



# **Siebel Store-and-Forward Messaging Guide for Handheld**

Siebel Innovation Pack 2013

Version 8.1/8.2

September 2013

**ORACLE®**

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

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

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

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

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

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

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

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

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

#### Documentation Accessibility

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

#### Access to Oracle Support

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

# Contents

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

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

- About Siebel Mobile Products 9
- Overview of Siebel Store-and-Forward Messaging Functionality 12
- Overview of Store-and-Forward Messaging for Handheld Architecture 13
  - Store-and-Forward Messaging Logical Architecture 14
  - Store-and-Forward Messaging Server Components 15
- Overview of Store-and-Forward Messaging for Siebel Handheld Client Modules 17
- About IBM WebSphere MQe 17
- Overview of Siebel Store-and-Forward Messaging Administration 17
- Overview of Store-and-Forward Messaging in the Handheld Applications 19
- Using Store-and-Forward Messaging with the Siebel Service Handheld Application 19
- Overview of Asynchronous Query on the Handheld Device 20
- Using Asynchronous Query with the Siebel Service Handheld Application 20

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

- Prerequisites for Siebel Store-and-Forward Messaging Installation 23
- Process of Installing Siebel Store-and-Forward Messaging 23
  - Enabling Component Groups for Store-and-Forward Messaging 24
  - Adding Responsibilities for Store-and-Forward Messaging 25
  - Entering Information for the Store-and-Forward Messaging Application 25
  - Creating Store-and-Forward Messaging Message Objects 26
  - Configuring Data Message Timeout Parameters for Store-and-Forward Messaging 27
  - Optional Step: Configuring Store-and-Forward Messaging Workflows 30
  - Activating Workflows and Reloading Runtime Events for Store-and-Forward Messaging 30
  - Testing for Internationalization 31

Process of Deploying IBM MQe Queue Managers	31
Installing the MQe Servlet	32
Enabling the SMQReceiver	34
Optional Step: Creating Additional SMQ Subsystems	35
Deploying SMQReceivers on UNIX Systems	35
Adjusting the Store-and-Forward Configuration Parameters	37
Setting Up Load Balancing for Store-and-Forward Messaging	38
About the Siebel Load Balancer	39
Process of Setting Up the Siebel Load Balancer	40
About Multiple Queue Managers	42
Setting Up Multiple Queue Managers	43
Process of Installing and Configuring Handheld Applications	44
Preparing for Installation of Siebel Handheld Applications	44
Installing Siebel Handheld Applications	46
Setting Up a Network Connection for Handheld Devices	46
Synchronizing and Registering the Handheld Application for Store-and-Forward Messaging	47
Optional Task: Tuning Handheld Application Parameters	47
Configuring a Business Object and Associated Business Components for Store-and-Forward Messaging	48
About Setting Up Third-Party Wireless Gateway Servers	49
Troubleshooting Store-and-Forward Messaging Setup and Configuration	49
Siebel Store-and-Forward Messaging Server-Side Troubleshooting Checklist	50
Siebel Store-and-Forward Messaging Server-Side Common Error Messages	52
Siebel Store-and-Forward Messaging Client-Side Troubleshooting Checklist	54
Siebel Store-and-Forward Messaging Client-Side Common Error Messages	56

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

Overview of Siebel Store-and-Forward Messaging Administration	59
Process of Administering Applications with Siebel Store-and-Forward Messaging	60
Enabling Applications for Store-and-Forward Messaging	60
Administering Messages in Store-and-Forward Messaging	60
Administering Handheld Devices for Store-and-Forward Messaging	61
Reviewing and Deleting Store-and-Forward Messaging History	62
Managing Outdated MQe Queues and Handheld Synchronization Extracts	67
Moving PDA Users to a Different MQe Queue Manager and Inbound Queue	67
Process of Setting Up Siebel Store-and-Forward Messaging on a Siebel Handheld Application	68

Preconfiguring the Siebel Handheld Application	69
Adding Store-and-Forward Messaging Views to the Handheld Application	70
Adding Alter Sound User Properties to the Siebel Handheld Application	70
Adding the Siebel Store-and-Forward Messaging Toolbar to the Handheld Application	71
Adding Application Find to the Handheld Application	71
Compiling Changes Made to the Handheld Application	72
Updating the User's Responsibilities and Views to Use Store-and-Forward Messaging	72
Adding Settings to Enable Store-and-Forward Messaging	73
Setting up the Siebel Handheld Application Setup.ini File for Store-and-Forward Messaging	74
Installing the Siebel Handheld Application	74
Process of Configuring Asynchronous Query for Siebel Handheld Applications	74
Configuring the Find Object in Siebel Tools	74
Creating the Asynchronous Query Results Applet	77
Creating the Asynchronous Query Results View	78
Adding Each New View to the Hidden Screens View	78
Adding Each New View and Responsibilities to the Application	79
Configuring Unsupported Business Components for Asynchronous Query	79
Example of Configuring an Inaccessible Applet	79

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

About Store-and-Forward Messaging Workflows	81
SMQ Outbound Activity to One Recipient	82
SMQ Outbound Activity to Primary Owned By	83
SMQ Outbound Activity to Team	84
SMQ SMS Outbound to one recipient	84
SMQ Delete Old Message History	85
SMQ Process User Queue	85
Transaction Delete Message Workflows	86
About the Transaction Delete Message Workflows	86
SMQ PreDelete FS Invoice Record	87
SMQ Delete FS Invoice Record	87
SMQ Outbound Delete in Action Business Object	88
Configuring Transaction Delete Workflows for Different Business Components and Business Objects	88

## **Chapter 6: Store-and-Forward Messaging for Handheld Business Services**

SMQ Message Service 93

SMQ Transport Service 94

### **Index**

# 1

## What's New in This Release

### What's New in Siebel Store-and-Forward Messaging Guide for Handheld, Version 8.1/8.2

Table 1 lists the changes described in this version of the documentation to support this release of the software. The new features described in Table 1 are available in Siebel CRM version 8.1.1.11, Siebel CRM version 8.2.2.4, and later.

Table 1. New Product Features in Siebel Store-and-Forward Messaging Guide for Handheld, Version 8.1/8.2

Topic	Description
<a href="#">"Optional Step: Configuring Store-and-Forward Messaging Workflows" on page 30</a>	Modified topic. Added a caution that all servers hosting workflows must have Java installed.
<a href="#">"Adjusting the Store-and-Forward Configuration Parameters" on page 37</a>	New topic. You must check the connection parameters for discrepancies and make any necessary adjustments.
<a href="#">"Adding an Exception to the PDA Connection Settings" on page 46</a>	New topic. Describes the procedure for adding an exception to the PDA connection settings.
<a href="#">"Managing Outdated MQe Queues and Handheld Synchronization Extracts" on page 67</a>	New topic. Contains instructions for requesting removal of outdated queues and handheld synchronization extracts.
<a href="#">"Moving PDA Users to a Different MQe Queue Manager and Inbound Queue" on page 67</a>	New topic. Describes the procedure for moving PDA users to a different MQe Queue Manager and queue.



