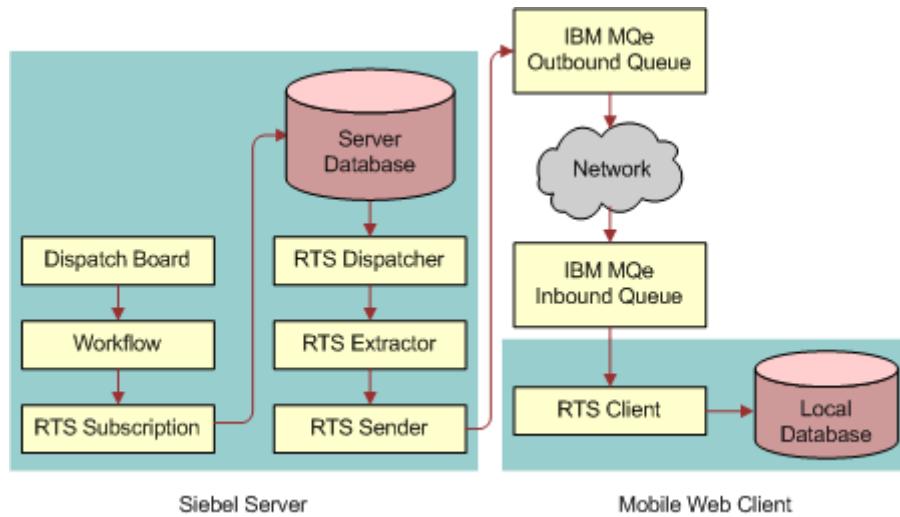


Figure 3. Components Used to Transmit Store-and-Forward Messages from the Siebel Server to Client

Figure 4 shows the components that are used to transmit messages from the Mobile Web Client and receive them on the Siebel Server using Store-and-Forward Messaging.

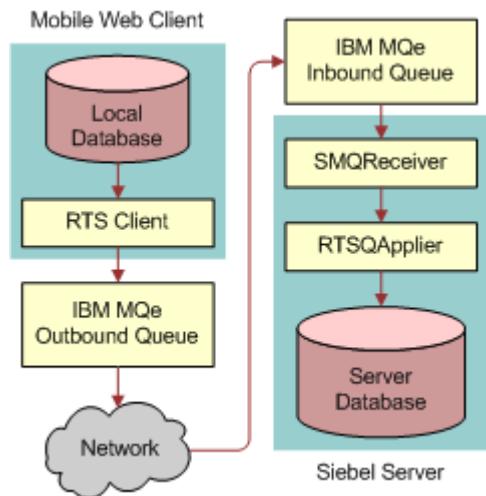


Figure 4. Components Used to Transmit Store-and-Forward Messages from Client to Server

Figure 5 shows the mixture of Siebel Remote components and Store-and-Forward components that are used to transmit all file attachments from the Siebel Server and receive them on the Mobile Web Client.

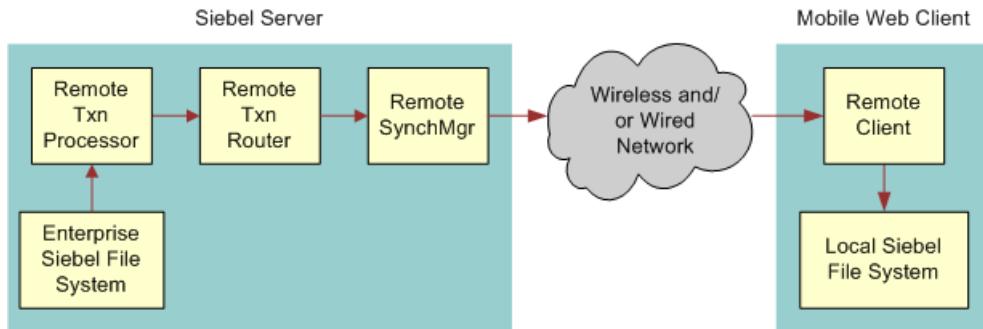


Figure 5. Siebel Remote and Store-and-Forward Components Sending All Attachments to Mobile Web Client

Figure 6 shows the mixture of Siebel Remote components and Store-and-Forward components that are used to transmit large file attachments from the Mobile Web Client and receive them on the Siebel Server.

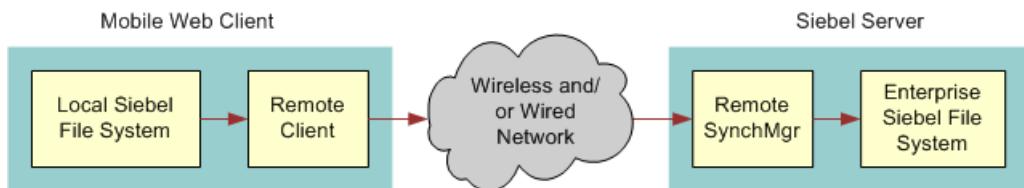


Figure 6. Siebel Remote and Store-and-Forward Components Sending Large Attachments to Server

Store-and-Forward architectural elements are described in the following paragraphs. For information about Siebel Remote elements, see *Siebel Remote and Replication Manager Administration Guide*.

RTSDispatcher

This server component retrieves transactions from the transaction log and creates Store-and-Forward work items that identify which Mobile Web Clients will receive Store-and-Forward messages.

RTSExtractor

This server component extracts the needed data from the database and creates the appropriate Store-and-Forward messages.

RTSSender

This server component sends Store-and-Forward Messages to the designated Mobile Web Clients by means of IBM MQe, and then deletes the associated work item.

IBM MQe

IBM WebSphere MQ Everyplace (MQe) provides a secure transport layer for transmitting Store-and-Forward messages over wired and wireless networks. For more information about MQe, see ["About IBM WebSphere MQe" on page 15](#).

RTSQApplier

This server component retrieves messages from the SMQ Receiver server component and applies the updates that are contained in the messages to the Siebel Server's database.

SMQ Receiver

This server component receives Store-and-Forward messages that Mobile Web Clients send by means of the transport service. SMQReceiver invokes the RTSQApplier component and passes the message to that component.

RTSClient

This architectural element is a background process on the Mobile Web Client. It performs the following functions:

- Sending database updates from the Mobile Web Client to the Siebel Server as Store-and-Forward messages
- Receiving Store-and-Forward messages from the Siebel Server and applying the database changes in the messages to the Mobile Web Client's local database.

About IBM WebSphere MQe

Siebel Store-and-Forward Messaging transmits messages to and from Siebel Mobile Web Clients, using IBM WebSphere MQ Everyplace (MQe) as a secure transport. A single instance of MQe is called a queue manager. An MQe queue manager contains multiple queues that hold messages being passed between the Siebel Server and the Mobile Web Clients. Both the Siebel Server and the Mobile Web Clients communicate with the queue manager to exchange messages.

NOTE: Store-and-Forward Messaging for Mobile Web Client is preconfigured to support delivery of Activities data from the Siebel Server to the Mobile Web Client. If you want to use Store-and-Forward Messaging to deliver other information types from the Siebel Server to the Mobile Web Client, contact Oracle's Application Expert Services for assistance.

Overview of Siebel Store-and-Forward Messaging Administration

You can administer Siebel Store-and-Forward Messaging from the Administration—Store-and-Forward screen and the Administration—Siebel Remote screen in a Siebel Web Client.

The Administration—Store-and-Forward screen has the following view related to Store-and-Forward Messaging:

- Store-and-Forward Message History

The Administration—Siebel Remote screen has the following views related to Store-and-Forward Messaging:

- Store-and-Forward Administration
- Mobile Clients

The following paragraphs describe these administration views. For detailed information about administration tasks, see [Chapter 3, “Installation and Configuration of Siebel Store-and-Forward Messaging”](#) and [Chapter 4, “Administration of Siebel Store-and-Forward Messaging.”](#)

Store-and-Forward Message History View

The Store-and-Forward Message History view in the Administration—Store-and-Forward screen provides administrative information for each outbound message. This view allows you to view such characteristics of outbound messages as the unique message ID, the application that was used for the message, the time the message was sent, the message size, the message status, and so on. You can use this view to track outbound messages for any user in your system, either to troubleshoot or to monitor application and system usage.

Store-and-Forward Administration View

The Store-and-Forward Administration view in the Administration - Siebel Remote screen displays the definition of the default integration object that is used in the synchronization of Activity information. The view can also be used for defining additional integration objects and for setting the timeout values that govern timely delivery of Store-and-Forward messages.

For more information about defining Integration Objects, see [“About Configuring Store-and-Forward Messaging Integration Objects” on page 23](#). For more information about setting timeout values, see [“Configuring the Data Message Timeout Parameter for Store-and-Forward Messaging” on page 24](#).

Mobile Clients View

The Mobile Clients view in the Administration - Siebel Remote screen is primarily intended for administration of Mobile Web Clients. However, it also contains the SAF Queue Subsystem field, which must be configured in order to enable Store-and-Forward Messaging for Mobile Web Clients.

About Applications and Store-and-Forward Messaging for Mobile Web Client

Siebel Store-and-Forward Messaging for Mobile Web Client can be used with any Siebel Business Application or any Siebel Industry Application that is compatible with the Mobile Web Client. Regardless of which Siebel applications your deployment uses, all Mobile Web Clients that use Store-and-Forward Messaging use the same preconfigured application settings, which you can view by navigating to Administration - Store-and-Forward > Store-and-Forward Applications.

NOTE: Store-and-Forward Messaging for Handheld has different requirements for configuring application settings from those described in the preceding paragraph. For more information, see *Siebel Store-and-Forward Messaging Guide for Handheld*.

About Data Flow in Store-and-Forward Messaging for Mobile Web Client

This topic describes how data flows for certain common operations that involve Store-and-Forward messaging, including the following:

- **Server to Client Data Flow.** Creation, assignment, and transmission of Activity information from the Siebel Server and Enterprise database to the Siebel Mobile Web Client.
- **Client to Server Data Flow.** Update to Activity information on the Mobile Web Client and transmission to the Siebel Server and the Enterprise database.
- **Change in ownership of an Activity record.**
- **Deletion of a child record of an Activity record.**

Store-and-Forward Messaging from Server to Mobile Web Client

The following steps describe what happens when new Activity information originates on the Server and Store-and-Forward transmits it to the Mobile Web Client.

- 1 A Siebel Web Client user logs in to the Siebel Service application and creates an Activity record.
- 2 The Dispatch Board lets an authorized user such as a Service Manager assign the Activity record to an appropriate Mobile Web Client user and to schedule the activity for an appropriate time frame. The Dispatch Board - Generic workflow and the SAF - Activity workflow are used in this step.

NOTE: If the Activity record includes an attachment, Store-and-Forward Messaging does not send the attachment to the Mobile Web Client. Instead, the attachment is queued for delivery during Siebel Remote synchronization.

- 3 The RTSDispatcher component retrieves information from the transaction log and creates Store-and-Forward work items that identify which Mobile Web Clients will receive Store-and-Forward messages.

