



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

Version 8.0

December 2006

**ORACLE®**

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

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

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

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

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

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

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

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

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

# Contents

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

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

- About Siebel Mobile Products 7
- Overview of Siebel Store-and-Forward Messaging Functionality 9
- Overview of Store-and-Forward Messaging for Mobile Web Client Architecture 10
- About IBM WebSphere MQe 13
- Overview of Siebel Store-and-Forward Messaging Administration 14
- About Applications and Store-and-Forward Messaging for Mobile Web Client 15
- About Data Flow in Store-and-Forward Messaging for Mobile Web Client 15

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

- Prerequisites for Siebel Store-and-Forward Messaging Installation 19
- Process of Installing Siebel Store-and-Forward Messaging 19
  - Enabling Component Groups for Store-and-Forward Messaging 19
  - Adding Responsibilities for Store-and-Forward Messaging 21
  - About Configuring Store-and-Forward Messaging Integration Objects 21
  - Configuring the Data Message Timeout Parameter for Store-and-Forward Messaging 22
  - Configuring Maximum Attachment Size 23
  - Optional Step: Configuring Store-and-Forward Messaging Workflows 23
  - Activating Workflows for Store-and-Forward Messaging 24
- Deploying IBM MQe Queue Managers 24
  - Installing the MQe Servlet 24
  - Enabling the SMQReceiver 26
  - Optional Step: Creating Additional SMQ Subsystems 27
  - Deploying SMQReceivers on UNIX Systems 27
  - Setting Up Multiple Queue Managers 28
- Process of Installing and Configuring Mobile Web Client Applications 29
  - Setting SMQ Values for the Mobile Web Client 29
  - Changing the Queue a Mobile Web Client Uses 31

Setting Up a Network Connection for Mobile Web Client	31
Registering a Mobile Web Client for Store-and-Forward Messaging	32
Troubleshooting Store-and-Forward Messaging Setup and Configuration	32
Siebel Store-and-Forward Messaging Server-Side Troubleshooting Checklist	32
Siebel Store-and-Forward Messaging Server-Side Common Error Messages	35
Siebel Store-and-Forward Messaging Client-Side Troubleshooting Checklist	37

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

Overview of Siebel Store-and-Forward Messaging Administration	39
Reviewing and Deleting Store-and-Forward Messaging History	39

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

About Store-and-Forward Messaging Workflows	43
Dispatch Board - Generic Workflow	44
SAF Activity Workflow	45
Transaction Delete Message Workflows	46
About the Transaction Delete Message Workflows	46
SMQ PreDelete FS Invoice Record	47
SMQ Delete FS Invoice Record	48
SMQ Outbound Delete in Action Business Object	48
Configuring Transaction Delete Workflows for Different Business Components and Business Objects	49

## **Index**

# 1

## What's New in This Release

### What's New in Siebel Store-and-Forward Messaging Guide for Mobile Web Client, Version 8.0

Release 8.0 is the first Siebel release to include Store-and-Forward Messaging support for Mobile Web Clients. Version 7.7 and version 7.8 Siebel applications supported Store-and-Forward Messaging for handheld devices only.

Table 1 lists the changes that were made to an earlier *Siebel Store-and-Forward Messaging Guide* in order to describe Store-and-Forward support for Mobile Web Clients in version 8.0. For information about the changes that were made to describe Store-and-Forward support for handheld devices in version 8.0, see *Siebel Store-and-Forward Messaging Guide for Handheld*.

Table 1. New Product Features in *Siebel Store-and-Forward Messaging Guide for Mobile Web Client, Version 8.0*

Topic	Description
<a href="#">"Overview of Siebel Store-and-Forward Messaging Administration" on page 14</a>	Modified this topic to provide information about the administration views available for Store-and-Forward Messaging Administration for Mobile Web Clients.
<a href="#">"Enabling Component Groups for Store-and-Forward Messaging" on page 19</a>	Modified this topic to match the requirements for using Store-and-Forward Messaging with Mobile Web Clients.
<a href="#">"Adding Responsibilities for Store-and-Forward Messaging" on page 21</a>	Modified this topic to match the requirements for using Store-and-Forward Messaging with Mobile Web Clients.
<a href="#">"About Configuring Store-and-Forward Messaging Integration Objects" on page 21</a>	New topic describing the configuration of integration objects. These objects are required if you want to use Store-and-Forward Messaging with business components other than Activities.
<a href="#">"Overview of Siebel Store-and-Forward Messaging Functionality" on page 9</a>	Added information about the network configurations supported for Store-and-Forward Messaging for Mobile Web Clients.
<a href="#">"Overview of Store-and-Forward Messaging for Mobile Web Client Architecture" on page 10</a>	Added a new topic describing the architecture of Store-and-Forward Messaging for Mobile Web Client.
<a href="#">"Overview of Siebel Store-and-Forward Messaging Administration" on page 14</a>	Modified this topic to provide information about administering Store-and-Forward Messaging for Mobile Web Clients.