# 2

## Overview of Siebel Store-and-Forward Messaging

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

- [“About Siebel Mobile Products” on page 9](#)
- [“Overview of Siebel Store-and-Forward Messaging Functionality” on page 12](#)
- [“Overview of Store-and-Forward Messaging for Handheld Architecture” on page 13](#)
- [“Overview of Store-and-Forward Messaging for Siebel Handheld Client Modules” on page 17](#)
- [“About IBM WebSphere MQe” on page 17](#)
- [“Overview of Siebel Store-and-Forward Messaging Administration” on page 17](#)
- [“Overview of Store-and-Forward Messaging in the Handheld Applications” on page 19](#)
- [“Using Store-and-Forward Messaging with the Siebel Service Handheld Application” on page 19](#)
- [“Overview of Asynchronous Query on the Handheld Device” on page 20](#)
- [“Using Asynchronous Query with the Siebel Service Handheld Application” on page 20](#)

### About Siebel Mobile Products

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

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

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

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

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

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

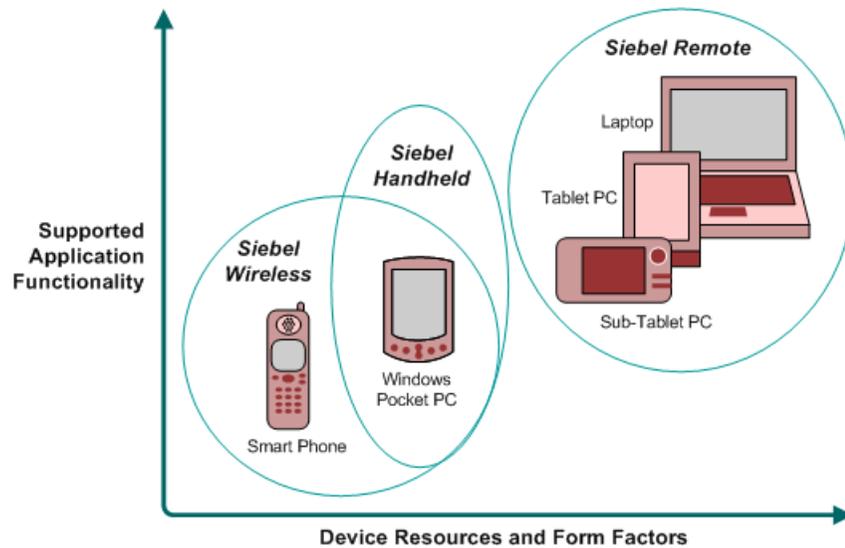


Figure 1. Siebel Mobile Product Platforms

Table 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>

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

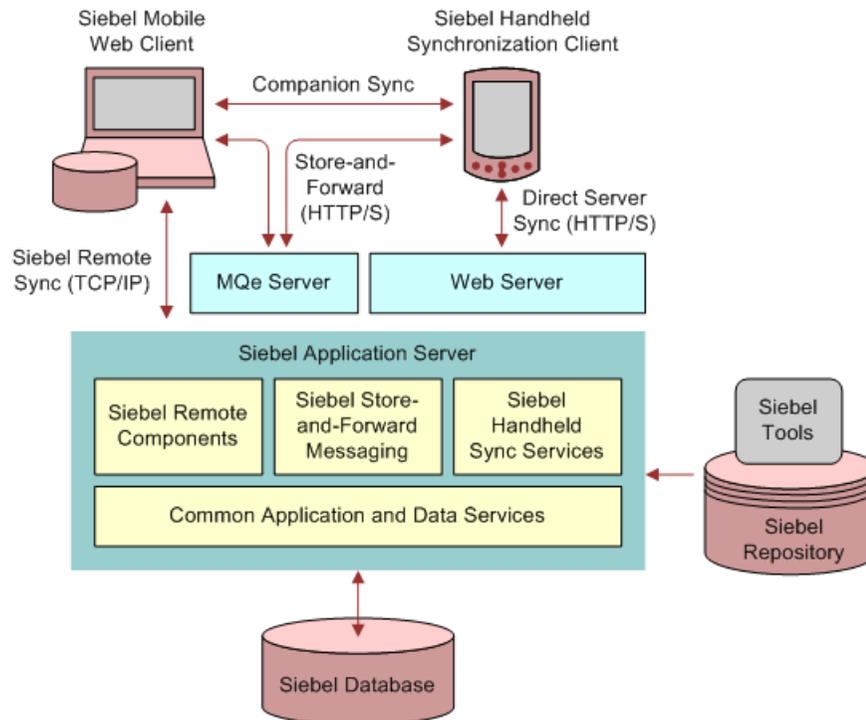


Figure 2. Synchronization Infrastructure for Siebel HandHeld Client and Siebel Mobile Web Client.

## Overview of Siebel Store-and-Forward Messaging Functionality

Siebel Store-and-Forward Messaging provides push-based information delivery between Siebel Business Applications and Siebel Handheld 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 with a Siebel Handheld application. The changes made by the mobile user in the Siebel Handheld application are delivered to the Siebel Server through Store-and-Forward Messaging.

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

Additionally, Siebel Store-and-Forward Messaging provides asynchronous query, which allows a Siebel Handheld application user to access data not available in the Siebel Handheld application's local database. For example, if a mobile user would like to access information on a particular part which is not available in his Service Handheld database, he can send a query to the Siebel Server. The query result is delivered to his Service Handheld application by way of Siebel Store-and-Forward Messaging.

Siebel Store-and-Forward Messaging is available as part of the Siebel Service Handheld application, and can be configured for other Siebel Handheld applications.

**NOTE:** Store-and-Forward Messaging is supported for Direct Server Synchronization mode, but not for Companion Synchronization.

For a list of supported handheld devices, see *System Requirements and Supported Platforms* on Oracle Technology Network.

## Overview of Store-and-Forward Messaging for Handheld Architecture

Siebel Store-and-Forward Messaging consists of server components, client modules, and the IBM MQe queuing system, which is responsible for queuing and delivering messages over a variety of wireless or wired networks.

## Store-and-Forward Messaging Logical Architecture

Figure 3 displays the logical architecture for Siebel Store-and-Forward Messaging.