- 4 The RTSExtractor component extracts the needed data from the database and creates the appropriate Store-and-Forward messages.
- 5 The RTSSender component sends Store-and-Forward Messages to the designated Mobile Web Clients by means of IBM MQe, and then deletes the associated work items.
- 6 IBM MQe transmits the message in HTTP format to the assigned Mobile Web Client user.
- 7 The Store-and-Forward message travels over local intranet and internet channels, as needed, including any of the following network configurations that are needed to deliver the message:
 - Wired networks
 - A Wi-Fi hotspot within range of the Mobile Web Client machine (the machine must have Wi-Fi capability)
 - Cellular phone networks within range of the Mobile Web Client machine (the machine must be equipped with a cellular phone modem and a cellular phone service plan)
- 8 On the Mobile Web Client, IBM MQe receives the Store-and-Forward Message and passes it to the RTSClient process, autosync.exe.

The RTSClient process is automatically started as part of the process of starting a Siebel application and connecting to the local database. The process continues running even if the user exits from the Siebel application.
- 9 The RTSClient process performs the following actions:
 - a It displays an incoming message alert icon (a red arrow pointing down at an envelope) in the notification area of the Mobile Web Client's screen.
 - b It applies the database change that is contained in the message to the local database.
 - c It displays an outgoing message alert icon (a red arrow pointing up from an envelope) in the notification area, indicating that it is sending the Siebel Server an acknowledgement of the message just received.
- 10 The Mobile Web Client user clicks the incoming message alert icon in the notification area of the screen to display the Incoming Message List, and the message data is available in the local database.

Store-and-Forward Messaging from Mobile Web Client to Server

The following steps describe how Store-and-Forward Messaging operates when a Mobile Web Client user changes any record in the local database.

- 1 The Mobile Web Client user creates or updates a record. For example, the user might add information about an activity that is in progress, or record the activity's successful completion, or create a new Contact record that is not yet related to any activity.

- 2 The RTSClient component constructs a message to send to the Siebel Server. The RTSClient component extracts the relevant data from the local database and formats it into a message that is compatible with IBM MQe. The RTSClient component displays an outgoing message alert icon (a red arrow pointing up from an envelope) in the notification area of the Mobile Web Client's screen.

NOTE: Handling of any file attachment depends on the size of the attachment. If the file attachment is smaller than the maximum attachment size set by the administrator, then the attachment is included in the message to be sent to the Siebel Server using Store-and-Forward Messaging. If the file attachment exceeds the maximum attachment size, then the RTSClient component queues the attachment for delivery using Siebel Remote synchronization.

- 3 IBM MQe transmits the message in HTTP format to the Siebel Server.
- 4 The Store-and-Forward message travels over network channels, as needed, including any of the following network configurations that are needed to deliver the message:
 - Wired networks
 - A Wi-Fi hotspot within range of the Mobile Web Client machine (the machine must have Wi-Fi capability)
 - Cellular phone networks within range of the Mobile Web Client machine (the machine must be equipped with a cellular phone modem and a cellular phone service plan)
- 5 On the application server, the SMQ Receiver component receives the update message from MQe and passes to it to the RTSQApplier component.
- 6 The RTSQApplier component applies the update to the Siebel Enterprise Database.

Change in Ownership of an Activity Record

If an Activity record in the Enterprise database is changed to add a Mobile Web Client as the Activity's owner, that change triggers the use of Store-and-Forward Messaging to deliver information about that activity to that Mobile Web Client.

If an activity record in the Enterprise database is changed to end a Mobile Web Client's ownership of that activity, that change triggers the use of Store-and-Forward Messaging to delete the record from that Mobile Web Client's local database. That Mobile Web Client receives no further updates about that Activity record.

Deletion of a Child Record of an Activity Record

If a child record of the Activity business component is deleted from the Enterprise database, Store-and-Forward Messaging does not transmit the delete operation to any Mobile Web Clients, unless your implementation includes a workflow that is specifically constructed to monitor the child business component and propagate such deletions to Mobile Web Clients. However, normal synchronization using Siebel Remote does propagate the deletion. For example, deletion of an invoice associated with an activity would reach the appropriate Mobile Web Clients only by means of Siebel Remote synchronization.

This chapter describes how to install and configure Store-and-Forward Messaging. The following topics are included in this chapter:

- “Prerequisites for Siebel Store-and-Forward Messaging Installation” on page 21
- “Process of Installing Siebel Store-and-Forward Messaging” on page 21
- “Process of Deploying IBM MQe Queue Managers” on page 26
- “Process of Installing and Configuring Mobile Web Client Applications” on page 33
- “Troubleshooting Store-and-Forward Messaging Setup and Configuration” on page 35

Prerequisites for Siebel Store-and-Forward Messaging Installation

Perform the basic Siebel Server installation according to the instructions contained in the *Siebel Installation Guide* for the operating system you are using.

Process of Installing Siebel Store-and-Forward Messaging

To install Siebel Store-and-Forward Messaging you must perform the following tasks:

- “Enabling Component Groups for Store-and-Forward Messaging” on page 21
- “Adding Responsibilities for Store-and-Forward Messaging” on page 23
- “About Configuring Store-and-Forward Messaging Integration Objects” on page 23
- “Configuring the Data Message Timeout Parameter for Store-and-Forward Messaging” on page 24
- “Configuring Maximum Attachment Size” on page 25
- “Optional Step: Configuring Store-and-Forward Messaging Workflows” on page 25
- “Activating Workflows for Store-and-Forward Messaging” on page 26

Enabling Component Groups for Store-and-Forward Messaging

The following procedure describes how to enable the component groups that are required for Store-and-Forward Messaging, using an administrative view in a Siebel Application. If you wish, you can enable the required component groups from the Server Manager, instead.

To enable component groups for Store-and-Forward Messaging

- 1 Log into a Siebel Web Client application as the System Administrator.
- 2 Navigate to Administration - Server Configuration > Enterprises > Component Groups.
- 3 On the Component Groups view tab, query for each of the component groups listed in the following table and click Enable.

Component Group	Comments
Enterprise Application Integration (EAI)	Used for Store-and-Forward Messaging server components such as SMQReceiver. Enable the entire component group.
Field Service Object Manager or Field Service	Used to enable Siebel Field Service related features and Store-and-Forward Messaging workflows. For deployment of Store-and Forward Messaging, only the Field Service Object Manager component is needed from the Field Service component group, but you can enable the entire group if you wish.
MWC Real Time Sync	Used for Store-and-Forward Messaging components such as RTSExtractor, RTSSender, RTSQApplier, and RTSDispatcher. Enable the entire component group.
Call Center Object Manager or Siebel Call Center	Used to enable Siebel Call Center-related features. For deployment of Store-and-Forward Messaging, only the Call Center Object Manager component is needed from the Siebel Call Center component group, but you can enable the entire group if you wish.
Siebel Remote	Used to handle all Store-and-Forward Messaging attachments that are sent from Siebel Server to Mobile Web Client, and attachments that are larger than a specified size and are sent from Mobile Web Client to Siebel Server. Also duplicates messages that are sent from Siebel Server to Mobile Web Client using Store-and-Forward Messaging, for redundancy. Enable the entire component group.
Sales Object Manager or Siebel Sales	Used to enable Siebel Sales-related features. For deployment of Store-and-Forward Messaging, only the Sales Object Manager component is needed from the Siebel Sales component group, but you can enable the entire group if you wish.

NOTE: If you are configuring a Siebel Industry application (such as Siebel Pharma Handheld) to use Store-and-Forward Messaging, you must enable the HandheldsyncSIS component group as well.

- 4 Click the Synchronize tab and click Synchronize.
- 5 Stop the Siebel Server, and then stop the Siebel Gateway Server.
- 6 Restart the Siebel Gateway Server, then restart the Siebel Server.

Adding Responsibilities for Store-and-Forward Messaging

The responsibilities listed in [Table 4](#) are related to Store-and-Forward Messaging system administrator or users.

Table 4. Responsibilities for Store-and-Forward Messaging

Responsibility	Comments
SMQ User	Allows access to the Store-and-Forward Messaging administration views. NOTE: It is important to verify that all Store-and-Forward Messaging administrators have the SMQ User responsibility.
Standard Client Responsibilities	Mobile Web Client users must have the responsibilities that give them access to the Siebel application views that they need to perform their job duties. No additional responsibilities are required for using Store-and-Forward Messaging as an end user.
Standard Siebel Administrator Responsibilities	Administrators for Store-and-Forward Messaging for Mobile Web Client must have the standard Siebel Administrator responsibility to gain access to the necessary administration views.

Associate these responsibilities with your end users, as needed, along with appropriate values for Organization and Service Region. For information about how to set up responsibilities, see *Siebel Security Guide*.

About Configuring Store-and-Forward Messaging Integration Objects

By default, Siebel Store-and-Forward for Mobile Web Client has a predefined integration object for the Activities business component. If you want your Mobile Web Clients to have Store-and-Forward functionality for records in another business component, contact Oracle's Application Expert Services for assistance in configuring additional integration objects.

NOTE: Oracle does not support the use of Store-and-Forward Messaging for Mobile Web Clients for scripted objects or objects that have attachments.

You must also contact Oracle's Application Expert Services for assistance if you want to use the Send business service (which is called RTS Subscription Service in Siebel Tools) in any custom portions of your implementation. This business service acts as a transport mechanism for delivering Store-and-Forward messages to a specific user ID.

For more information about integration objects, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.

Configuring the Data Message Timeout Parameter for Store-and-Forward Messaging

To make sure server outbound data messages are delivered in a timely manner, Siebel Store-and-Forward Messaging allows you to specify the following message timeout parameter:

- **Message Delivery Timeout.** This parameter determines the amount of time, in seconds, allowed for a given outbound data message to be delivered before it is timed out.

You configure the Message Delivery Timeout Parameter to indicate the timeout period of each message object. The default value is zero, meaning that the messages do not time out. You must change this value if you want to specify a timeout period for your dispatched messages.

The timeout length can be field-value dependent, for example, different activity messages can have different timeouts based on the value entered into the activity priority field. A message is removed from the user's server outbound queue if it is not delivered within the defined time period. Also, the related activity record's status is changed to Dispatch Failed when the system is unable to successfully transmit the message to a device.

The value set for this parameter is in seconds.

To set up the message delivery timeout parameter

- 1 Navigate to Administration - Siebel Remote > Store-and-Forward Administration.
- 2 In the Integration Objects list, select one of the following:
 - If you are using Store-and-Forward messaging to deliver Activity messages, select Real Time Sync Activity.
 - If you are using Store-and-Forward messaging to deliver other types of messages, select the applicable integration object for your data.
- 3 In the Timeout Admin list, click New to create a new record, and complete the following steps:
 - a In the Type field, enter Message Delivery Timeout.
 - b If you want the timeout period to vary depending on the value of a certain field (such as the Priority field), then enter the field name in the Field field; otherwise, leave the Field field empty.
 - c In the Default Time field, enter the number of seconds that must pass before an undelivered message times out.