Table 1. New Product Features in *Siebel Store-and-Forward Messaging Guide for Mobile Web Client*, Version 8.0

Topic	Description
<a href="#">“About Applications and Store-and-Forward Messaging for Mobile Web Client” on page 15</a>	Added a new topic describing the Siebel applications you can use with Store-and-Forward Messaging for Mobile Web Client.
<a href="#">“About Data Flow in Store-and-Forward Messaging for Mobile Web Client” on page 15</a>	Added a new topic describing the steps Store-and-Forward Messaging for Mobile Web Client goes through for typical operations.
<a href="#">“Configuring the Data Message Timeout Parameter for Store-and-Forward Messaging” on page 22</a>	Modified this topic to match the requirements for using Store-and-Forward Messaging with Mobile Web Clients.
<a href="#">“Optional Step: Configuring Store-and-Forward Messaging Workflows” on page 23</a>	Modified this topic to match the requirements for using Store-and-Forward Messaging with Mobile Web Clients.
<a href="#">“Activating Workflows for Store-and-Forward Messaging” on page 24</a>	Modified this topic to match the requirements for using Store-and-Forward Messaging with Mobile Web Clients.
<a href="#">“Enabling the SMQReceiver” on page 26</a>	Modified this topic to match the requirements for using Store-and-Forward Messaging with Mobile Web Clients.
<a href="#">“Optional Step: Configuring Store-and-Forward Messaging Workflows” on page 23</a>	Added information about using multiple Siebel Messaging and Queuing (SMQ) subsystems in Siebel implementations that have many Store-and-Forward users.
<a href="#">“Siebel Store-and-Forward Messaging Server-Side Troubleshooting Checklist” on page 32</a>	Modified this topic to adjust the troubleshooting details for using Store-and-Forward Messaging with Mobile Web Clients.
<a href="#">“Overview of Siebel Store-and-Forward Messaging Administration” on page 39</a>	Added more information about a Siebel responsibility required for administrative accounts.
<a href="#">“Reviewing and Deleting Store-and-Forward Messaging History” on page 39</a>	Added information about two new fields in the Message History list.

## Additional Changes

This version of the documentation also removed all references to the Asynchronous Query feature. Asynchronous Query is available for Siebel Handheld applications but not for Mobile Web Client.

# 2

## Overview of Siebel Store-and-Forward Messaging

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 9](#)
- [“Overview of Store-and-Forward Messaging for Mobile Web Client Architecture” on page 10](#)
- [“About IBM WebSphere MQe” on page 13](#)
- [“Overview of Siebel Store-and-Forward Messaging Administration” on page 14](#)
- [“About Applications and Store-and-Forward Messaging for Mobile Web Client” on page 15](#)
- [“About Data Flow in Store-and-Forward Messaging for Mobile Web Client” on page 15](#)

### 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, *form factors* refers to the general capabilities, screen shapes, and sizes of the various devices shown in the illustration.

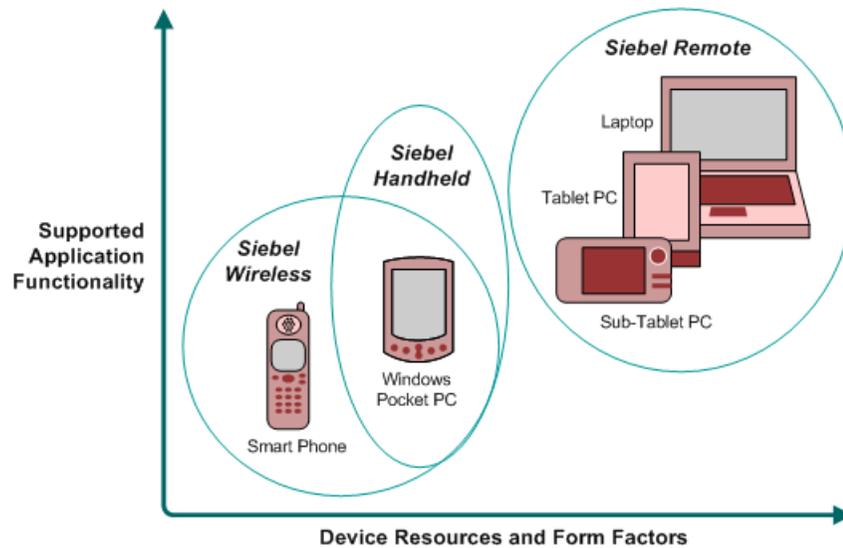


Figure 1. Siebel Mobile Product Platforms

Table 2 describes the Siebel mobile products' software components and supported hardware devices.

Table 2. Siebel Mobile Product Components and Supported Devices

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

Table 3 describes the Siebel mobile products' supported Siebel applications and reference sources on your Siebel Bookshelf for information about these products.