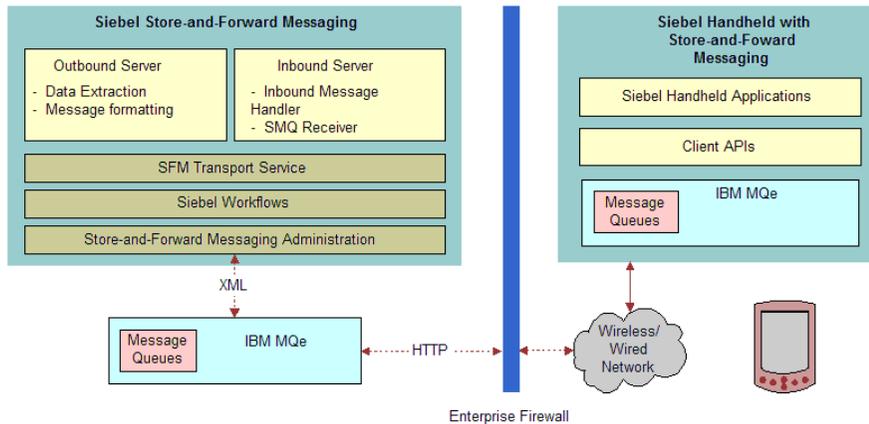


Figure 3. Store-and-Forward Messaging Logical Architecture

The Siebel Store-and-Forward Messaging Server includes the following components:

- An outbound server component to construct and send outbound messages to the server outbound queues
- An inbound server component to retrieve and process inbound messages
- A Transport service to deliver messages between the Siebel Server and server queues
- Siebel workflows to orchestrate the inbound and outbound server components
- Store-and-Forward Messaging administration to manage or administer the Store-and-Forward Messaging server

On the handheld client side, Store-and-Forward Messaging client modules are integrated with Siebel Handheld applications.

## Store-and-Forward Messaging Server Components

Figure 4 displays the server components for Siebel Store-and-Forward Messaging.

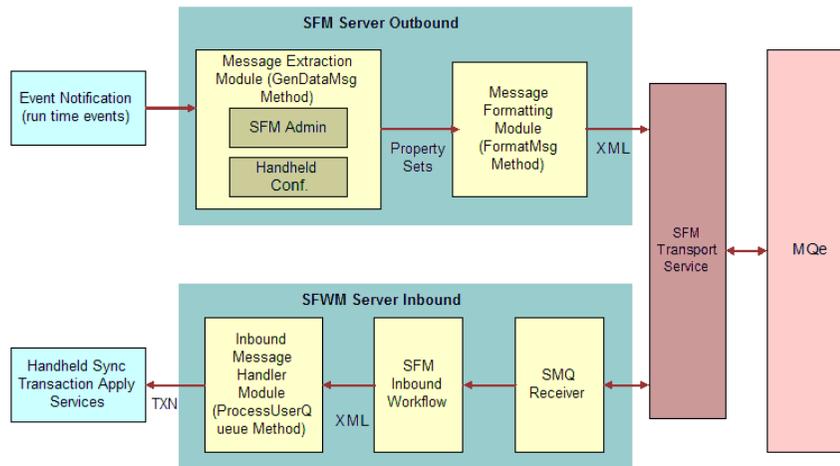


Figure 4. Siebel Store-and-Forward Messaging Server Components

A Siebel Server with Store-and-Forward Messaging enabled can support multiple Siebel Handheld applications. Store-and-Forward Messaging includes the following services and server components.

### Store-and-Forward Messaging Message Service (SMQ Message Service)

The Store-and-Forward Messaging Message Service is called SMQ Message Service. SMQ stands for Siebel Messaging and Queuing. The SMQ Message Service consists of the methods listed in Table 4.

Table 4. SMQ Message Service Methods

Method	Comments
GenDataMsg	Extracts data messages based on the Siebel Handheld application definition. For example, if the Action business component defined for a Siebel Service Handheld application includes activity status and activity description, the method extracts activity status and activity description data. The extracted data is delivered to a Service Handheld application by way of a data message.
FormatMsg	Converts Siebel data into well-formed XML data, or converts XML data to Siebel data.
ProcessUserQueue	Receives and processes server inbound messages. If the inbound message is a transaction-type message, the method calls the Siebel Handheld Synchronization engine.

## Store-and-Forward Messaging Transport Service (SMQ Transport Service)

The Store-and-Forward Messaging Transport Service (SMQ Transport Service) is a bidirectional transport service which interacts with the MQe queuing system. The transport service sends or receives messages to and from the server queues, provides data encryption or decryption, and performs data compression or decompression. Data is encrypted with the RC4 cipher and compressed with a suitable compression algorithm such as GZIP.

### The SMQ Receiver

The SMQ Receiver is a server component which polls the server inbound queues by way of the transport service. Server inbound queues consist of two-level queues, the main application queue and individual queues. For each Siebel Handheld application with Store-and-Forward Messaging enabled, there is one main application inbound queue; for each application user, there is one individual inbound queue.

SMQ Receiver dispatches messages from the main application queue into the individual user queues. It starts an inbound workflow task for each user to process the inbound messages in each individual queue. The inbound workflow calls the inbound message handler ProcessUserQueue method, which processes the message based on the message type. For example, if the message is a transaction message, the message is passed to the Handheld Sync Transaction Apply Services.

### Outbound Workflows

Siebel Store-and-Forward Messaging comes with the following sample workflows.

- **SMQ Outbound Activity to One Recipient.** Based on the Action business object. This workflow's main function is to generate a message, format the message, send the message to a server outbound queue, and update the message history table.
- **SMQ Outbound Activity \to Primary Owned By.** Uses SMQ Outbound Activity to One Recipient as a sub-workflow. This workflow is triggered by the WriteRecordUpdated runtime event in the Action business component. If an activity is assigned to a mobile user, or if an activity owned by a mobile user is updated, the workflow sends a data message to the mobile user with the updated activity and its associated data.
- **SMQ Outbound Activity to Team.** Uses SMQ Outbound Activity to One Recipient as a sub-workflow. The workflow allows you to send the same activity message to all service technicians on the same team.
- **SMQ Process User Queue.** Picks up server inbound messages from each individual inbound queue and processes them.
- **SMQ PreDelete FS Invoice Record.** Saves the information of a to-be-deleted record. When the PreDeleteRecord runtime event fires, it invokes this workflow.
- **SMQ Delete FS Invoice Record.** 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.

- **SMQ Outbound Delete in Action Business Object.** Generates a TXN Delete message and sends the message. When the SMQ Delete FS Invoice Record workflow runs, it invokes this workflow.

For more information about Store-and-Forward Messaging workflows, see [Chapter 5, “Siebel Store-and-Forward Messaging Sample Workflows.”](#) Additionally, you can examine these workflows with Siebel Tools. For more information, see *Siebel Business Process Framework: Workflow Guide*.

## Overview of Store-and-Forward Messaging for Siebel Handheld Client Modules

Store-and-Forward Messaging client modules include client inbound and outbound queues, client APIs, and a connectivity management module.

The connectivity management module periodically monitors the client connectivity and setup connections by way of the Pocket PC Connection Manager. You can, for example, set up a time interval that instructs the client module to check whether a connection exists each time the specified time interval passes. If, at the time of checking, the device is connected, the client module transfers messages (if any) in both directions and then shuts down. If the device is disconnected, the client module calls the Pocket PC Connection Manager to set up a connection. The connectivity module does not function if the handheld device or the wireless modem equipped with the handheld device does not use the Pocket PC Connection Manager to manage connectivity.