The Default Time value is used only if you do not specify separate timeouts for specific values in a specific field.
- 4 If you specified a value for the Field field in [Step 3](#), then click New in the Field Values and Timeout list, and complete the following steps:
 - a Verify that the Field field displays the name of the field that you want Store-and-Forward messaging to use to determine timeout values, such as the Priority field.
 - b In the Values field, enter a typical value for the selected field.

For a Priority field, typical values might be 1-ASAP, 2-High, 3-Medium, and 4-Low.

- In the Time field, enter the number of seconds that must pass before an undelivered message with the specified Value times out.

For example, you might specify that activities with Priority set to 1-ASAP time out after 600 seconds (10 minutes).

Repeat this step as needed to specify an appropriate timeout period for each typical value.

Configuring Maximum Attachment Size

When Store-and-Forward Messaging sends data from the Mobile Web Client to the Siebel Server that includes an attachment, the channel over which the attachment is sent varies depending on the size of the file and the value of the system preference called SFM: Maximum File Attachment setting. The following procedure briefly describes how to configure this setting. For more information about system preferences related to Siebel Remote, see *Siebel Remote and Replication Manager Administration Guide*.

NOTE: All attachments that are sent from the Siebel Server to the Mobile Web Client travel by means of Siebel Remote synchronization, not by Store-and-Forwarding.

To configure the maximum attachment size for Store-and-Forward Messaging

- 1 Navigate to Administration - Siebel Remote > Remote System Preferences.
- 2 In the Remote System Preferences form, in the SF: Maximum File Attachment field, enter the size in kilobytes of the largest file attachment you want Store-and-Forward Messaging to transmit from the Mobile Web Client to the Siebel Server.

Optional Step: Configuring Store-and-Forward Messaging Workflows

Siebel Store-and-Forward Messaging provides the following preconfigured workflows, which are accessed through the Administration - Business Process > Workflow Deployment screen:

- Dispatch Board - Generic
- SAF Activity
- SMQ PreDelete FS Invoice Record
- SMQ Delete FS Invoice Record
- SMQ Outbound Delete in Action Business Object

You can configure these workflows to meet specific business requirements. For more information on Store-and-Forward Messaging preconfigured workflows, see [Chapter 5, "Siebel Store-and-Forward Messaging Sample Workflows."](#)

Activating Workflows for Store-and-Forward Messaging

After the workflows are properly configured, you need to activate them.

To activate workflows

- 1 Navigate to Administration - Business Process > Workflow Deployment. The Repository Workflow Processes screen appears.
- 2 In the Active Workflow Processes view, query for SAF Activity and Dispatch Board - Generic. If these workflows appear, delete them.
- 3 In the Repository Workflow Processes screen, query for SAF Activity and Dispatch Board - Generic.
- 4 Activate these workflows one by one by clicking the Activate button.

Process of Deploying IBM MQe Queue Managers

An IBM MQe Queue Manager runs as a separate process from the Siebel Server. This process can run either on the same computer that hosts the Siebel Server, or on a different computer in the same enterprise. Either way, the Siebel Server and the Siebel Mobile Web Client application communicate with the queue manager using the HTTP protocol, so the queue manager must be located on a machine that is accessible to both the Siebel Server and the Mobile Web Client. IBM MQe is a Java-based product, and requires certain Java configurations.

CAUTION: If Siebel Server and IBM MQe Queue Managers are running on different hosts, their system clocks *must* be synchronized. If they are not, PING messaging may fail. For more information about the MQe requirement, see the IBM documentation.

To deploy MQe Queue Managers, perform the following tasks:

- “Installing the MQe Servlet” on page 27
- “Enabling the SMQReceiver” on page 29
- “Optional Step: Creating Additional SMQ Subsystems” on page 30
- “Deploying SMQReceivers on UNIX Systems” on page 31
- “Setting Up Multiple Queue Managers” on page 33

NOTE: If your Siebel implementation uses Store-and-Forward Messaging for both Siebel Handheld and Siebel Mobile Web Client, the SMQReceiver for Siebel Handheld and the SMQReceiver for Siebel Mobile Web Client can run on the same server, but they use different subsystems. For more information about setting up SMQReceiver for Siebel Handheld, see *Siebel Store-and-Forward Messaging Guide for Handheld*.

Installing the MQe Servlet

You deploy MQe as a Java 2 Enterprise Edition (J2EE)-compliant Web application, or servlet. You install this Web application into a J2EE-compliant Web server or Web server plug-in (container or servlet container).

To perform this procedure, it is recommended that you first be familiar with installing Java servlets on the Web server.

To install the MQe Servlet

- 1** Make sure your system meets the following requirements:
 - The MQe Queue Manager servlet deployment requires a J2EE-compliant Web server or servlet engine that is capable of running Java servlets. For information about the required J2EE servlet specification version, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

NOTE: The Microsoft IIS Web server does not support servlets. However, third-party plug-ins are available that allow you to run servlets on IIS. Other Web servers such as Tomcat include servlet support by default.

 - Each computer running an MQe Queue Manager must have a Java Runtime Environment (JRE) installed. For information about the required Java Runtime Environment (JRE) version, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.
 - The servlet uses your Web server's existing port (generally port 80, the default for HTTP). Your firewall must permit inbound connections on this port for mobile clients or regular Internet users.
- 2** Locate the MQe servlet on your installation media.

The servlet is delivered as part of the Siebel Server installation package, in the <Siebel_Server>\classes\SiebelSMQServlet.war file. You can open .war files with winzip or any extraction tool that can decompress zip files.
- 3** Follow the procedure specific to your Web server or servlet engine for installing the MQe servlet.

This might mean copying the servlet directory tree into a directory that is specified by the Web server or servlet engine. Alternatively, it might mean running a configuration utility built into the server or engine. For a Tomcat server, you run the Tomcat Manager utility and deploy Context Path /smq using the complete path to your SiebelSMQServlet.war file as the WAR or Directory URL. For details of this procedure, see the documentation for your Web server or servlet engine.

The MQe queue manager is deployed in a directory that conforms to the J2EE Web application specification. This structure consists of specially-named subdirectories containing the necessary Java class files and configuration information. The layout is as follows:

```
<root>\  
  WEB-INF\  
    web.xml  
    lib\
```

MQeBase.jar
MQeSecurity.jar
SiebelSMQ.jar

The files in this directory structure are described in the following table.

File	Comments
web.xml	Deployment descriptor file that configures the servlet according to the J2EE servlet specification
MQeBase.jar	Redistributable binary file from MQe
MQeSecurity.jar	Redistributable binary file from MQe
SiebelSMQ.jar	Siebel servlet implementation

4 If the Web server you are preparing is a UNIX machine, perform the following steps; otherwise, skip to [Step 5](#):

- Make a backup copy of the .cshrc file for the administrator account that will run MQe.
- Open the .cshrc file and locate the following line, where `$HOME/${buid}` is the directory in which you installed your Siebel Server software:

```
if (-f $HOME/${buid}/ses/siebsrvr/siebenv.csh) source $HOME/${buid}/ses/siebsrvr/siebenv.csh
```

- Add the following environmental variables immediately following the line described in [Step b](#):
 - ❑ JAVA_HOME
 - ❑ PATH

Typical commands for setting these variables would be similar to the following:

```
setenv JAVA_HOME /user/jdk1.5.0_10
```

```
setenv PATH {$PATH}: /usr/jdk1.5.0_10: /usr/jdk1.5.0_10/bin: /usr/jdk1.5.0_10/jre: /usr/jdk1.5.0_10/jre/bin
```

- Make sure that the lines that define the following environmental variables include the location of the libjvm.so file and the parent folder:
 - ❑ LD_LIBRARY_PATH (for Enterprise Linux (EL), Linux, and Solaris)
 - ❑ SHLIB_PATH (for HP)

Typical commands for setting these variables would be similar to the following:

```
setenv LD_LIBRARY_PATH $ORACLE_HOME/lib:/usr/openwin/lib:/usr/dt/lib:/usr/jdk1.5.0_10/jre/lib/i386/server:/usr/jdk1.5.0_10/jre/lib/i386
```

- To initialize the modified variables, exit from the session and open a new session.
- Make a backup copy of the Web server's web.xml file, then open the web.xml file in a text editor and complete the following steps to change the DataDir parameter to specify the directory where you want the queues to be stored:

a Locate the following portion of the file:

```
<web-app>
  ...
    <servlet>
      ...
        <servlet-name>ServerQM</servlet-name>
        <init-param>
          <param-name>DataDir</param-name>
          ...
          <param-value>CHANGE_ME</param-value>
        </init-param>
      </servlet>
```

b Replace CHANGE_ME with the path to the directory where you want Store-and-Forward Messaging queues to be located.

NOTE: Queues will be located in the MQe\server-name subdirectory of the directory that you specify. For example, if you specify a DataDir value of C:\Siebel and leave the default servlet name of ServerQM unchanged, then queues will be located in C:\Siebel\MQe\ServerQM.

c Save and close the web.xml file.

6 Restart the servlet.

At runtime, after the first time the servlet runs, the system populates the directory specified by the web.xml file's DataDir parameter with the queues.

After installation, the servlet starts automatically the first time a client or a Siebel Server accesses it. The servlet shuts down when the Web server shuts down.

Enabling the SMQReceiver

SMQReceiver must be enabled after the IBM MQe Queue Manager is set up and running. You must enable the SMQReceiver after the Siebel Server starts running.

To enable the SMQReceiver

1 Open up the Windows command prompt and start the Siebel Server Manager for the Siebel Server using the following command:

```
srvmngr /g <gateway server> /e <enterprise> /s <Siebel server> /u <userid>
/p <password>
```

NOTE: *userid* and *password* are the system administrator's user name and password.

2 From the Siebel Server Manager, run the following command and verify that the parameter values are correct for your implementation:

```
list parameters for named subsystem SmqRTSReceiverSubsys
```

If you need to correct any parameter values for the SmqRTSReceiverSubsys subsystem, use a command like the following, substituting the appropriate values for your implementation:

```
change param SmqQueueName=InboundRTSQ, SmqServerMachineName=server_name,  
SmqServerPortNumber=8080, smqserverl etpathName=/smq/serverQM for named subsystem  
SmqRTSReceiversubsys
```

3 From the Siebel Server Manager run the following command once to start the SMQReceiver.

```
start task for comp SMQReceiver with  
ReceiverConnecti onSubsystem=SmqRTSReceiversubsys,  
ReceiverDataHandleSubSystem=SmqRTSDi spatchSubsys,  
ReceiverMethodName=Recei veDi spatch
```

Input parameters for the SMQReceiver are defined in the named subsystems. The first named subsystem defines the parameters necessary to configure an SMQReceiver for receiving messages (see the predefined named subsystem, SmqRTSReceiverSubsys). The second named subsystem identifies the workflow for processing the received messages (see the predefined named subsystem, SmqDispatchRTSDi spatchSubsys).