Table 3. Siebel Mobile Product Supported Applications and References

Product Group	Supported Applications	Reference Sources
<b>Siebel Remote</b>	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>
<b>Siebel Handheld</b>	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>
<b>Siebel Wireless</b>	Siebel Sales Siebel Field Service Siebel Partner Relationship Management Siebel Self-Service Applications	<i>Siebel Wireless Administration Guide</i>

## 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 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 queuing software, in the context of a Siebel Business Application deployment that includes Siebel Remote.

Table 4 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 4. 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"> <li>■ All local database transactions.</li> <li>■ File attachments smaller than the maximum attachment size set by the administrator. For more information about this parameter, see <a href="#">"Configuring Maximum Attachment Size" on page 23.</a></li> </ul>	<ul style="list-style-type: none"> <li>■ File attachments larger than the maximum attachment size set by the administrator. For more information about this parameter, see <a href="#">"Configuring Maximum Attachment Size" on page 23.</a></li> </ul>
Server to Mobile Web Client	<ul style="list-style-type: none"> <li>■ New Activities assigned by means of the Dispatch Board to a Mobile Web Client that has Store-and-Forward.</li> <li>■ Any enterprise database transactions that update existing Activities that have already been sent to the Mobile Web Client using Store-and-Forward Messaging.</li> </ul>	<ul style="list-style-type: none"> <li>■ 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.</li> <li>■ All file attachments, regardless of size.</li> </ul>

Figure 2 shows the components that are used to transmit and receive messages using Store-and-Forward Messaging. Figure 3 shows the mixture of Siebel Remote components and Store-and-Forward components that are used to transmit all file attachments from the server and receive them on the client, and to transmit large file attachments from the client and receive them on the server.

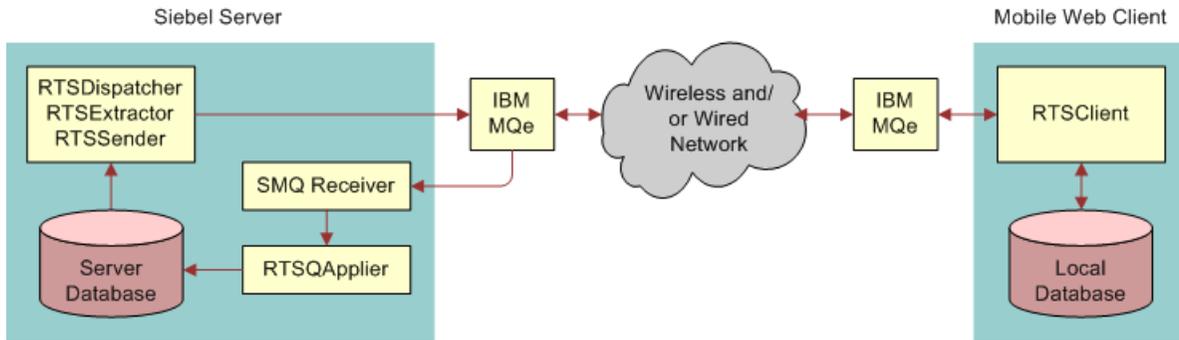


Figure 2. Components Used to Transmit and Receive Store-and-Forward Messages

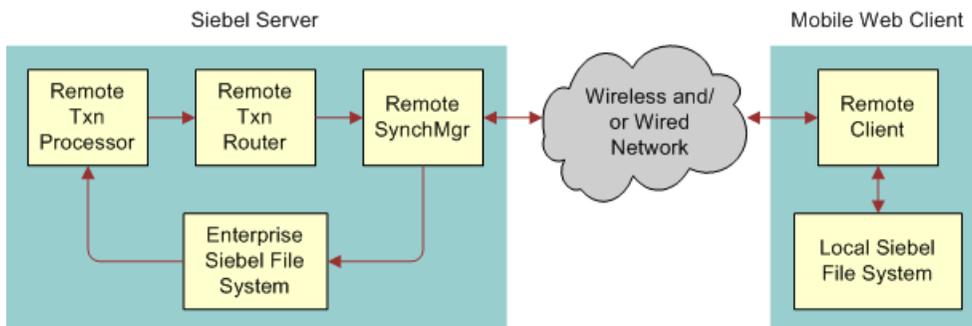


Figure 3. Siebel Remote Components Used Instead of Store-and-Forward for Attachments

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 clients should 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 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 13](#).

## RTSQApplier

This server component retrieves messages from the SMQ Receiver server component and applies the updates that are contained in the messages to the server 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 client to the server as Store-and-Forward messages
- Receiving Store-and-Forward messages from the server and applying the database changes in the messages to the 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 server to the client. If you want to use Store-and-Forward Messaging to deliver other information types from server to client, contact Siebel 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 21](#). For more information about setting timeout values, see [“Configuring the Data Message Timeout Parameter for Store-and-Forward Messaging” on page 22](#).

## 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 those described above. 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 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 clients should 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 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 client's screen.
  - b It applies the database change that is contained in the message to the local database.
  - c displays an outgoing message alert icon (a red arrow pointing up from an envelope) in the notification area, indicating that it is sending the 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 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 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 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 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.