Messages are transmitted between the client inbound and outbound queues and the Store-and-Forward Messaging Server inbound and outbound queues. The client module is responsible for transmitting all messages including transactions, asynchronous query requests, and registration requests to the server. The client also processes server responses as well as server outbound data messages, such as Activity dispatches and updates. The MQe implementation requires that the client polls the server periodically to transmit messages. The interval at which polling occurs can be configured in the Siebel Handheld installation setup.ini file.

## About IBM WebSphere MQe

Siebel Store-and-Forward Messaging transmits messages to and from Siebel Handheld applications, 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 Siebel Handheld applications. Both the Siebel Server and Siebel Handheld applications communicate with the queue manager to exchange messages.

## Overview of Siebel Store-and-Forward Messaging Administration

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

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

- Message List
  - Store-and-Forward Applications
  - Store-and-Forward Message Timeout
- Store-and-Forward Message History
- User List
  - Device Administration
  - Store-and-Forward User Message History

## Store-and-Forward Device Administration View

The Device Administration view displays mobile users and their user device's name, mobile application, sync node ID, address type, and device address information. This view is populated for a user when the user synchronizes a Siebel Handheld application with the Siebel Server for the first time.

## Store-and-Forward Applications View

The Store-and-Forward Applications view displays how the data messages are constructed. The view allows you to review Message Objects, Message Components, and Fields. Based on the Siebel Handheld application definition, all the information is generated when a user synchronizes his Siebel Handheld application with the server for the first time. Not all the data messages listed in the Message Objects applet are enabled for Store-and-Forward Messaging.

By default, only the ActionSHCE data message is enabled for the Siebel Service Handheld application. The ActionSHCE message object corresponds to the ActionSHCE business object defined for Service Handheld application. Its Store-and-Forward Message Components map to the business components defined for the ActionSHCE business object. The fields for a particular Message Component map to the fields defined for the corresponding business component.

## Store-and-Forward Message Timeout View

The Store-and-Forward Message Timeout view allows you to specify message timeout parameters to make sure messages are delivered in a timely manner. For more information on these parameters, see [“Configuring Data Message Timeout Parameters for Store-and-Forward Messaging” on page 27](#).

## Store-and-Forward Message History Views

There are two message history views: Store-and-Forward User Message History and Store-and-Forward Message History. These views provide administrative information for each inbound or outbound message. The Store-and-Forward User Message History View displays the message information for a particular user, and the Store-and-Forward Message History View displays the message information for all users. These views allow you to view such characteristics of inbound or outbound messages as the unique message ID, the application that was used for the message, the message direction, the time a message was sent or received, the message size, the message status, and so on. From these views, you can track all the inbound and outbound messages for any user in your system and use this information to troubleshoot or monitor application or system usage.

# Overview of Store-and-Forward Messaging in the Handheld Applications

Siebel Store-and-Forward Messaging is set up for Service Handheld Application and can be configured for other Siebel Handheld applications. To set up Store-and-Forward Messaging for a Siebel Handheld application, see [“Process of Setting Up Siebel Store-and-Forward Messaging on a Siebel Handheld Application”](#) on page 68.

## Using Store-and-Forward Messaging with the Siebel Service Handheld Application

If a Siebel Service Handheld application is enabled with Store-and-Forward Messaging, several icons appear on the top-right corner of the screen, as shown in [Table 5](#).

Table 5. Store-and-Forward Messaging and Asynchronous Query icons

Icon	Comments
	<b>Inbound Message Icon.</b> When an inbound message arrives at the Siebel Handheld application, a beep sound sounds and the Inbound Message icon blinks. Tapping the icon stops the icon from blinking and opens the Wireless Alerts List view. This view displays the inbound messages received by the application in reverse chronological order.
	<b>Asynchronous Query Icon.</b> When asynchronous query results arrive at the Siebel Handheld application, the Asynchronous Query icon blinks. Tapping the icon opens the Async Query List view to display the returned results.  For more information about asynchronous query functionality, see <a href="#">“Overview of Asynchronous Query on the Handheld Device”</a> on page 20.
	<b>Outbound Message Icon.</b> When an outbound message is created and inserted into a user's outbound queue on his device, the icon blinks. The icon stops blinking when all the messages in the user's device outbound queue have been delivered to the server inbound queue.

For example, when the activity arrives at the Siebel Handheld application, a beep sounds and the Inbound Message icon flashes. The user taps the icon to open the Wireless Alerts List view. This view provides a list of alerts, indicating the inbound messages that have arrived at the device and the times when the messages were received. Closing the window navigates the user back to the previous screen or view. If the user updates an activity on the Service Handheld application, such as by changing the activity status, a message is created and inserted into the user's outbound queue on the device and the Outbound Message Icon flashes. When the outbound message is delivered to the server inbound queue, the Outbound Message Icon stops flashing.

**NOTE:** After a user performs a handheld synchronization, the application removes all the data in the Wireless Alerts List view.

## Overview of Asynchronous Query on the Handheld Device

Asynchronous Query allows a Siebel Handheld application user to access data not available in the Siebel Handheld application's local database. If a user cannot find desired information on a Handheld application's local data store, the user can send a query in a format of a request message to the target server. The server receives the request message, performs the query, and returns the query results in a data message to the requesting Siebel Handheld application. Later on, the data message arrives at the Siebel Handheld application and its information is inserted into a temporary table on the Siebel Handheld application. The user can then review the query results.

Asynchronous Query is set up for the Service Handheld application and can be configured to support other Siebel Handheld applications. To set up Asynchronous Query for a Siebel Handheld Application, see [“Process of Configuring Asynchronous Query for Siebel Handheld Applications” on page 74](#).

## Using Asynchronous Query with the Siebel Service Handheld Application

If a Service Handheld application is enabled with Asynchronous Query, you can perform asynchronous query for Products, Assets, Orders, and Solutions.

### *To perform an asynchronous query*

- 1 Navigate to the Find window by tapping the Edit menu, then selecting Find.
- 2 Tap the Look In field, and select a Find object from the drop-down list.  
Depending on the object you select, the window displays the relevant Find fields.
- 3 Enter the query criteria into the displayed Find fields.

- 4 Tap the Server Find button at the bottom of the window.

Your Siebel Handheld application sends a query request to the server by way of Store-and-Forward Messaging and displays a confirmation message.

**NOTE:** Tapping Find performs a query in the Siebel Handheld application's local database, and the operation returns the query results in the corresponding view. It does *not* retrieve asynchronous query results.

When the asynchronous query results arrive at your Siebel Handheld application, the Asynchronous Query icon flashes.

- 5 Tap the Asynchronous Query icon to open the Async Query List view, which displays the returned asynchronous query results. This view displays the columns listed in the following table.

Column	Comments
Search Specification	Search criteria you entered.
Object	Find object on which the query is based.
Sent Time	Time when the asynchronous query request message was sent to the server.
Received Time	Time when the query results arrived at the Siebel Handheld application.

- 6 Tap any button in the Async Query List view.