4 From the Siebel Server Manager, run the following commands:

```
Start task for comp RTSDi spatcher  
Start task for comp RTSExtractor  
Start task for comp RTSSender
```

Optional Step: Creating Additional SMQ Subsystems

You can use additional SMQ subsystems in Siebel implementations that have more users than a single IBM MQe queue manager can support. Each IBM MQe queue manager must communicate with an SMQ subsystem that is dedicated to that queue manager. Each Mobile Web Client is configured to communicate with a specific SMQ Subsystem (and its associated queue manager).

NOTE: Regardless of how many SMQ subsystems you create, you only need one Siebel Mobile Web Client record to be listed in Administration > Store-and-Forward > Message List > Store-and-Forward Applications.

To create additional SMQ subsystems

■ From the Siebel Server Manager, execute the following command, substituting the appropriate server name:

```
create named subsystem SmqRTSReceiversubsys1 for subsystem SMQRcvSubsys with  
SmqQueueType=MQe, SmqQueueName=InboundRTSQ, SmqQueueManagerName=ServerQM,  
SmqServerl etPathName=/smqServerQM, SmqServerMachineName=sdch70a012,  
SmqServerPortNumber=8080, SmqTransportServiceName="SMQ Transport Service",  
SmqTransportMethodName="WaitForMessage", SmqCompressi on="GZIP", SmqEncrypti on="RC4",  
SmqProcessComponentName="RTSQAppl ier", SmqProcessMethodName="RunRTSAppl ier",  
SmqProcessServiceName="RTS Message Apply Service"
```

NOTE: The value for SmqServerPortNumber must match the port for the Web server where you deploy the MQe Queue Manager servlet. The value of 8080 that is shown in this topic is the default port for a Tomcat Web server.

To start SMQReceiver for this new subsystem, use the following server manager command:

```
start task for comp SMQRecei ver1 wi th
Recei veConnecti onSubsystem=SmqRTSRecei verSubsys1,
Recei verDataHandl i ngSubsystem=SmqRTSDi spatchSubsys,
Recei verMethodName=Recei veDi spatch
```

Deploying SMQReceivers on UNIX Systems

This topic describes how to deploy SMQReceivers for various non-Windows platforms: Solaris, AIX, Red Hat Linux, and SUSE Linux. Use the procedure that applies to your implementation.

The following procedure describes how to deploy SMQReceivers on Solaris systems.

To install SMQReceivers on UNIX, Solaris platform

- 1 Make sure Java is installed on the following servers:
 - Tomcat/MQe server
 - SMQReceiver

For information about the required Java Runtime Environment (JRE) version, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

- 2 Set the following environment variables, substituting the correct paths for your environment:

```
setenv LD_LI BRARY_PATH  {$LD_LI BRARY_PATH}: /usr/j dk/j dk1. 5. 0_01/j re/l i b/sparc/
server

setenv JAVA_HOME /usr/j dk/j dk1. 5. 0_01
```

- 3 Make sure the directory that contains libjvm.so is in LD_LIBRARY_PATH.

For a UNIX machine, a typical path to the file would be /usr/jdk1.5.0_10/jre/lib/i386/server/libjvm.so.

NOTE: The libjvm.so file is required. If you do not find it, make sure that the servers are running the required version of the full Java Developer Kit (JDK).

The following procedure describes how to deploy SMQReceivers on AIX systems.

To install SMQReceivers on UNIX, AIX platform

- 1 Set the following environment variables:

```
setenv LIBPATH {$LIBPATH}: /usr/j ava1. 5/sdk/j re/bi n: /usr/j ava1. 5/sdk/j re/bi n/j 9vm/
setenv JAVA_HOME /usr/j ava1. 5/sdk
```

2 Make sure the directory that contains libjvm.so is in LIBPATH.

If necessary, add the directory to the LIBPATH environmental variable.

NOTE: The libjvm.so file is required. If you do not find it, make sure that the servers are running the required version of the full Java Developer Kit (JDK).

3 Create SmqJVMSubsys: Use Server Manager and issue the following commands:

```
create named subsystem SmqJVMSubsys for subsystem JVMSubsys
```

```
change param CLASSPATH="<Instal I _Di r>/si ebsrvr/cl asses/Si ebel JI _Common. jar: <Instal I _Di r>/si ebsrvr/cl asses/Si ebel JI . jar: <Instal I _Di r>/si ebsrvr/cl asses/Si ebel JI _psl . jar: <Instal I _Di r>/si ebsrvr/cl asses/Si ebel JI _enu. jar: <Instal I _Di r>/si ebsrvr/cl asses/Si ebel . jar" for named subsystem SmqJVMSubsys
```

NOTE: If the server you are working on does not accept commands this long, then place the command in a text file called cmd.txt, located in the server's bin directory. Then execute the following command:

```
smgr -i cmd.txt
```

The following procedure describes how to deploy SMQReceivers on Red Hat Linux systems or SUSE Linux systems.

To install SMQReceivers on UNIX, Red Hat Linux or SUSE Linux platform

1 The EAI component group is needed to run SMQReceiver.

2 Make sure the directory that contains libjvm.so and its parent directory are in LD_LIBRARY_PATH.

If necessary, add these directories to the LD_LIBRARY_PATH environmental variable.

NOTE: The libjvm.so file is required. If you do not find it, make sure that the servers are running the required version of the full Java Developer Kit (JDK).

3 Create SmqJVMSubsys: Use Server Manager and issue the following commands:

```
create named subsystem SmqJVMSubsys for subsystem JVMSubsys
```

```
change param CLASSPATH="{$SI EBEL_HOME}/si ebsrvr/cl asses/Si ebel JI _enu. jar: {$SI EBEL_HOME}/si ebsrvr/cl asses/Si ebel . jar" for named subsystem SmqJVMSubsys
```

NOTE: If the server you are working on does not accept commands this long, then place the command in a text file called cmd.txt, located in the server's bin directory. Then execute the following command:

```
smgr -i cmd.txt
```

Setting Up Multiple Queue Managers

A Store-and-Forward messaging application can send and receive messages by way of multiple IBM MQe Queue Managers. Each MQe Queue Manager is a server process. It has a unique name, can run on any host, listens on a specified port number, and has its own storage for queues. Each Queue Manager has one SMQReceiver component with its individual subsystem. If you use multiple Queue Managers, then you must create multiple subsystems.

Your Siebel application comes with one default subsystem called SmqRTSReceiverSubsys. Within the subsystem is a set of parameters that specify its characteristics. To support multiple Queue Managers, you create multiple subsystem parameter sets, one for each MQe Queue Manager. Each SMQ Receiver has a parameter that specifies the name of the subsystem to use. Each Mobile Web Client is configured to communicate with a specific subsystem.

For information about creating multiple subsystems, see ["Optional Step: Creating Additional SMQ Subsystems" on page 30](#). For information about specifying a subsystem for an SMQ Receiver, see ["Enabling the SMQReceiver" on page 29](#).

Process of Installing and Configuring Mobile Web Client Applications

After setting up the Siebel Server and the queue manager, you are now ready to install and configure the Siebel Mobile Web Client applications to enable Store-and-Forward Messaging.

To install and configure the Siebel Mobile Web Client applications, perform the following tasks:

- 1 Set up the Mobile Web Client machine to use Wi-Fi and/or cell phone networks, if you plan to use them for Store-and-Forward Messaging.
- 2 Set up the Mobile Web Client's network connection. See ["Setting Up a Network Connection for Mobile Web Client" on page 34](#).
- 3 Assign the Mobile Web Client to an SMQ Receiver subsystem. See ["Registering a Mobile Web Client for Store-and-Forward Messaging" on page 34](#).
- 4 Extract and initialize the local database for the Mobile Web Client according to the instructions in *Siebel Remote and Replication Manager Administration Guide*. This task must be performed after the SMQReceiver subsystem is assigned.
- 5 Verify that the Mobile Web Client's autosync.cfg file contains the correct host name for the queue manager.

Changing the Queue a Mobile Web Client Uses

From time to time, you may need to modify queue manager information for one or more users. For example, this can happen if an employee transfers to an office that is served by a different queue manager, or if your organization adopts a new network domain name. In these cases, complete the steps in the following procedure.

To change which queue a Mobile Web Client uses for Store-and-Forward Messaging

- 1 Make sure that there are no pending messages in the inbound or outbound queues on either the Mobile Web Client or the Siebel Server.
- 2 Log in to your Siebel application (connected to the server), and navigate to Administration - Siebel Remote > Mobile Clients.
- 3 In the Parent Server form, use standard query techniques to select the server that the Mobile Web Client communicates with.
- 4 In the Mobile Clients list, find the record for the Mobile Web Client that you are assigning to a new queue.
- 5 In the SAF Queue Subsystem field, enter the name of the new subsystem for the Mobile Web Client.
- 6 Re-extract and reinitialize the local database for the Mobile Web Client.

For more information, see *Siebel Remote and Replication Manager Administration Guide*.

- 7 Inspect the Mobile Web Client's autosync.cfg file to make sure that the initial extract process placed the correct information about the new queue in the file.

Setting Up a Network Connection for Mobile Web Client

You must set up a network connection for the Mobile Web Client machine so that it can communicate with the queue manager by way of an IP network through the HTTP protocol. The network connection can be either wired or wireless, as long as the Mobile Web Client has access to the appropriate queue manager. For information about setting up Mobile Web Clients, see *Siebel Remote and Replication Manager Administration Guide*.

Registering a Mobile Web Client for Store-and-Forward Messaging

Each Mobile Web Client must be registered for Siebel Remote communication with a particular server—either the HQ (headquarters) server, or a Regional Node. At the same time as you register a Mobile Web Client to communicate with a particular server, you can also specify the SMQReceiver subsystem that the Mobile Web Client will use for Store-and-Forward Messaging.

Briefly, this task is performed by creating a Mobile Web Client record in the Siebel Remote Administration screen's Mobile Web Client view. While specifying other values needed for Siebel Remote operations, you can enable Store-and-Forward Messaging for the Mobile Web Client by entering the SMQReceiver subsystem name (such as SmqRTSReceiveSubsystem) in the SAF Queue Subsystem field.

For detailed instructions on how to complete this task, see *Siebel Remote and Replication Manager Administration Guide*. For more information about queue subsystems, see “[Enabling the SMQReceiver](#)” on page 29, and “[Optional Step: Creating Additional SMQ Subsystems](#)” on page 30.

NOTE: This task must be performed before you extract the local database for the Mobile Web Client. Any time you change whether a Mobile Web Client uses Store-and-Forward Messaging, that client must be re-extracted in order to make the client and server use the same encryption key. Each subsystem and queue manager has its own encryption key, which the Mobile Web Client obtains during local database initialization. When a Mobile Web Client and a server have non-matching encryption keys, incoming messages from either machine will be incomprehensible to the other.

Troubleshooting Store-and-Forward Messaging Setup and Configuration

When a problem occurs for Store-and-Forward Messaging setup and configuration, the issue may reside on either the server side or the client side of the Siebel Mobile Web Client application. Troubleshooting information provided here includes the following topics:

- “[Siebel Store-and-Forward Messaging Server-Side Troubleshooting Checklist](#)” on page 35
- “[Siebel Store-and-Forward Messaging Server-Side Common Error Messages](#)” on page 38
- “[Siebel Store-and-Forward Messaging Client-Side Troubleshooting Checklist](#)” on page 39

NOTE: You can use an administrative utility to resolve failed transactions. For more information about this utility, see 954172.1 (Article ID) on My Oracle Support and the online help for the utility.