# 3

## Installation and Configuration of Siebel Store-and-Forward Messaging

The following topics are included in this chapter:

- [“Prerequisites for Siebel Store-and-Forward Messaging Installation” on page 19](#)
- [“Process of Installing Siebel Store-and-Forward Messaging” on page 19](#)
- [“Deploying IBM MQe Queue Managers” on page 24](#)
- [“Process of Installing and Configuring Mobile Web Client Applications” on page 29](#)
- [“Troubleshooting Store-and-Forward Messaging Setup and Configuration” on page 32](#)

### 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 19](#)
- [“Adding Responsibilities for Store-and-Forward Messaging” on page 21](#)
- [“About Configuring Store-and-Forward Messaging Integration Objects” on page 21](#)
- [“Configuring the Data Message Timeout Parameter for Store-and-Forward Messaging” on page 22](#)
- [“Configuring Maximum Attachment Size” on page 23](#)
- [“Optional Step: Configuring Store-and-Forward Messaging Workflows” on page 23](#)
- [“Activating Workflows for Store-and-Forward Messaging” on page 24](#)

### 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.

*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
<b>Enterprise Application Integration</b>	Used for Store-and-Forward Messaging server components such as SMQReceiver.
<b>Field Service Object Manager or Field Service</b>	Used to enable Siebel Field Service related features and Store-and-Forward Messaging workflows. For deployment of Store-and-Forward Messaging for Mobile Web Client, 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.
<b>MWC Real Time Sync</b>	Used for Store-and-Forward Messaging components such as RTSExtractor, RTSSender, RTSQApplier, and RTSDispatcher.
<b>Siebel Call Center</b>	Used to enable Siebel Call Center-related features.
<b>Siebel Sales</b>	Used to enable Siebel Sales-related features.

**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 and restart the Siebel Gateway Server.
- 6 Stop and restart the Siebel Server.

## Adding Responsibilities for Store-and-Forward Messaging

The responsibilities listed in [Table 5](#) are related to Store-and-Forward Messaging system administrator or users.

Table 5. Responsibilities for Store-and-Forward Messaging

Responsibility	Comments
<b>SMQ User</b>	Allows access to the Store-and-Forward Messaging administration views.  <b>NOTE:</b> Implementations that use AIX servers may need to add this responsibility for any Store-and-Forward administration accounts other than SADMIN. Other environments do not require that you add this responsibility for administrators.
<b>Standard Client Responsibilities</b>	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.
<b>Standard Siebel Administrator Responsibilities</b>	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. 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 Siebel 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 Siebel 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, so you must change this value in order to provide 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 substeps:
  - 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 substeps:
  - 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 client to the 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 server to the client travel by means of Siebel Remote synchronization, not by Store-and-Forwarding.

### *To configure the maximum attachment size for Store-and-Forward*

- 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 client to the 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.

## 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. IBM MQe is a Java-based product, and requires certain Java configurations.

The tasks for deploying MQe Queue Managers are as follows:

- [“Installing the MQe Servlet” on page 24](#)
- [“Enabling the SMQReceiver” on page 26](#)
- [“Optional Step: Creating Additional SMQ Subsystems” on page 27](#)
- [“Deploying SMQReceivers on UNIX Systems” on page 27](#)
- [“Setting Up Multiple Queue Managers” on page 28](#)

## 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 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 Siebel SupportWeb.

**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 Siebel SupportWeb.
- 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 comes as a Zip file delivered as part of the Siebel Server installation package in <Siebel\_Server>\classes\SiebelSMQServlet.war.

**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 server or engine. Alternatively, it might mean running a configuration utility built into the server or engine. 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 Open the web.xml file in a text editor and change the DataDir parameter to specify the directory where you want the queues to be stored.
- 5 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:

```
srvmgr /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 once to start the SMQReceiver.

```
start task for comp SMQReceiver with  
ReceiverConnectionSubsystem=SmqRTSReceiverSubsys,  
ReceiverDataHandlingSubsystem=SmqRTSDispatchSubsys,  
ReceiverMethodName=ReceiverDispatch
```

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, SmqDispatchRTSDispatchSubsys).

- 3 From the server manager, run the following commands:

```
Start task for comp RTSDispatcher
```

```
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 server manager, execute the following command:

```
create named subsystem SmqRTSReceiverSubsys1 for subsystem SMQRcvSubsys with  
SmqQueueType=MQe, SmqQueueName=InboundRTSQ, SmqQueueManagerName=ServerQM,  
SmqServletPathName=/smqServerQM, SmqServerMachineName=sdch70a012,  
SmqServerPortNumber=8080, SmqTransportServiceName="SMQ Transport Service",  
SmqTransportMethodName="WaitForMessage", SmqCompression="GZIP", SmqEncryption="RC4",  
SmqProcessComponentName="RTSAppIier", SmqProcessMethodName="RunRTSAppIier",  
SmqProcessServiceName="RTS Message Apply Service"
```

**NOTE:** The value for SmqServerPortNumber should 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, you would use the following server manager command:

```
start task for comp SMQReceiver1 with  
ReceiverConnectionSubsystem=SmqRTSReceiverSubsys1,  
ReceiverDataHandlingSubsystem=SmqRTSDispatchSubsys,  
ReceiverMethodName=ReceiverDispatch
```

## Deploying SMQReceivers on UNIX Systems

To deploy SMQReceivers in a UNIX environment, perform the applicable procedure.