The buttons are described in the following table:

Button	Comments
Details	Opens the Async Query Results view, which displays the list of records meeting the asynchronous query criteria. The columns displayed in this view depend on the Find object on which the query is based.
Delete	Deletes the selected query record.
Delete All	Removes all the queries.
Refresh	If a new query result message arrives, tapping Refresh displays the new message.
OK	Closes the Asynchronous Query List view.

**NOTE:** After a user performs a handheld synchronization, all the data in the Async Query List and the Async Query Results views is removed.



# 3

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

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

- [“Prerequisites for Siebel Store-and-Forward Messaging Installation” on page 23](#)
- [“Process of Installing Siebel Store-and-Forward Messaging” on page 23](#)
- [“Process of Deploying IBM MQe Queue Managers” on page 31](#)
- [“Adjusting the Store-and-Forward Configuration Parameters” on page 37](#)
- [“Setting Up Load Balancing for Store-and-Forward Messaging” on page 38](#)
- [“Process of Installing and Configuring Handheld Applications” on page 44](#)
- [“Configuring a Business Object and Associated Business Components for Store-and-Forward Messaging” on page 48](#)
- [“About Setting Up Third-Party Wireless Gateway Servers” on page 49](#)
- [“Troubleshooting Store-and-Forward Messaging Setup and Configuration” on page 49](#)

### 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 24](#)
- [“Adding Responsibilities for Store-and-Forward Messaging” on page 25](#)
- [“Entering Information for the Store-and-Forward Messaging Application” on page 25](#)
- [“Creating Store-and-Forward Messaging Message Objects” on page 26](#)
- [“Configuring Data Message Timeout Parameters for Store-and-Forward Messaging” on page 27](#)
- [“Optional Step: Configuring Store-and-Forward Messaging Workflows” on page 30](#)
- [“Activating Workflows and Reloading Runtime Events for Store-and-Forward Messaging” on page 30](#)
- [“Testing for Internationalization” on page 31](#)

## Enabling Component Groups for Store-and-Forward Messaging

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

### *To enable component groups for Store-and-Forward Messaging*

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

Component Group	Comments
<b>Enterprise Application Integration (EAI)</b>	Used for Store-and-Forward Messaging server components such as SMQReceiver. Enable the entire component group.
<b>Field Service Object Manager or Field Service</b>	Used to enable Siebel Field Service related features For deployment of Store-and Forward Messaging, only the Field Service Object Manager component is needed from the Field Service component group, but you can enable the entire group if you wish.
<b>Handheld Synchronization</b>	Used for Siebel Handheld applications.
<b>Call Center Object Manager or Siebel Call Center</b>	Used to enable Siebel Call Center-related features. For deployment of Store-and-Forward Messaging, only the Call Center Object Manager component is needed from the Siebel Call Center component group, but you can enable the entire group if you wish.
<b>Sales Object Manager or Siebel Sales</b>	Used to enable Siebel Sales-related features. For deployment of Store-and-Forward Messaging, only the Sales Object Manager component is needed from the Siebel Sales component group, but you can enable the entire group if you wish.
<b>Workflow Management</b>	Used for Store-and-Forward Messaging workflows.

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

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

## Adding Responsibilities for Store-and-Forward Messaging

The responsibilities listed in [Table 6](#) are related to Store-and-Forward Messaging system administrator or users.

Table 6. Responsibilities for Store-and-Forward Messaging

Responsibility	Comments
Field Engineer - Handheld	Allows access to the Service Handheld application views. <b>NOTE:</b> You must add this responsibility to both the System Administrator (SADMIN) and the users of the Siebel Service Handheld application.
Field Technician - Handheld	Allows access to the Service Handheld application views. <b>NOTE:</b> You must add this responsibility to both the System Administrator (SADMIN) and the users of the Siebel Service Handheld application.
SMQ User	Allows access to the Store-and-Forward Messaging administration views and certain Siebel Handheld administration views. <b>NOTE:</b> It is important to verify that all Store-and-Forward Messaging administrators have the SMQ User responsibility.

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

## Entering Information for the Store-and-Forward Messaging Application

You must enter certain required Store-and-Forward application information in the Store-and-Forward Applications administration screen.

### *To enter Store-and-Forward Messaging application information*

- 1 Log into a Siebel Web Client application as a system administrator.
- 2 Navigate to Administration - Store-and-Forward > Store-and-Forward Applications.
- 3 Select the application for which you would like Store-and-Forward Messaging enabled.  
For example, Siebel Service for CE, which is the Siebel Service Handheld application.
- 4 Enter the Inbound Queue Name for the application.

This name should also be set in the Siebel Handheld application's registry during its installation. Typically, this value is InboundQ.

- 5 In the Queue Info field, enter the SMQReceiver Subsystem parameter.  
For example, SmqReceiverSubsys.
- 6 Save the record.

## Creating Store-and-Forward Messaging Message Objects

A Message Object is designed to transport all of the necessary data for a given object instance to the Siebel Handheld database. The object must reflect the definition of the underlying object in the Siebel Handheld configuration. To modify a message object, you must first modify the underlying Handheld objects and then perform a Generate Message Objects. In this way the message object is indirectly altered to ensure it stays in line with the Siebel Handheld application's definition of the business object.

You must populate message object and message component fields in the Store-and-Forward Applications view. Doing this allows you to review message objects, message components and fields for Siebel Handheld applications, or to set up Store-and-Forward Messaging timeout parameters for Siebel Handheld applications.

Observe the following requirements for the Handheld Synchronization Object Manager:

- All servers that host a Handheld Synchronization Object Manager must be based on the Windows operating system.
- Java is not required for the Handheld Synchronization Object Manager.

**NOTE:** This task can be completed by following the steps below, or if the Siebel Handheld application has been enabled to use Store-and-Forward Messaging, the first handheld synchronization session can populate the data also. To populate the data by means of the first Siebel Handheld synchronization session, the first user used for synchronization must have the responsibility to see all the views for the Siebel Handheld application. Otherwise, the message objects populated might be incomplete.

### *To manually create Store-and-Forward Messaging message objects*

- 1 Navigate to Administration—Store-and-Forward > Store-and-Forward Applications.
- 2 From the Store-and-Forward Applications drop-down menu, select Create Message Objects.

- 3 In the Store-and-Forward Generate Objects Popup applet, enter the following values:

In this field...	Enter this value...
HH Sync Component Name	The object manager name for the handheld sync component; for example, ServiceCEObjMgr_enu for the American English version of the Siebel Service Handheld application.  <b>NOTE:</b> If you are configuring a Siebel Industry application (such as Siebel Pharma Handheld) to use Store-and-Forward Messaging, enter SIAServiceCEObjMgr_enu.
HH Sync Server Name	The Siebel Server name where the handheld sync server component runs.

- 4 Click the Generate button.
- 5 After the generation process, the message objects, message components, and message fields are populated.