Siebel Store-and-Forward Messaging Server-Side Troubleshooting Checklist

Use the following checklist to diagnose problems that originate with Store-and-Forward Messaging servers.

- 1 Make sure the MQe Servlet is running properly in the J2EE server.
If a Tomcat server is running properly, it has a Status of Start. If you have another J2EE server, consult your server’s documentation.
- 2 If WfProcMgr, SMQReceiver, and the Tomcat server are deployed on multiple hosts, make sure their system clocks are synchronized. If they are not, the Expire Time property in IBM MQe will be calculated incorrectly.
- 3 Make sure the SMQReceiver is running. You can check the running tasks to verify whether the SMQReceiver is running from the Siebel Server Manager command line. To check the status of the SMQReceiver, use the following command from the Siebel server manager:

```
list task for comp SMQRecei ver
```

When the SMQReceiver is running, there should be a log file SMQReceiver(number).log in the Siebel Server log directory.

4 Make sure that RTSDi spatcher, RTSExtractor, and RTSSender are also running. To check the status of these components, use the following commands from the Siebel server manager:

```
list task for comp RTSDi spatcher
list task for comp RTSExtractor
list task for comp RTSSender
```

5 Check the logs to make sure that RTSQApplier is running intermittently.

Look for log files that have names similar to RTSQApplier_0041_42992997.log, but with different numerals.

Alternatively, you can check the RTSQApplier in your Siebel application, by performing the following steps:

- a** Navigate to Administration - Server Management > Enterprises.
- b** In the Enterprise Servers list, select your Enterprise Server.
- c** In the Servers list, select your Siebel Server.
- d** Scroll down and click the Tasks tab.
- e** In the Tasks list, use standard query techniques to locate the record for Real Time Sync Transaction Applier.

NOTE: This record is visible only if the RTSRemote component group is enabled.

- f** Inspect the Running Tasks field for the record. If RTSQApplier is currently running, the value of this field is greater than zero. When the task is not running, the value of this field is zero.

6 Make sure the queue directory tree on the queue manager host computer is created.

NOTE: In the example below, <USERNAME> stands for the user name of the Mobile Web Client user.

The default directory structure looks like the following, where DataDir is a parameter specified in web.xml:

```
<DataDir r>\MQe\ServerQM\Queues\ServerQM
  Admi nQ
  Admi nRepl yQ
  InboundRTSQ
  [InboundRTSQ. <USERNAME>]
  [<USERNAME>. SFQ] //
```

Check whether InboundRTSQ.<USERNAME> and <USERNAME>.SFQ have been created for the specific Mobile Web Clients that you are testing. If so, see if those directories contain messages.

7 Check messages sent from the Siebel Server from the Administration—Store-and-Forward screen. Navigate to Administration—Store-and-Forward > Store-and-Forward User Message History, then query for the user and verify whether any message has been sent to the Siebel Mobile Web Client application. See “[To review all message history records](#)” on page 44.

8 Check that the integration object is populated in the Administration - Siebel Remote > Store-and-Forward Administration view.

If the integration object is not populated, create a service request (SR) on My Oracle Support. Alternatively, you can phone Oracle Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers are listed on My Oracle Support.

9 Examine the log files such as SMQReceiver log files, RTSDispatcher, RTSExtractor, RTSSender, and RTSQApplier log files to find more information for troubleshooting.

You can enable or increase the server logging levels using the Siebel Server Manager commands. The commands are:

- For SMQReceiver,

```
change evtl ogl vl SMQMsgSvc=5 for comp SMQRecei ver
```

```
change evtl ogl vl JVMI ni t=5 for comp SMQRecei ver
```

```
change evtl ogl vl EAI Tranport=5 for comp SMQRecei ver
```

- For RTSDispatcher, RTSExtractor, RTSSender and RTSQApplier,

```
change evtl ogl vl RTSEventLog = 5 for comp RTSDi spatcher
```

```
change evtl ogl vl RTSEventLog = 5 for comp RTSExtractor
```

```
change evtl ogl vl RTSEventLog = 5 for comp RTSSender
```

```
change evtl ogl vl RTSEventLog = 5 for comp RTSQAppl i er
```

10 Test the outbound workflow. Create a new activity and use the Dispatch Board to assign it to a Mobile Web Client that is set up to receive Store-and-Forward Messaging. This event triggers the Dispatch Board - Generic workflow and the SAF Activity workflow. A data message should reach the user's Siebel Mobile Web Client application if the Mobile Web Client is connected to the queue manager. Server outbound messages are put in the [UserQueueName].SFQ directory temporarily until the Mobile Web Client polls for the messages. If there is a network connection issue, the messages sit in this directory until they expire. The expiry time is specified by the Message Delivery Timeout parameter in the Store-and-Forward Timeout Administration applet.

11 Sometimes, the workflow runtime events cannot be triggered after activation. Restarting the Siebel Server can fix this problem.

12 The Siebel Server components SRProc and SRBroker are essential for Siebel Store-and-Forward Messaging. They can fail for many different reasons. Check their corresponding log files, SRProc*.log and SRBroker*.log to see if there are any errors.

13 SMQ Receiver must be enabled after the queue manager is set up and running.

For more information about troubleshooting in a Siebel Remote environment, see *Siebel Remote and Replication Manager Administration Guide*.

Siebel Store-and-Forward Messaging Server-Side Common Error Messages

Some common server-side error messages, and their solutions, are described in [Table 5](#).

Table 5. Server-Side Common Error Messages

Symptom or Error Message	Diagnostic Steps or Cause	Solution
MQe Internal Error: RemoveMsg: Message inside of Inbound Queue is not deleted successfully. Error Code: 121	Error Code 121 means there is no matching message in the queue. Multiple Receivers or Tasks may be trying to process or dispatch from the same queue at the same time.	Check svrmgr to find out how many SMQReceivers are running in the server. Reconfigure as needed to make sure that there is only one SMQReceiver for each application.
ObjMgrBusServiceLog Error 1 000008e0440515c0:0 2006-03-02 00:05:23 (srmsyncsv.cpp (1214)) SBL-SRQ-00103: Unable to find definition for component WfProcMgr	This error may appear in the SMQReceiver log file. The server components may not be synchronized to the internal table.	To synchronize the server components to the internal table, perform the following steps: 1 In the Administration - Server Configuration > Synchronize screen, click Synchronize. 2 Restart the SMQReceiver component.
JVM Init Exception: Can't load JVM dll: libjvm	This error may appear in the SMQReceiver log file on Solaris and Linux systems. There may be an incorrect value in LD_LIBRARY_PATH.	Check the value of LD_LIBRARY_PATH by running one of the following commands: ■ env ■ echo \$LD_LIBRARY_PATH On AIX, check \$LIBPATH.

Table 5. Server-Side Common Error Messages

Symptom or Error Message	Diagnostic Steps or Cause	Solution
Java HotSpot(TM) Server VM (build 1.5.0_06-b05, mixed mode) IBM WebSphere MQ Everyplace 2.0.2.4 Created session for subsys SmqReceiverSubsys GetClientNames: com.ibm.mqe.MQException: Message not found[ServerQM 27] (code=121) atcom.ibm.mqe.MQeTransporter._remoteExecute(DashoA8173) at com.ibm.mqe.MQeTransporter.remoteExecute(DashoA8173) at com.ibm.mqe.MQeTransporter.getMessage(DashoA8173) at com.ibm.mqe.MQeRemoteQueue._synchronousGetMessage(DashoA8173) at com.ibm.mqe.MQeRemoteQueue.getMessage(DashoA8173) at com.ibm.mqe.MQeQueueControlBlock.getMessage(DashoA8173) at com.ibm.mqe.MQeQueueManager.waitForMessage(DashoA8173) at com.ibm.mqe.MQeQueueManager.waitForMessage(DashoA8173) at com.siebel.smq.session.MQeBaseSession.executeAdmin(MQeBaseSession.java:279) at com.siebel.smq.session.MQeServerSession.getClientNames(MQeServerSession.java:255)	This error may appear in the SMQReceiver Logoff. System clocks on different hosts may not be synchronized.	Synchronize the clocks on the host machines for WfProcMgr, SMQReceiver, and the Tomcat server.

Siebel Store-and-Forward Messaging Client-Side Troubleshooting Checklist

Use the following checklist to diagnose problems that originate with Store-and-Forward Messaging clients.

- 1 Verify that the following parameters are in the [SMQ] section of the autosync.cfg file for the Mobile Web Client. This file is located in the bin subdirectory of the directory where the Mobile Web Client software is installed. [Table 6](#) lists the default values.

Table 6. Parameter Values in the Mobile Web Client's autosync.cfg File

Parameter	Default Value
MQeServerAddr	[hostname] or IP address of the computer hosting the queue manager
MQeServerCmd	/smq/ServerQM
MQeServerPort	8082
MQeServerQMGr	ServerQM
MQeServerQueue	InboundRTSQ
QueuingSoftware	MQe. IBM MQe is the only queueing software supported in the current version of Store-and-Forward Messaging.
SMQCompression	The type of compression used for Store-and-Forward Messaging for Mobile Web Client. The only value currently supported is GZIP.
SMQConnectOnSend	Controls whether Message Transport software automatically attempts to connect to the server when a message is placed in the client's outbound queue. Default value is FALSE.
SMQConnectOnStart	Controls whether Message Transport software automatically attempts to connect to the server when a new client session is started. Default value is FALSE.
SMQEncryption,	The type of encryption used for Store-and-Forward Messaging for Mobile Web Client. The only value currently supported is RC4.
SMQLogMessages	When TRUE, copies of messages are logged as XML files. Default value is TRUE.
SMQPollingInterval	Specifies how often the Message Transport software polls for new messages. Default value is 30 seconds.

If any of the parameter values do not match the server configuration, then modify the autosync.cfg file and reinstall the Siebel Mobile Web Client with the correct parameters.

NOTE: The value of the MQeServerPort parameter must be set to match the port number of the Web server where you deploy the MQe Queue Manager servlet. For example, the default port number for a Tomcat Web server is 8080.

- 2 Examine the Siebel application log file on the Mobile Web Client to find more information for troubleshooting.

For information on working with the application log file, see *Siebel Remote and Replication Manager Administration Guide*.

You can also look for log files with names similar to rtsthrd_7824_5828.log, but with different numerals. Names in this format indicate a log file for the RTSClient.