### 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 Siebel SupportWeb.

- 2 Set the following environment variables:

```
setenv LD_LIBRARY_PATH {LD_LIBRARY_PATH}: /usr/jdk/jdk1.5.0_01/jre/lib/sparc/server
```

```
setenv JAVA_HOME /usr/jdk/jdk1.5.0_01
```

- 3 Make sure the directory that contains libjvm.so is in LD\_LIBRARY\_PATH.

### **To install SMQReceivers on UNIX, AIX platform**

- 1 Set the following environment variables:

```
setenv LIBPATH {LIBPATH}: /usr/java1.5/sdk/jre/bin: /usr/java1.5/sdk/jre/bin/j9vm/
```

```
setenv JAVA_HOME /usr/java1.5/sdk
```

- 2 Create SmqJVMSubsys: Use Server Manager and issue the following commands:

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

```
change param CLASSPATH="<InstallDir>/siebsvr/classes/SiebelJI_enu.jar: <InstallDir>/siebsvr/classes/Siebel.jar" for named subsystem SmqJVMSubsys
```

### **To install SMQReceivers on UNIX, Redhat Linux or SUSE Linux platform**

- 1 The EAI component group is needed to run SMQReceiver.

- 2 Make sure the directory that contains libjvm.so is in LD\_LIBRARY\_PATH.

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

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

```
change param CLASSPATH="{SIEBEL_HOME}/siebsvr/classes/SiebelJI_enu.jar: {SIEBEL_HOME}/siebsvr/classes/Siebel.jar" for named subsystem SmqJVMSubsys
```

## **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 27. For information about specifying a subsystem for an SMQ Receiver, see [“Enabling the SMQReceiver”](#) on page 26.

## 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 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 .cfg file your clients will use. See [“Setting SMQ Values for the Mobile Web Client”](#) on page 29.
- 3 Set up the client’s network connection. See [“Setting Up a Network Connection for Mobile Web Client”](#) on page 31.
- 4 Assign the client to an SMQ Receiver subsystem. See [“Registering a Mobile Web Client for Store-and-Forward Messaging”](#) on page 32.
- 5 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.

### Setting SMQ Values for the Mobile Web Client

As part of the preparation for deploying Store-and-Forward Messaging for Mobile Web Clients, you must modify the Siebel application .cfg file or files that your Mobile Web Clients will use, to include information about SMQ. The following procedure describes how set the appropriate values.

#### *To set SMQ Values in a Siebel application .cfg file*

- 1 Locate the .cfg file for the Siebel application that your Mobile Web Clients will use.

For example, the .cfg file for Siebel Service is sfs.cfg, and the .cfg file for Siebel Sales is siebel.cfg. These files are normally located in the *siebsrvr\_root\bin\language* directory on your Siebel Server. (When Mobile Web Client setup is complete, they also reside in the *siebel\_client\_root\bin\language* directory on the Mobile Web Client.

- 2 Use a text editor to set up values in the [SMQ] section, as shown in the following table.

For more information about IBM MQe, see [“Deploying IBM MQe Queue Managers” on page 24](#).

Setting Name	Comment
MqeServerAddr	The Server domain name or IP address of the MQe server that hosts the Queue Manager to which you are assigning the Mobile Web Client. Make sure that the domain name or IP address is resolvable from the Mobile Web Clients in the deployed environment.
MqeServerCmd	Address used to reach the queue manager that runs as a servlet. The default value is /smq/ServerQM.
MqeServerPort	Number of the port to which the assigned Queue Manager is listening. The default value is 8082. The value of this setting should match the value of the subsystem parameter SmqServerPortNumber, for the subsystem that will communicate with this client's Queue Manager. This value should also match the port number for the Web server where you deploy the MQe Queue Manager servlet. For example, if you are using the default port on a Tomcat Web server, you would change the default value of 8082 to 8080.
MqeServerQMgr	Name of the MQe server Queue Manager to which you are assigning the Mobile Web Client. The default value is ServerQM. If you are assigning this client to a non-default Queue Manager, the value of this setting should match the value you specified for the subsystem parameter SmqQueueManagerName, when you created the subsystem that will communicate with this client's Queue Manager.
MqeServerQueue	Name of the inbound MQe server queue. The default value is InboundRTSQueue. If you are assigning this client to a non-default Queue Manager, the value of this setting should match the value you specified for the subsystem parameter SmqQueueName, when you created the subsystem that will communicate with this client's Queue Manager.
QueueingSoftware	The type of queueing software used for Store-and-Forward Messaging for Mobile Web Client. The only value currently supported is MQe.
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.

Setting Name	Comment
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.