## Configuring Data Message Timeout Parameters 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 parameters:

- **Message Available Notification.** If a mobile device is not connected to a server and the server has outbound messages for delivery to the device, the server can send a SMS message. This parameter determines the maximum number of seconds the server will wait for a client to retrieve an available message before the server sends an SMS message to notify the mobile user to manually connect the device to the server.

SMS stands for “Short Message Service,” and is the common medium for short messages sent to phones and other mobile devices. This medium is available whether or not the device has a network connection. Consequently, the system can use it to wake up the device and have it connect automatically to get new data content.

For more information about message types, see [“About Message Types” on page 64](#).

**NOTE:** The SMQ Outbound Activity to One Recipient workflow includes an optional SMS notification step that is activated if the recipient mobile device does not receive an outbound message within the Message Available Notification timeout time. For more information about the SMQ Outbound Activity to One Recipient workflow, see [“SMQ Outbound Activity to One Recipient” on page 82](#).

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

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

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

- **Message Acceptance Timeout.** This parameter determines the amount of time, in seconds, within which a server outbound data message needs to be accepted or declined by the mobile user. This is an application-level parameter and can be used in a customized workflow or business process to define proper actions. The timeout length can be field-value dependent. For example, by setting values for activities, you can determine the amount of time, within which a service technician needs to tap Accept or Decline after a dispatched activity, received through Store-and-Forward Messaging, reaches his or her device.

**NOTE:** This parameter is currently not deployed by default in the Service Handheld application.

- **Start to Send.** This parameter determines a time window, in seconds, within which a newly created data message may be sent to a Siebel Handheld application. This is an application-level parameter. You can only associate this parameter with a time-based field, such as the Planned Start field for an activity. The value you set up for this parameter defines the delay of an outbound data message. For example, if the parameter is set to 36 hours, and if its associated field is Planned Start for an activity, then an outbound data message for a new activity will be sent to a handheld device when the activity's Planned Start time is within 36 hours of the current time. If the current time is already within the Start to Send window, then the message will be sent immediately.

Start to Send is deployed in the SMQ Outbound Activity to One Recipient workflow. For more information about this workflow, see [“SMQ Outbound Activity to One Recipient” on page 82](#).

The values set for all message timeout parameters are in seconds.

### *To set up outbound data message timeout parameters*

- 1 Navigate to Administration—Store-and-Forward > Store-and-Forward Message Timeout.
- 2 In the Store-and-Forward Applications screen, select the Store-and-Forward Messaging application.  
For example, Siebel Service for CE.
- 3 In the Message Timeout view, select the desired message object.  
For example, ActionSHCE, which is the data message object defined for the Action business object.
- 4 In the Store-and-Forward Timeout Administration sub-view, you can set up a timeout parameter and its timeout values by doing one or more of the following:

- Set the same Timeout value for every field and every field value of the triggering business record; for example, 600 seconds.

To set a timeout value for all fields and field values, go to [Step 5](#).

- Set Timeout values for values of a specific field of the triggering business record; for example, 600 seconds for a priority 1 Activity and 1200 seconds for a priority 2 Activity.

To set timeout values for selections in one field's list of values, go to [Step 6](#).

- 5 To set up a timeout value for all fields and field values, in the Store-and-Forward Timeout Administration sub-view, create a new record with values as shown in the following table.

Field	Value
Type	Select a timeout parameter.
Default Time	Specify the desired time frame; for example, 600. The value is in seconds. The default time value you set applies to all values of all fields.

- 6 To set up a timeout value for selections in a field's list of values, do the following:

- a In the Store-and-Forward Timeout Administration sub-view, create a new record with values as shown in the following table

Field	Value
Type	Select a timeout parameter.
Field	Select a field; for example, Priority.
Default Time	Specify the desired time frame; for example, 600. The value is in seconds. The default time value you set applies to all values of the specified field.

- b In the Field Values and Time Out subview, create a record for each field value that will need its own timeout specification.

For example, you can vary the timeout values for activity records based on the Priority field, as shown in the following table.

Field	Value	Time
Priority	1 - ASAP	600
Priority	2 - Important	1200

**NOTE:** These timeout specifications override the Default Time you entered in [Step a](#). The Default Time will apply to all field values for which you do *not* set timeout specifications here.

## Optional Step: Configuring Store-and-Forward Messaging Workflows

Siebel Store-and-Forward Messaging provides multiple preconfigured workflows. The typical preconfigured workflows are as follows:

- SMQ Process User Queue, an inbound workflow
- SMQ Outbound Activity to Primary Owned By, an outbound workflow
- SMQ Outbound Activity to One Recipient, an outbound workflow
- SMQ Outbound Activity to Team, an outbound workflow
- SMQ Delete Old Message History
- SMQ SMS Outbound to one recipient
- 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."](#)

**CAUTION:** Store-and-Forward Messaging-related workflows use the SMQ Message Service and SMQ Transport Service business services for communication with the MQe server. Both business services are based on Java. Consequently, all servers that host workflows must have Java installed. If a Store-and-Forward Messaging-related workflow runs on a server where Java is not installed, the call to the workflow business service generates an error and the operation fails.

## Activating Workflows and Reloading Runtime Events for Store-and-Forward Messaging

After the workflows are properly configured, you need to activate them and reload runtime events.

### *To activate workflows and reload runtime events*

- 1 Navigate to Administration - Business Process > Workflow Deployment.  
The Repository Workflow Processes screen appears.
- 2 In the Active Workflow Processes view, query for all the workflow processes beginning with SMQ. If any of the workflows described in ["Optional Step: Configuring Store-and-Forward Messaging Workflows"](#) on page 30 appear, delete them.

- 3 In the Repository Workflow Processes screen, query for all the workflow processes beginning with SMQ.  
The workflows described in [“Optional Step: Configuring Store-and-Forward Messaging Workflows” on page 30](#) are returned by this query.
- 4 Activate the Store-and-Forward Messaging workflows one by one by clicking the Activate button. The workflow “SMQ Outbound Activities to Primary Owned By” is triggered by the runtime events.
- 5 Reload the runtime events by navigating to Administration - Runtime Events > Events, and select Reload Runtime Events from the command menu.
- 6 After workflows are activated, associate the SMQ Process User Queue inbound workflow with the application.
  - a Navigate to Administration – Store-and-Forward > Message List.
  - b Select the application for which you have enabled Store-and-Forward Messaging.  
For example, Siebel Service for CE, which is the Siebel Service Handheld application.
  - c In the application record’s Inbound Workflow Name field, select SMQ Process User Queue.

## Testing for Internationalization

When creating an application for any language in the Store-and-Forward Applications View, make sure the QueueInfo column for each language has the same value. The SMQReceiver is language independent, and the default value, SmqReceiverSubsys, should remain the same for each application in each language you create.