This chapter describes administrative tasks for Store-and-Forward Messaging. The following topics are included in this chapter:

- ["Overview of Siebel Store-and-Forward Messaging Administration" on page 43](#)
- ["Reviewing and Deleting Store-and-Forward Messaging History" on page 43](#)
- ["About Inspecting and Clearing Store-and-Forward Messaging Queues" on page 47](#)
- ["Inspecting and Clearing Store-and-Forward Messaging Queues" on page 48](#)

Overview of Siebel Store-and-Forward Messaging Administration

Siebel Store-and-Forward Messaging administration screens and Siebel Remote administration screens allow you to administer mobile users, their Mobile Web Client machines, detailed Store-and-Forward Messaging message information, and timeout parameters.

This chapter presents an overview of tasks you perform when administering Siebel Store-and-Forward Messaging. Except where explicitly told otherwise, you administer Store-and-Forward Messaging for Mobile Web Client from the Administration - Siebel Remote > Store-and-Forward Administration screen.

NOTE: To access Store-and-Forward Administration screens, you must have the responsibility SMQ User. This responsibility is provided for the SADMIN account in seed data. If you are using a different account to administer Store-and-Forward Messaging, such as QADMIN, you may need to add this responsibility. For more information about associating responsibilities with user accounts, see ["Adding Responsibilities for Store-and-Forward Messaging" on page 23](#).

Reviewing and Deleting Store-and-Forward Messaging History

You can review message histories in the Store-and-Forward Message History view for all the outbound messages. (Store-and-Forward Messaging for Mobile Web Client provides message history only for outbound messages.) You may also review an individual user's outbound messages in the Personal Message History view. These views allow you to access characteristics of an outbound message, including the unique message ID, the application used for the message, the time when the message was received or sent, the message size, message status, and any error information. You can use this information to troubleshoot or monitor application or system usage.

You can also specify criteria for deleting old message history records for all messages or for an individual user's messages. You do this by specifying a time threshold beyond which the system deletes records for received messages only.

NOTE: The Store-and-Forward Message History view can display information for Siebel Handheld Clients, Siebel Mobile Web Clients, or both.

The tasks you can perform to review message history and delete message history records are listed below:

- "To review all message history records" on page 44
- "To delete all old message history records" on page 45
- "The following procedure describes how to review message history records for a specific user's outbound messages." on page 46
- "To delete a user's old message history records" on page 46

The following procedure describes how to review message history records for all outbound messages.

To review all message history records

- 1 Navigate to Administration - Store-and-Forward > Store-and-Forward Message History.
- 2 Review the information in the Store-and-Forward Message History view.

The data in the message history records is generated automatically, and the records are read only. Some fields are described in the following table.

Field	Description
Application	The name of the Siebel Mobile Web Client application.
Direction	The direction the message has traveled. Store-and-Forward Messaging for Mobile Web Client currently records only From Server messages.
Error	Message content if this is an ERR (Error) type message.
Message ID	The message ID in the screen is the row ID of the message history table.
Message Type	Available types change vary according to the value in the Direction field.
Original Message ID	ID of the corresponding request message if this message is a response message.
Size	The message size, as transported to the queuing system, after compression.

Field	Description
Status	<ul style="list-style-type: none"> ■ Pending. The outbound data message has been created but is delayed until the specified Start to Send time. For information about the Start to Send parameter, see "Configuring the Data Message Timeout Parameter for Store-and-Forward Messaging" on page 24. ■ Sent. The server outbound message has been sent to the queuing system. ■ Server Received. The server has received the message from the Siebel Mobile Web Client application. ■ Client Received. The Mobile Web Client application has received the message from the server. ■ Overdue - Not Received. The message is not delivered within the Message Delivery Timeout and has been removed from the user's server outbound queue. For information about the Message Delivery Timeout parameter, see "Configuring the Data Message Timeout Parameter for Store-and-Forward Messaging" on page 24. ■ Invalid. A message can no longer be generated due to record being deleted or reassigned to a different user. The message appears in the message table to log the fact that a workflow has been initiated, but something was wrong so a message cannot be generated. ■ Send Failed. The server outbound message cannot be delivered due to a transport error.
Time Received	Time a message was received.
Time Sent	Time a message was sent.
Transport ID	Transport-level ID used by the system.
User ID	User login name.

The following procedure describes how to delete message history records for all outbound messages.

To delete all old message history records

- 1 Navigate to Administration—Store-and-Forward > Store-and-Forward Message History.
- 2 In the Store-and-Forward Message History title bar, choose Menu > Delete Old Messages.

3 In the Store-and-Forward Delete All Messages Popup Applet, complete the fields in the following table.

Field	Comments
Repeat Weekly?	Select to automatically repeat the delete each week. Occurs at the same day and time each week.
Message Age	Enter a message age in minutes. The system will delete messages older than that.

4 Click Delete All Messages.

The following procedure describes how to review message history records for a specific user's outbound messages.

To review a user's message history records

- 1 Navigate to Administration—Store-and-Forward > User List > Store-and-Forward User Message History.
- 2 On the Persons list, select an end user whose messages you want to view.
- 3 Review the information on the Message History tab.

The table in ["To review all message history records" on page 44](#) describes the fields in the Message History list.

The following procedure describes how to delete a specific user's message history records for outbound messages.

To delete a user's old message history records

- 1 Navigate to Administration—Store-and-Forward > User List > Store-and-Forward User Message History.
- 2 On the Persons list, select an end user whose old messages you want to delete.
- 3 On the Message History tab, choose Menu > Delete My Old Messages.
- 4 In the Store-and-Forward Delete All Messages Popup Applet, in the Message Age field, enter a message age in minutes.

The system will delete this user's messages that are older than the age you specified.

- 5 Click Delete All Messages.

About Inspecting and Clearing Store-and-Forward Messaging Queues

From time to time, you may find that a Store-and-Forward Messaging queue is not operating properly. You can use the sfmutil.exe utility to inspect and clear inbound Store-and-Forward Messaging queues. You may need to do this after a server crash, a queue manager crash, or if you move a user from one queue to another without making sure that both inbound and outbound queues are empty.

For information about the command-line arguments that sfmutil.exe uses, see [Table 7](#). For information about how to use sfmutil.exe for specific operations, see ["Inspecting and Clearing Store-and-Forward Messaging Queues" on page 48](#).

Table 7. Arguments for the Sfmutil.exe Utility

Argument	Description
/A <i>file_name</i>	(Optional) Save a copy of the message to the specified file. Does not affect the presence of the message in the queue. If you specify the name of an existing file, sfmutil overwrites that file. In some circumstances, a Support representative may ask you for this file as part of the effort to resolve a problem with the queue.
/C <i>app_cfg_file</i>	Full path and file name for the server copy of the configuration file for your Siebel application, such as <i>path/uagent.cfg</i> . For sfmutil.exe to work properly, the instance of the file that resides on the server must contain a [SmqRTSReceiverSubsys] section such as the following: [SmqRTSReceiverSubsys] SmqReceiverTimeout = 30 SmqCompression = GZIP SmqEncryption = RC4 SmqQueueType = MQe SmqQueueManagerName = ServerQM SmqQueueName = InboundRTSQ SmqServerMachineName = Local host SmqServerPortNumber = 8080 SmqServerPathName = /smq/ServerQM
	NOTE: The values shown here are provided as one example. Your implementation may require you to use different values for some of these settings.
/D <i>table_owner</i>	Siebel Table Owner.
/L <i>language_code</i>	Language code. Default value is ENU for United States English.
/M Y	(Optional) For the first message in the queue, displays the internal message number and message type on screen. Does not affect the contents of the queue.

Table 7. Arguments for the Sfutil.exe Utility

Argument	Description
/N <i>mobile_web_client</i>	Name of any Mobile Web Client that may be affected by queue problems, as shown by the logs or queue folders. The sfutil.exe utility performs the actions that you specify on the individual inbound queue for this Mobile Web Client.
/P <i>password</i>	Password of the user that you specify with the /U argument.
/Q <i>queue_name</i>	Name of the queue.
/R Y	(Optional) Remove the first message in the queue.
/S <i>ODBC_data_source</i>	ODBC Data Source for the Siebel Enterprise database.
/U <i>userID</i>	User name of a Siebel Administrator.
/X <i>subsystem_name</i>	Name of the SMQ subsystem that is associated with the specified queue.
/Z Y	(Optional) Remove all messages from the queue. NOTE: When you remove messages using this argument, the messages are discarded and they cannot be recovered.

Inspecting and Clearing Store-and-Forward Messaging Queues

The sfutil.exe utility lets you perform the following actions on problematic Store-and-Forward Messaging queues for Mobile Web Clients:

- Gather information about the first message in a queue
- Save the first message in a queue to a file
- Remove the first message from a queue
- Remove all messages from a queue

This topic describes how to perform each of these actions. For more information about the sfutil.exe utility, see ["About Inspecting and Clearing Store-and-Forward Messaging Queues" on page 47](#).

The following procedure describes how to use sfutil.exe to gather information about the first message in a problematic queue.

To gather information about the first message in a problematic queue

- 1 On the Siebel Server where the queue resides, navigate to the siebsrvr\bin directory.
- 2 From the Windows Start menu, choose Start > Run, then enter cmd.exe and click OK to open a command window.
- 3 In the command window, enter the following command:

```
sfutil /u admin_userID /p password /s ODBC_data_source /D table_owner /c  
app_cfg_file /l language_code /N mobile_web_client /M Y /Q queue_name /X  
subsystem_name
```

Sfutil displays the message type and ID number on screen.

The following procedure describes how to use sfutil.exe to save a copy of the first message in a problematic queue to a file. In some circumstances, a Support representative may ask you for this file as part of the effort to resolve a problem.

To save the first message in a problematic queue to a file

- 1 On the Siebel Server where the queue resides, navigate to the siebsrvr\bin directory.
- 2 From the Windows Start menu, choose Start > Run, then enter cmd.exe and click OK to open a command window.
- 3 In the command window, enter the following command:

```
sfutil /u admin_userID /p password /s ODBC_data_source /D table_owner /c  
app_cfg_file /l language_code /N mobile_web_client /A file_name /Q queue_name /X  
subsystem_name
```

Sfutil writes a copy of the message to the file that you specify with the /A argument. The /A argument does not affect the contents of the queue.

The following procedure describes how to use sfutil.exe to remove the first message from a problematic queue.

To remove the first message from a problematic queue

- 1 On the Siebel Server where the queue resides, navigate to the siebsrvr\bin directory.
- 2 From the Windows Start menu, choose Start > Run, then enter cmd.exe and click OK to open a command window.
- 3 In the command window, enter the following command:

```
sfutil /u admin_userID /p password /s ODBC_data_source /D table_owner /c  
app_cfg_file /l language_code /N mobile_web_client /R Y /Q queue_name /X  
subsystem_name
```

The message is moved from InboundRTSQ to InboundRTSQ.*MWC_name*, where *MWC_name* is the name of the Mobile Web Client associated with the message. Then the message is deleted from the InboundRTSQ.*MWC_name* queue.

NOTE: If you combine the use of /A and /R arguments, sfutil saves a copy of the first message in the queue before deleting that message from the queue.

The following procedure describes how to use sfutil.exe to remove all messages from a problematic queue.

CAUTION: When you remove all messages from a queue, the messages are discarded and they cannot be recovered.