**NOTE:** If you need to modify queue manager information for one or more users, see [“Changing the Queue a Mobile Web Client Uses”](#).

## 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 client or the server.
- 2 On the server, 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 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 client.
- 6 Reextract the local database for the client.

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

- 7 Edit the client's .cfg files as needed to specify the new queue name.

For more information, see [“Setting SMQ Values for the Mobile Web Client”](#) on page 29.

## 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 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 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 26](#), and [“Optional Step: Creating Additional SMQ Subsystems” on page 27](#).

**NOTE:** This task must be performed before you extract the local database for the Mobile Web Client. Any time you change whether or not a 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 client obtains during local database initialization. When a 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 32](#)
- [“Siebel Store-and-Forward Messaging Server-Side Common Error Messages” on page 35](#)
- [“Siebel Store-and-Forward Messaging Client-Side Troubleshooting Checklist” on page 37](#)

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

- 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 SMQReceiver
```

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

- 4 Make sure that RTSDispatcher, 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 RTSDispatcher
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.

- 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>\MQe\ServerQM\Queues\ServerQM
  AdminQ
  AdminReplyQ
  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 40](#).
- 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, contact Siebel Technical Support for assistance.

- 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 evtloglvl SMQMsgSvc=5 for comp SMQReceiver

change evtloglvl JVMInit=5 for comp SMQReceiver

change evtloglvl EAITransport=5 for comp SMQReceiver

- For RTSDispatcher, RTSExtractor, RTSSender and RTSQApplier,

change evtloglvl RTSEventLog = 5 for comp RTSDispatcher

change evtloglvl RTSEventLog = 5 for comp RTSExtractor

change evtloglvl RTSEventLog = 5 for comp RTSSender

change evtloglvl RTSEventLog = 5 for comp RTSQApplier

- 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 6](#).

Table 6. Server-Side Common Error Messages

Symptom/Error Message	Diagnostic Steps/Cause	Solution
MQe Internal Error: RemoveMsg: Message inside of Inbound Queue is not deleted successfully. Error Code: 121	<p>Error Code 121 means there is no matching message in the queue.</p> <p>Multiple Receivers or Tasks may be trying to process or dispatch from the same queue at the same time.</p>	<p>Check srvmgr to find out how many SMQReceivers are running in the server.</p> <p>There should be only one SMQReceiver for each application.</p>
ObjMgrBusServiceLog Error 1 000008e0440515c0:0 2006-03-02 00:05:23 (srmsyncsv.cpp (1214)) SBL-SRQ-00103: Unable to find definition for component WfProcMgr	<p>This error may appear in the SMQReceiver log file.</p> <p>The server components may not be synchronized to the internal table.</p>	<ol style="list-style-type: none"> <li>1 In the Administration - Server Configuration &gt; Synchronize screen, click Synchronize.</li> <li>2 Restart the SMQReceiver component.</li> </ol>

Table 6. Server-Side Common Error Messages

Symptom/Error Message	Diagnostic Steps/Cause	Solution
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:  <ul style="list-style-type: none"> <li>■ env</li> <li>■ echo \$LD_LIBRARY_PATH</li> </ul> On AIX, check \$LIBPATH.
Java HotSpot(TM) Server VM (build 1.5.0_06-b05, mixed mode) IBM WebSphere MQ Everywhere 2.0.2.4 Created session for subsystem SmqReceiverSubsys GetClientNames: com.ibm.mqe.MQException: Message not found[ServerQM 27] (code=121) at com.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

- 1 Verify that the following parameters are in the [SMQ] section of the .cfg file for the Siebel application. Table 7 lists the default values.