To remove all messages from a problematic queue

- 1 On the Siebel Server where the queue resides, navigate to the siebsrvr\bin directory.
- 2 From the Windows Start menu, choose Start > Run, then enter cmd.exe and click OK to open a command window.
- 3 In the command window, enter the following command:

```
sfutil /u admin_userID /p password /s ODBC_data_source /D table_owner /c  
app_cfg_file /l language_code /N mobile_web_client /Z Y /O queue_name /X  
subsystem_name
```

Sfutil removes all messages from the queue and discards them permanently.

a

This chapter describes the workflows that Store-and-Forward Messaging uses. The following topics are included in this chapter:

- ["About Store-and-Forward Messaging Workflows" on page 51](#)
- ["Dispatch Board - Generic Workflow" on page 52](#)
- ["SAF Activity Workflow" on page 53](#)
- ["Transaction Delete Message Workflows" on page 54](#)

About Store-and-Forward Messaging Workflows

Store-and-Forward Messaging provides sample workflows to demonstrate its functionality. You can customize these workflows to meet different business process requirements. You can examine the definitions of these workflows in Oracle's Siebel Tools. For more information about workflows, see *Siebel Business Process Framework: Workflow Guide*.

For Store-and-Forward Messaging for Mobile Web Client, The Oracle-provided workflows are Dispatch Board - Generic and SAF Activity. The following topics provide information about these two workflows.

Dispatch Board - Generic Workflow

This topic describes the Dispatch Board - Generic workflow for Store-and-Forward Messaging. The workflow is shown in [Figure 7](#).

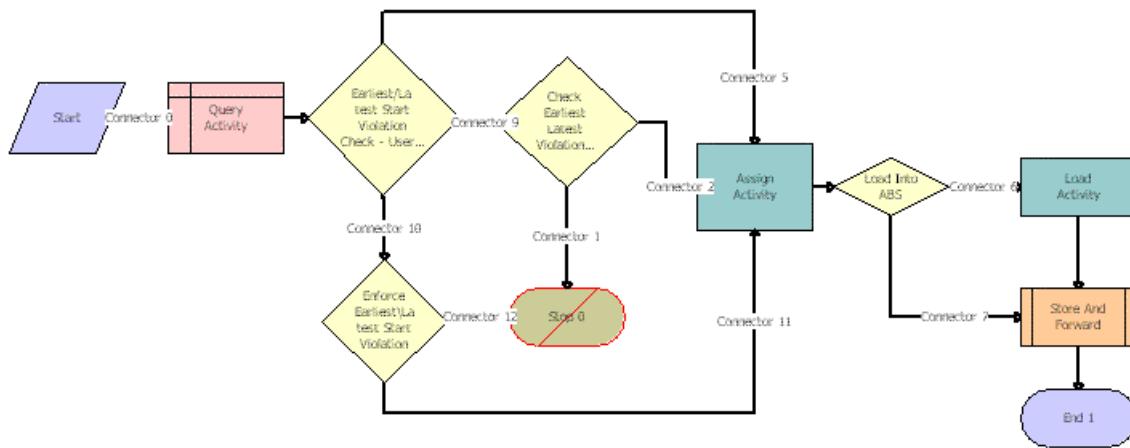


Figure 7. Dispatch Board - Generic Workflow

As shown in [Figure 7](#), the Dispatch Board - Generic workflow enforces start time restrictions during manual assignment of activities to field service engineers. It also calls the SAF - Activity workflow when Store-and-Forward messaging is the appropriate channel for dispatching the activity to the engineer. This workflow contains the following steps:

- 1 **Query Activity:** This step gets the information for the selected activity
- 2 **Earliest/Latest Start Violation Check - User Property:** This step checks whether an administrator has enabled or disabled a check for start and end time restrictions. Specifically, this step checks the value of the Enable Earliest/Latest Start Violation Check user property on the FS DB Planned GanttChart AX applet.
 - If the value is TRUE, the check is enabled, and the workflow proceeds to [Step 3](#).
 - If the value is not set, the workflow skips to [Step 4](#).
 - If the value is FALSE, the check is disabled, and the workflow skips to [Step 5](#).
- 3 **Enforce Earliest\Latest Start Violation:** This step enforces the check for start and end time restrictions.
 - If the current time is earlier than the earliest start time for the selected activity, or if the current time is later than the latest start time for the selected activity, the workflow stops, preventing the assignment of the activity.
 - If the current time falls within the allowed start time range for the selected activity, the workflow proceeds to [Step 5](#).

- 4 Check Earliest Latest Violation: This step checks whether the current user has enabled or disabled a check for start and end time restrictions. Specifically, this step checks the value of the Enable Earliest Latest Start Violation user preference on the Dispatch Board User Preference applet.
 - If the value of the user preference is TRUE, the workflow enforces the check for start and end time restrictions.
 - If the current time is earlier than the earliest start time for the selected activity, or if the current time is later than the latest start time for the selected activity, the workflow stops, preventing the assignment of the activity.
 - If the current time falls within the allowed start time range for the selected activity, the workflow proceeds to [Step 5](#).
 - If the value of the user preference is FALSE, the workflow proceeds to [Step 5](#) without checking whether the current time meets the specified start time restrictions.
- 5 Assign Activity: This step assigns the selected activity to the employee chosen by the user of the Dispatch Board.
- 6 Load into ABS: This step checks whether or not to use the Appointment Booking Engine (ABS) to schedule the activity.
 - If the system is set to use ABS, the workflow proceeds to [Step 7](#).
 - If the system is not set to use ABS, the workflow skips to [Step 8](#).
- 7 Load Activity: This step uses the Appointment Booking Engine (ABS) to schedule the activity.
- 8 Store and Forward: This step calls the SAF Activity workflow to use Store-and-Forward Messaging to dispatch the activity to the assigned employee.

SAF Activity Workflow

This topic describes the SAF Activity workflow for Store-and-Forward Messaging. The workflow is shown in [Figure 8](#).

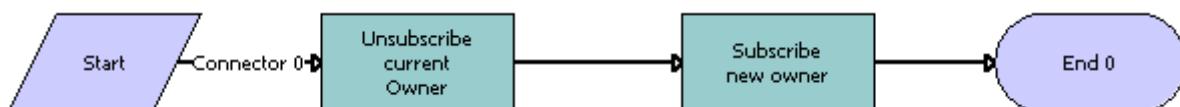


Figure 8. SAF Activity Workflow

As shown in [Figure 8](#), the SAF Activity workflow changes the owner of an Activity record. This Workflow is based on the Action business object and is invoked as a subprocess from the Dispatch Board - Generic workflow. This workflow consists of two steps that invoke the methods of the RTS Subscription Service so that an Activity can be subscribed to an owner and sent out using the Store-and-Forward communication channel.

This workflow has two process properties:

- **ActivityId.** This is the Row_Id of the Activity record that is being sent.

- **EmployeeId.** This is the Row_Id of the Employee who is being assigned to this Activity.

This workflow contains the following steps:

- 1 Unsubscribe Current Owner: This step calls the UnsubscribeAll business service method and is responsible for making sure that the current owner of the activity is unsubscribed as the owner. This step takes the following two input arguments:
 - a RowId. The ActivityId process property is passed to this input argument.
 - b TableName. This input argument has a hardcoded value of S_EVT_ACT (the table where Activities are stored).
- 2 Subscribe new owner: This step calls the SendSimple business service method and is responsible for subscribing the Activity to the new owner. This step takes the following three input arguments:
 - a RecipientId. The EmployeeId process property is passed to this input argument.
 - b RowId. The ActivityId process property is passed to this input argument.
 - c TableName. This input argument has a hardcoded value of S_EVT_ACT.

Transaction Delete Message Workflows

Your Siebel application comes with several sample workflows that work together to delete child records without deleting their parent records. This topic describes those workflows. This topic includes the following sub-topics:

- "About the Transaction Delete Message Workflows" on page 54
- "SMQ PreDelete FS Invoice Record" on page 55
- "SMQ Delete FS Invoice Record" on page 56
- "SMQ Outbound Delete in Action Business Object" on page 56
- "Configuring Transaction Delete Workflows for Different Business Components and Business Objects" on page 57

About the Transaction Delete Message Workflows

When a record is deleted, either manually by a user action or automatically by a system process, the system invokes the Delete method. This method triggers a PreDeleteRecord runtime event and a DeleteRecord runtime event. When a FS Invoice record is deleted from an Activity in the Action business object, those runtime events trigger two of the SMQ Delete sample workflows (SMQ PreDelete FS Invoice Record and SMQ Delete FS Invoice Record).

The SMQ Delete sample workflows can delete a child business component record while leaving its parent business component records undeleted. The workflows generate a Transaction (TXN) type Delete message. A transaction message delivers an action; in this case, a Delete action. The workflows send this TXN Delete message to a queue. The message will eventually be sent to the client application to delete the record. Each message deletes only one record.

The format of the TXN Delete message identifies the to-be-deleted record. The last <BC> is the record to be deleted. In the following example message, the child BC record is the one to be deleted.

```

<smq>
  <TXN mi d=" " act="D">
    <B0 nam=" " >
      <BC nam=" " rid=" " />
      ...
      <BC nam=" " rid=" " /> //grand parent BC
      <BC nam=" " rid=" " /> //parent BC
      <BC nam=" " rid=" " /> //child BC
    </B0>
  </TXN>
</smq>

```

All <BC>s above the last one are parent records. They identify the to-be-deleted record's position in the parent-child hierarchy. The message does not specify whether the delete is a cascading delete.

NOTE: These workflows function in the Action Object when a FS Invoice record is deleted from an activity parent record. You can custom-configure them for other business components or business objects. The workflows are capable of deleting any record whose business component is within a business object that has a primary business component defined. The restriction comes from the workflows.

SMQ PreDelete FS Invoice Record

This workflow saves the information of the to-be-deleted record; for example, its business component, its row ID, the parent business component, the parent records' row IDs if any, and the row ID of the primary business component's current record. When the PreDeleteRecord runtime event fires, it invokes this workflow. The workflow is shown in [Figure 9](#).



Figure 9. SMQ PreDelete FS Invoice Record Workflow

If you want to delete a record from a different business component, change the PreDeleteRecord runtime event's business component and set the ActionBusComp input argument in the Set PreDelete Information step to the PreDeleteRecord runtime event's business component.

SMQ Delete FS Invoice Record

This workflow invokes the [SMQ Outbound Delete in Action Business Object](#) workflow and passes the information saved by the [SMQ PreDelete FS Invoice Record](#) workflow to it. When the DeleteRecord runtime event fires, it invokes this workflow. The workflow is shown in [Figure 10](#).

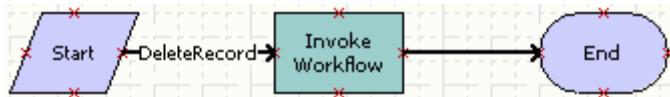


Figure 10. SMQ Delete FS Invoice Record Workflow

If you want to delete a record from a different business component, change the DeleteRecord runtime event's business component. See ["Configuring Transaction Delete Workflows for Different Business Components and Business Objects" on page 57](#).

SMQ Outbound Delete in Action Business Object

This workflow generates a TXN Delete message and sends the message. When the [SMQ Delete FS Invoice Record](#) workflow runs, it invokes this workflow. The workflow is shown in [Figure 11](#).

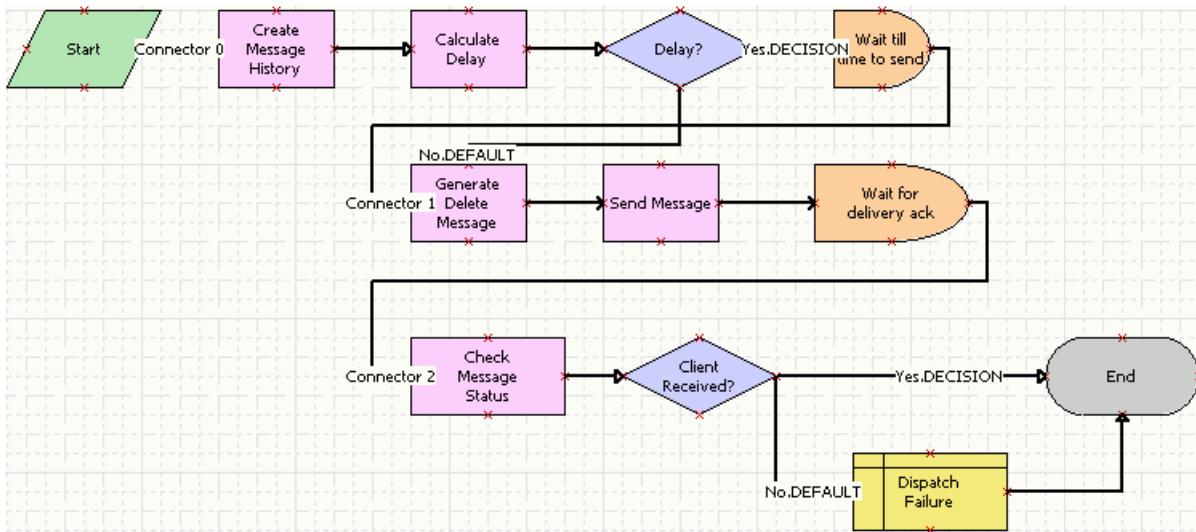


Figure 11. SMQ Outbound Delete in Action Business Object

If you want to delete a record from a different business component, but if both business components are in the same business object, you do not have to change this workflow.

About the Calculate Delay Step

Calculate Delay is not a required step. It is useful where the system assigns an activity to an owner and then deletes a FS Invoice record for that activity. In this case, Calculate Delay can prevent the workflow from sending a TXN Delete message before sending the DAT (Data) message generated when an activity is assigned to an owner. However, Calculate Delay cannot prevent the messages from being generated in the wrong order if the value of the Start-to-Send dependent field is changed. This is because the Calculate Delay step is based on the field value of a Start-to-Send dependent field. If that value is changed, the delay time is changed as well.

Configuring Transaction Delete Workflows for Different Business Components and Business Objects

You can configure the sample Transaction Delete workflows for a business component, a business object, or both, that are different from those specified in the workflows. To do so, you must identify which business component records in which business object you want the workflows to delete, and change that information in the workflows accordingly.

To customize the Transaction Delete workflows, perform one or more of the following procedures:

- ["Deleting a Record from Another Business Component" on page 57](#)
- ["Deleting a Record from Another Business Object" on page 58](#)
- ["Deleting a Record from Another Business Component and Business Object" on page 58](#)

Deleting a Record from Another Business Component

To configure the workflows to delete a record from a business component other than Invoice, but in the same (Action) business object that includes the Invoice business component, follow this procedure.

In this procedure, <BusComp> represents the new business component of the records to be deleted.

To delete a record from another business component

- 1 Copy the SMQ PreDelete FS Invoice Record workflow and do the following:
 - Name the new workflow differently from the original; for example, SMQ PreDelete <BusComp> Record.
 - Change the Event business component to <BusComp> for the PreDeleteRecord runtime event.
 - In the Set PreDelete Information step, set the ActionBusComp input argument to <BusComp>.
- 2 Copy the SMQ Delete FS Invoice Record workflow and do the following:
 - Name the new workflow differently from the original; for example, SMQ Delete <BusComp> Record.

- Change the Event business component to <BusComp> for the DeleteRecord runtime event.

NOTE: There is no need to change the SMQ Outbound Delete in Action Business Object workflow.

Deleting a Record from Another Business Object

To configure the workflows to delete Invoice business component records from a business object other than Action, follow this procedure.

In this procedure, <BusObj> represents the new business object of the records to be deleted.

To delete a record from another business object

- 1 Copy the SMQ PreDelete FS Invoice Record workflow and change the Business Object of the new workflow to <BusObj>.
- 2 Copy the SMQ Delete FS Invoice Record workflow and do the following:
 - Change the Business Object of the new workflow to <BusObj>.
 - In the Invoke Workflow step, set the ProcessName input argument to the workflow name you will designate in [Step 3](#).
- 3 Copy the SMQ Outbound Delete in Action Business Object workflow and do the following:
 - Name the new workflow differently from the original; for example, SMQ Outbound Delete in <BusObj> Business Object.
 - Change the Business Object for the new workflow to <BusObj>.
 - In the Generate Delete Message step, set the BusObj input argument to <BusObj>.
 - In the Generate Delete Message step, modify the User input argument if necessary.

For example, if you are setting up the workflows to delete records in the Account business object, you might need to change the User input argument to the Sales Rep field.

Deleting a Record from Another Business Component and Business Object

To configure the workflows to delete a record from a business component other than Invoice, and from a business object other than Action, follow this procedure.

In this procedure, <BusComp> represents the new business component of the records to be deleted, and <BusObj> represents the new business object of the records to be deleted.

To delete a record from another business component

- 1 Copy the SMQ PreDelete FS Invoice Record workflow and do the following:
 - Name the new workflow differently from the original; for example, SMQ PreDelete <BusComp> <BusObj> Record.
 - Change the Event business component to <BusComp> for the PreDeleteRecord runtime event.

- In the Set PreDelete Information step, set the ActionBusComp input argument to <BusComp>.
- Change the Business Object of the new workflow to <BusObj>.

2 Copy the SMQ Delete FS Invoice Record workflow and do the following:

- Name the new workflow differently from the original; for example, SMQ Delete <BusComp> <BusObj> Record.
- Change the Event business component to <BusComp> for the DeleteRecord runtime event.
- Change the Business Object of the new workflow to <BusObj>.
- In the Invoke Workflow step, set the ProcessName input argument to the workflow name you will designate in [Step 3](#).

3 Copy the SMQ Outbound Delete in Action Business Object workflow and do the following:

- Name the new workflow differently from the original; for example, SMQ Outbound Delete in <BusObj> <BusComp> Business Object.
- Change the Business Object for the new workflow to <BusObj>.
- In the Generate Delete Message step, set the BusObj input argument to <BusObj>.
- In the Generate Delete Message step, modify the User input argument if necessary.

For example, if you are setting up the workflows to delete records in the Account business object, you might need to change the User input argument to the Sales Rep field.

Index

A

administration

administration screens, about 43
Administration-Store-and-Forward screen views 16
message history views 16
message history, reviewing 43
outbound data message timeout Parameters, setting up 24

Administration-Store-and-Forward screen

about and views 16
message history views 16

architecture, logical architecture diagram

attachments, configuring maximum size

C

client-side checklist, troubleshooting

39

component groups, enabling

21

configuring Mobile Web Client applications

See installing and configuring Mobile Web Client application

D

data flow in Store-and-Forward

Messaging 17

delete transactions, workflows for

54

E

error messages, common server-side

38

F

functionality overview, Store-and-Forward Messaging

10

I

IBM MQe queue manager

client queue assignment, changing 33
deploying, about 26
MQe servlet, installing 27
SMQ subsystems, creating additional 30
SMQReceiver, deploying on UNIX systems 31
SMQReceiver, enabling 29

IBM WebSphere MQe, about

15

installing and configuring Mobile Web Client application

client-side checklist, troubleshooting 39
network connection, setting up 34
process tasks 33
registering 34
server-side checklist, troubleshooting 35
server-side common error messages 38

installing Store-and-Forward Messaging

component groups, enabling 21
maximum attachment size, configuring 25
Message Delivery Timeout parameter, configuring 24
prerequisites, about 21
process tasks 21
responsibilities, about adding 23
Store-and-Forward Messaging Integration Objects, creating 23
workflows, activating 26

L

logical architecture, diagram

11

M

maximum attachment size, configuring

25

Message Administration screen, about

43

Message Delivery Timeout parameter, configuring

24

message history

reviewing 43
views, about 16

Message Repository screen, about

43

messages

outbound data message timeout parameters, setting up 24

Mobile Web Client

queue, changing 33
Store-and-Forward Messaging, about configuring for 17

MQe servlet, installing

27

N

network connection, setting up

34

O

outbound data message timeout parameters

setting 24

outbound messages

message history views, about 16

message history, reviewing 43

R

registering a Mobile Web Client 34

responsibilities, about adding 23

S

Siebel Handheld 7

Siebel Handheld applications

configuration requirements, differing 17

Message History view 44

Siebel mobile products 7

Siebel Mobile Web Client and Siebel mobile products 7

Siebel Mobile Web Client applications

client-side checklist, troubleshooting 39

IBM WebSphere MQe, about using 15

installing and configuring tasks, list of 33

message history, reviewing 43

network connection, setting up 34

registering 34

server-side checklist, troubleshooting 35

server-side common error messages 38

SMQ subsystems, creating additional 30

Siebel Remote

attachments, handling of 25

Siebel mobile products, about 7

Store-and-Forward Messaging, shared data flow with 17

synchronization types handled by 11

Siebel Server

IBM MQe queue manager, about

deploying 26

server-side checklist, troubleshooting 35

server-side common error messages 38

Siebel Wireless 7

SMQ subsystems, creating additional 30

SMQ User responsibility, about adding 23

SMQReceiver

deploying on UNIX systems 31

enabling 29

Store-and-Forward Message History, about 16

Store-and-Forward Messaging Integration Objects, creating 23

Store-and-Forward User Message History, about 16

system administrator responsibilities 23

T

timeout, Message Delivery Timeout parameter 24

troubleshooting

client-side checklist, troubleshooting 39

overview 35

server-side checklist 35

server-side common error messages 38

U

UNIX systems, deploying SMQReceiver on 31

User Administration screen, about 43

W

wireless network, about connecting using 34

workflows

activating 26

Dispatch Board - Generic workflow 52

SAF Activity workflow 53

sample workflows, about 51

SMQ Delete FS Invoice Record workflow 56

SMQ Outbound Delete in Action Business

Object workflow 56

SMQ PreDelete FS Invoice Record

workflow 55

transaction delete workflows overview 54

transaction delete workflows, about 54

transaction delete workflows, configuring custom 57