Table 7. Parameter Values for the .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.

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

**NOTE:** The value of the MQeServerPort parameter should 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.



# 4

## Administration of Siebel Store-and-Forward Messaging

The following topics are included in this chapter:

- [“Overview of Siebel Store-and-Forward Messaging Administration” on page 39](#)
- [“Reviewing and Deleting Store-and-Forward Messaging History” on page 39](#)

### 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 SMO 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 21](#).

### 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 40](#)
- ["To delete all old message history records" on page 41](#)
- ["To review a user's message history records" on page 42](#)
- ["To delete a user's old message history records" on page 42](#)

### ***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"> <li>■ <b>Pending.</b> 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 <a href="#">“Configuring the Data Message Timeout Parameter for Store-and-Forward Messaging”</a> on page 22.</li> <li>■ <b>Sent.</b> The server outbound message has been sent to the queuing system.</li> <li>■ <b>Server Received.</b> The server has received the message from the Siebel Mobile Web Client application.</li> <li>■ <b>Client Received.</b> The Mobile Web Client application has received the message from the server.</li> <li>■ <b>Overdue - Not Received.</b> 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 <a href="#">“Configuring the Data Message Timeout Parameter for Store-and-Forward Messaging”</a> on page 22.</li> <li>■ <b>Invalid.</b> 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.</li> <li>■ <b>Send Failed.</b> The server outbound message cannot be delivered due to a transport error.</li> </ul>
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.

**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.

#### ***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 40](#) describes the fields in the Message History list.

#### ***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.

# 5

## Siebel Store-and-Forward Messaging Sample Workflows

The following topics are included in this chapter:

- [“About Store-and-Forward Messaging Workflows” on page 43](#)
- [“Dispatch Board - Generic Workflow” on page 44](#)
- [“SAF Activity Workflow” on page 45](#)
- [“Transaction Delete Message Workflows” on page 46](#)

### 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 4](#).

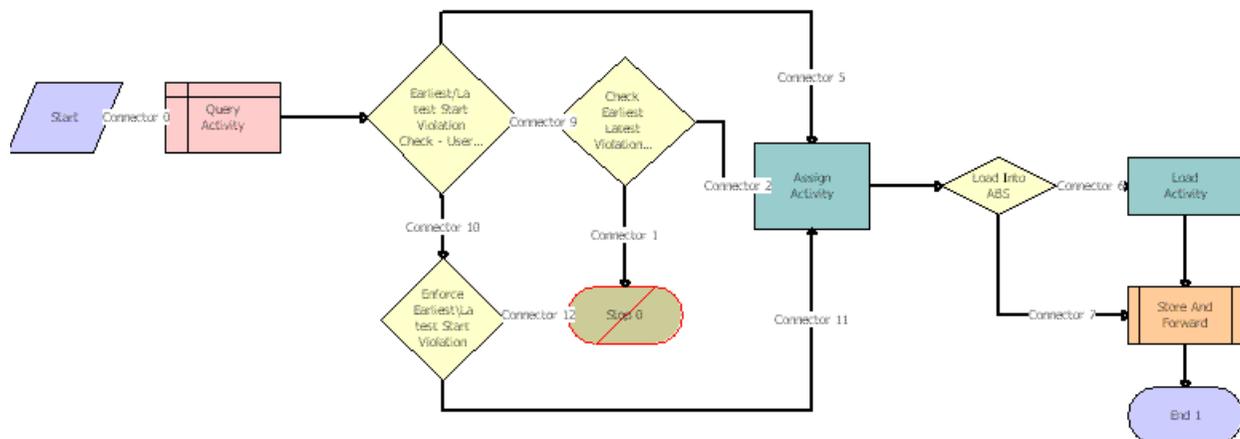


Figure 4. Dispatch Board - Generic Workflow

As shown in [Figure 4](#), 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 5](#).

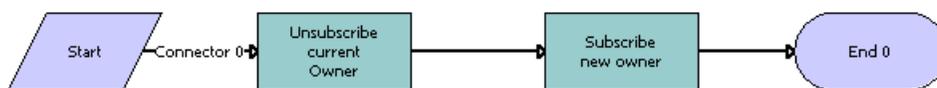


Figure 5. SAF Activity Workflow

As shown in [Figure 5](#), 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 2 steps which invoke 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 46](#)
- [“SMQ PreDelete FS Invoice Record” on page 47](#)
- [“SMQ Delete FS Invoice Record” on page 48](#)
- [“SMQ Outbound Delete in Action Business Object” on page 48](#)
- [“Configuring Transaction Delete Workflows for Different Business Components and Business Objects” on page 49](#)

### 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">
    <BO nam=" ">
      <BC nam=" " ri d=" "/>
      ...
      <BC nam=" " ri d=" "/> //grand parent BC
      <BC nam=" " ri d=" "/> //parent BC
      <BC nam=" " ri d=" "/> //chi l d BC
    </BO>
  </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 6.



Figure 6. 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 7](#).



Figure 7. 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 49.

## 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 8](#).

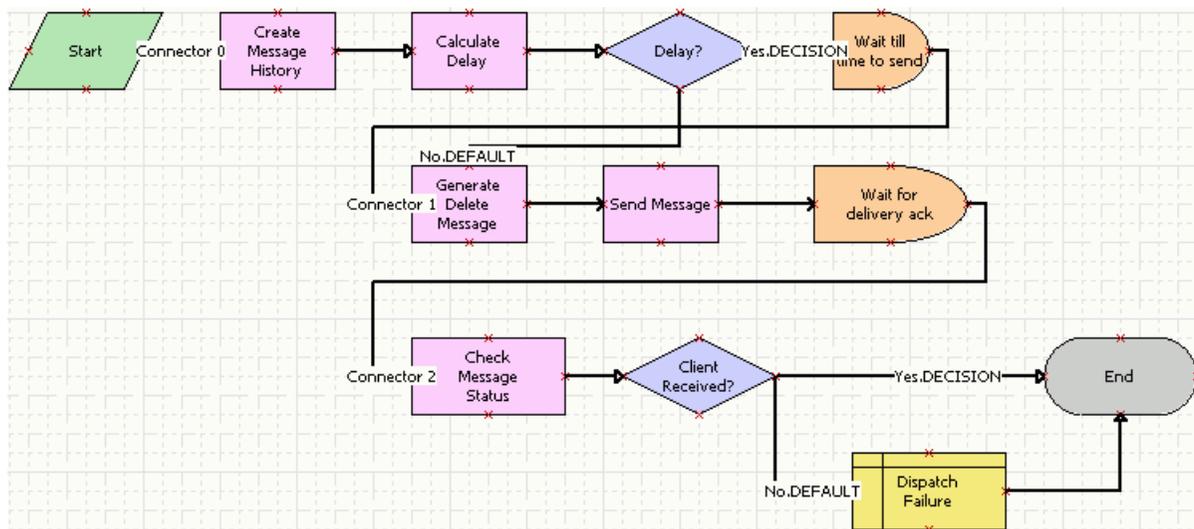


Figure 8. SMQ Outbound Delete in Action Business Object

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 49](#)
- [“Deleting a Record from Another Business Object” on page 50](#)
- [“Deleting a Record from Another Business Component and Business Object” on page 50](#)

### 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 39
- Administration-Store-and-Forward screen
  - views 14
- message history views 14
- message history, reviewing 39
- outbound data message timeout Parameters, setting up 22

### Administration-Store-and-Forward screen

- about and views 14
- message history views 14

### architecture, logical architecture diagram 10

### attachments, configuring maximum size 23

## C

### client-side checklist, troubleshooting 37

### component groups, enabling 19

### configuring

- See installing and configuring Mobile Web Client application

## D

### data flow in Store-and-Forward

- Messaging 15

### delete transactions, workflows for 46

## E

### error messages, common server-side 35

## F

### functionality overview, Store-and-Forward

- Messaging 9

## I

### IBM MQe queue manager

- client queue assignment, changing 31
- deploying, about 24
- MQe servlet, installing 24
- SMQ subsystems, creating additional 27
- SMQReceiver, deploying on UNIX systems 27
- SMQReceiver, enabling 26

### IBM WebSphere MQe, about 13

### installing and configuring Mobile Web Client application

- client-side checklist, troubleshooting 37
- network connection, setting up 31
- process tasks 29
- registering 32
- server-side checklist, troubleshooting 32
- server-side common error messages 35

### installing Store-and-Forward Messaging

- component groups, enabling 19
- maximum attachment size, configuring 23
- Message Delivery Timeout parameter, configuring 22
- prerequisites, about 19
- process tasks 19
- responsibilities, about adding 21
- Store-and-Forward Messaging Integration Objects, creating 21
- workflows, activating 24

## L

### logical architecture, diagram 10

## M

### maximum attachment size, configuring 23

### Message Administration screen, about 39

### Message Delivery Timeout parameter, configuring 22

### message history

- reviewing 39
- views, about 14

### Message Repository screen, about 39

### messages

- outbound data message timeout parameters, setting up 22

### Mobile Web Client

- queue, changing 31
- Store-and-Forward Messaging, about configuring for 15

### MQe servlet, installing 24

## N

### network connection, setting up 31

## O

### outbound data message timeout parameters

setting 22

### outbound messages

message history views, about 14

message history, reviewing 39

## R

registering a Mobile Web Client 32

responsibilities, about adding 21

## S

Siebel Handheld 7

### Siebel Handheld applications

configuration requirements, differing 15

Message History view 39

Siebel mobile products 7

Siebel Mobile Web Client and Siebel mobile products 7

### Siebel Mobile Web Client applications

client-side checklist, troubleshooting 37

IBM WebSphere MQe, about using 13

installing and configuring tasks, list of 29

message history, reviewing 39

network connection, setting up 31

registering 32

server-side checklist, troubleshooting 32

server-side common error messages 35

SMQ subsystems, creating additional 27

### Siebel Remote

attachments, handling of 23

Siebel mobile products, about 7

Store-and-Forward Messaging, shared data flow with 15

synchronization types handled by 10

### Siebel Server

IBM MQe queue manager, about  
deploying 24

server-side checklist, troubleshooting 32

server-side common error messages 35

Siebel Wireless 7

SMQ subsystems, creating additional 27

SMQ User responsibility, about adding 21

SMQReceiver

deploying on UNIX systems 27

enabling 26

### Store-and-Forward Message History,

about 14

### Store-and-Forward Messaging Integration Objects, creating 21

Store-and-Forward User Message History,  
about 14

system administrator responsibilities 21

## T

timeout, Message Delivery Timeout

parameter 22

### troubleshooting

client-side checklist, troubleshooting 37

overview 32

server-side checklist 32

server-side common error messages 35

## U

UNIX systems, deploying SMQReceiver  
on 27

User Administration screen, about 39

## W

wireless network, about connecting  
using 31

### workflows

activating 24

Dispatch Board - Generic workflow 44

SAF Activity workflow 45

sample workflows, about 43

SMQ Delete FS Invoice Record workflow 48

SMQ Outbound Delete in Action Business  
Object workflow 48

SMQ PreDelete FS Invoice Record  
workflow 47

transaction delete workflows overview 46

transaction delete workflows, about 46

transaction delete workflows, configuring  
custom 49