# Process of Deploying IBM MQe Queue Managers

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

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

To deploy MQe Queue Managers, perform the following tasks:

- [“Installing the MQe Servlet” on page 32](#)
- [“Enabling the SMQReceiver” on page 34](#)
- [“Optional Step: Creating Additional SMQ Subsystems” on page 35](#)
- [“Deploying SMQReceivers on UNIX Systems” on page 35](#)

- “Setting Up Multiple Queue Managers” on page 43

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

## 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).

**NOTE:** The Web server that hosts the MQe Servlet can be based on either the Windows or Unix operating system.

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

### *To install the MQe Servlet*

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

**NOTE:** The Microsoft IIS Web server does not support servlets. However, third-party plug-ins are available that allow you to run servlets on IIS. Other Web servers such as Tomcat include servlet support by default.
  - Each computer running an MQe Queue Manager must have a Java Runtime Environment (JRE) installed. For information about the required Java Runtime Environment (JRE) version, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.
  - The servlet uses your Web server’s existing port (generally port 80, the default for HTTP). Your firewall must permit inbound connections on this port for mobile clients or regular Internet users.

- 2 Locate the MQe servlet on your installation media.

The servlet is delivered as part of the Siebel Server installation package, in the <Siebel\_Server>\classes\SiebelSMQServlet.war file. You can open .war files with winzip or any extraction tool that can decompress zip files.

- 3 Follow the procedure specific to your Web server or servlet engine for installing the MQe servlet.

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

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

```
<root>\
  WEB-INF\
    web.xml
    lib\
      MQeBase.jar
      MQeSecurity.jar
      SiebelSMQ.jar
```

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

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

- 4 Make a backup copy of the Web server's web.xml file, then open the web.xml file in a text editor and complete the following steps to change the DataDir parameter to specify the directory where you want the queues to be stored:
  - a Locate the following portion of the file:

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

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

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

- c Save and close the web.xml file.

- 5 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.

Observe the following requirements for the SMQReceiver:

- The server hosting the SMQReceiver can be based on either the Windows or Unix operating system.
- Java must be installed on all servers that host a SMQReceiver

### 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:

```
srvrMgr /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  
ReceiverConnecti onSubsystem=SmqRecei verSubsys,  
ReceiverDataHandl ingSubsystem=SmqDi spatchSubsys,  
ReceiverMethodName=Recei veDi spatch
```

Input parameters for the SMQReceiver are defined in the named subsystems. The first named subsystem defines the parameters necessary to configure an SMQReceiver for receiving messages for a single application (see the predefined named subsystem, SmqReceiverSubsys). The second named subsystem identifies the workflow for processing the received messages (see the predefined named subsystem, SmqDispatchSubsys).

- 3 From the server manager, run the following command once to make sure that the SMQReceiver starts itself every time the Siebel Server is restarted:

change param Dfl tTasks=1 for component SMQReceiver

When the SMQReceiver is running, there is exactly one log file SMQReceiver\_XXX.log in the Siebel server log directory.

## Optional Step: Creating Additional SMQ Subsystems

This step is required if you want to run Siebel Store-and-Forward Messaging for more than one Siebel Handheld application. Each Siebel Handheld application should have its own subsystems. This step allows each application to receive messages in its own queue, independent of all other applications. The following steps illustrate how to create additional subsystems for an application that uses an inbound queue named InboundFSQ and a workflow named SMQ Process FS User Queue.

### To create additional SMQ subsystems

- From the server manager, execute the following two commands:

```
create named subsystem SmqFSReceiverSubsys for subsystem SMQRcvSubsys with  
SmqQueueType=MQe, SmqQueueName=InboundFSQ, SmqQueueManagerName=ServerQM, SmqServerPortNumber=8082,  
SmqTransportServiceName="SMQ Transport Service", SmqTransportMethodName=WaitForMessage
```

and

```
create named subsystem SmqFSDispatchSubsys for subsystem EAITransportDataHandlingSubsys with  
DispatchWorkflowProcess="SMQ Process FS User Queue"
```

## Deploying SMQReceivers on UNIX Systems

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

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

### To install SMQReceivers on UNIX, Solaris platform

- 1 Make sure Java is installed on the following servers:

- Tomcat/MQe server
- SMQReceiver

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

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

```
setenv LD_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.

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

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

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

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

- 1 Set the following environment variables:

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

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

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

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

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

```
create named subsystem SmqJVMSubsys for subsystem JVMSubsys

change param CLASSPATH="
<Instal I_Dir>/siebsvr/classes/SiebelJI_Common.jar: <Instal I_Dir>/siebsvr/classes/
SiebelJI.jar: <Instal I_Dir>/siebsvr/classes/SiebelJI_psl.jar: <Instal I_Dir>/
siebsvr/classes/SiebelJI_en.jar: <Instal I_Dir>/siebsvr/classes/Siebel.jar" for
named subsystem SmqJVMSubsys
```

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

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

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

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

- 1 The EAI component group is needed to run SMQReceiver.
- 2 Make sure the directory that contains libjvm.so and its parent directory are in LD\_LIBRARY\_PATH.

If necessary, add these directories to the LD\_LIBRARY\_PATH environmental variable.

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

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

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

```
change param CLASSPATH="{ $SI EBEL_HOME }/si ebsrvr/cl asses/  
SiebelJI_enu.jar: { $SI EBEL_HOME }/si ebsrvr/cl asses/Siebel.jar" for named subsystem  
SmqJVMSubsys
```

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

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

## Adjusting the Store-and-Forward Configuration Parameters

The default connection parameters for the Store-and-Forward Messaging components might not always match. After deploying the IBM MQe Queue Managers, you must check these connection parameters for discrepancies and make any necessary adjustments. For more information, see ["Process of Deploying IBM MQe Queue Managers" on page 31](#).

### To adjust the Store-and-Forward configuration parameters

- 1 Display the connection parameters from the MQe Server servlet by opening a Web browser and entering the following URL:

```
http://server name:port number/SiebelSMQServlet
```

If the Web browser runs on same computer as the Web server of the MQe Server servlet, then *server name* can be localhost. Commonly used *port number* values include 8080 and 8082.

Output text similar to the following example appears:

```
----- begin output from Siebel SMQServlet -----  
Queue manager servlet ServerQM started Fri Dec 03 15:50:54 CET 2010  
Using IBM WebSphere MQ Everyplace version 2.0.2.4  
Queue storage at C:\sia81\si ebsrvr\MQe\ServerQM
```

Client connection settings should be:

```
QueuingSoftware = MQe  
MQeServerAddr = <client-visible address of this host>  
MQeServerPort = 8080  
MQeServerCmd = /SiebelSMQServlet (optional: for logging add "?log=1&client=<NAME>")  
MQeServerQMgr = ServerQM  
MQeServerQueue = <see your administrator>
```

Server connection settings should be:

```
SmqQueueType = MQe
SmqServerMachineName = <server-visible address of this host>
SmqServerPortNumber = 8080
SmqServletPathName = /SiebelSMQServlet
SmqQueueManagerName = ServerQM
SmqQueueName = <see your administrator>
----- end of output from Siebel SMQServlet -----
```

The default value of the SmqQueueName server connection setting is InboundQ.

- 2 Locate and open the setup.ini file for the Handheld application.

For more information about the Handheld application setup.ini file, see [“Preparing for Installation of Siebel Handheld Applications” on page 44](#).

- 3 Compare the client connection settings displayed by [Step 1](#) with the values in the setup.ini file, and correct any discrepancies in the setup.ini file.

Discrepancies might, for example, be found in port numbers and in the server command (/smq as opposed to /SiebelSMQServlet).

- 4 Display the server connection parameter values in SMQReceiverSubsystem by starting the server manager and entering the following command:

```
svrmgr> list parameters for named subsystem SmqReceiverSubsys
```

- 5 Compare the server connection settings displayed by [Step 1](#) with the values in SMQReceiverSubsystem, and correct any discrepancies in SMQReceiverSubsystem.

Example corrections are as follows:

```
svrmgr> change param SmqServletPathName=/SiebelSMQServlet for named subsystem
SmqReceiverSubsys
```

```
svrmgr> change param SmqServerPortNumber=8080 for named subsystem
SmqReceiverSubsys
```

```
svrmgr> change param SmqServletUse=True for named subsystem SmqReceiverSubsys
```

**NOTE:** If the SmqServletUse parameter value is False, you must set it to True.

## Setting Up Load Balancing for Store-and-Forward Messaging

Siebel Store-and-Forward messaging requests service from the Handheld Sync server components. To prevent a Handheld Sync server component from becoming a bottleneck, you can use a load balancer to process messages on multiple Handheld Sync server components in different Siebel servers. For this purpose, you can deploy either the Siebel Load Balancer or a third-party load balancer.

Store-and-Forward Messaging also allows you to deploy multiple queue managers for individual remote devices or machines. Siebel Load Balancer and Multiple Queue Managers are separate features. Both of these features are optional and can be used individually or together.

This section describes the deployment and use of the Siebel Load Balancer. For information about deploying a third-party load balancer, see 477835.1 (Doc ID) on My Oracle Support. This document was previously published as Siebel Technical Note 540.

This section includes the following topics:

- [“About the Siebel Load Balancer” on page 39](#)
- [“Process of Setting Up the Siebel Load Balancer” on page 40](#)
- [“About Multiple Queue Managers” on page 42](#)
- [“Setting Up Multiple Queue Managers” on page 43](#)

## About the Siebel Load Balancer

Siebel Store-and-Forward Messaging provides a load balancing framework that allows messages to be received, not only by the server on which the handheld application is registered, but other servers as well. With load balancing deployed, Store-and-Forward Messaging can use any available server where the Handheld Application Server is running. During inbound user queue processing, if an error occurs, the message will not be removed from the queue but will be retried.

The Siebel Load Balancer works as follows:

- The Siebel Session Manager reads the content of a virtual hosts file and dispatches client requests to the servers defined in that file in a round-robin fashion. For example, if there are three Siebel Servers, A, B, and C, the requests are processed one by one by servers A, B, then C repeatedly.
- If SiebelServA is down, the system retries a connection request sent to SiebelServA automatically. A retry occurs when the system is trying to establish a connection. If a connection is established and broken in the middle of a session, no retry is attempted.
- If the session manager attempts to connect to SiebelServA and SiebelServA is unavailable, the session manager first retries connecting to SiebelServA. If it cannot establish a connection on this retry, the next retry goes to SiebelServB. If the session manager cannot establish a connection with SiebelServB, subsequent retries go to A, B, A, B, and so on, until a connection to either server can be established.
- There is no guarantee that the Session Manager will always succeed in establishing a connection with the server through any given number of retries. If the Session Manager does not succeed, it enters an error message in the log file. For more information about retries, see 477835.1 (Doc ID) on My Oracle Support. This document was previously published as Siebel Technical Note 540.
- If SiebelServA is up again, the Session Manager detects this and resumes routing requests to that server.

In planning for load balancing, it is recommended that you first decide how many server machines will be used for Load Balancing, then install the required Siebel server components on the server machines, and finally configure the load balancing setup before any Siebel Handheld connects.

## Process of Setting Up the Siebel Load Balancer

The process for deploying the Siebel Load Balancer is as follows:

- 1 "Setting Up the Virtual Hosts File" on page 40
- 2 "Pointing the Workflow Process Manager to the Virtual Hosts File" on page 41
- 3 "Turning On Load Balancing" on page 41

### Setting Up the Virtual Hosts File

To work with Siebel Load Balancer, every machine that can serve as a caller connecting to a virtual host (that is, a machine hosting the Workflow Process Manager component) must have a VirtualHostsFile parameter pointing to a file location on that machine.

The Siebel Session Manager reads the content of a virtual hosts file and dispatches client requests to the servers defined in that file in a round-robin fashion. To deploy the Siebel Load Balancer, you first generate a virtual hosts file.

#### To set up a virtual hosts file

- 1 Connect to the Siebel Server Manager and run the following command:

```
generate lbconfig
```

This creates the lbconfig.txt file in the SiebSrvr\ADMIN directory. For more information about the lbconfig file and the Siebel Server Manager, see *Siebel System Administration Guide*.

- 2 Open a command prompt, navigate to eappweb\bin, and open the file eapps.cfg file.

**NOTE:** If you are enabling load balancing for a Siebel Industry application, open the eapps\_sia.cfg file.

- 3 In the eapps.cfg file, make the following changes:

- a In the [ConnMgmt] section, enter the following settings:

```
EnableVirtualHosts = true
```

```
VirtualHostsFile = <Siebel_Server>\ADMIN\lbconfig.txt
```

- b In the [/servicece\_enu] section, enter the following settings:

**NOTE:** If you are enabling load balancing for a Siebel Industry application, go to the [/siaservicece\_enu] section instead.

```
ConnectionString = siebel.TCPIP.None.None://VirtualServer/Siebel/SCCObjMgr_enu  
(instead of "Siebel Server: SCBroker Port" changed to "Virtual Server")
```

```
WebPublicRootDir = <eappweb>\public\enu
```

```
WebUpdatePassword = OMjqhm75+uQBEBEDw==
```

- 4 Save and close the file.
- 5 Stop and restart the IIS server.

## Pointing the Workflow Process Manger to the Virtual Hosts File

To use the Siebel Load Balancer for Handheld Sync Server components, you must enable load balancing in the Workflow Process Manger (WfProcMgr).

The virtual hosts file (lbconfig.cfg) has information about which Handheld Sync Servers together will serve as the same virtual host. To process SMQ messages, the Workflow Process Manger must send requests to a Handheld Sync Server. To use the Siebel Load Balancer, the Workflow Process Manger must have a VirtualHostsFile parameter that specifies the full pathname of the virtual hosts file. The Workflow Process Manger can then open the virtual hosts file and dispatch the request to the Handheld Sync Servers listed therein.

Observe the following requirements for the Workflow Process Manager:

- All servers hosting a Workflow Process Manager must be based on the Windows operating system.
- In a mixed operating system environment, every Workflow Process Manager must run on a Windows-based server only. Store-and-Forward Messaging does not work correctly when one or more of the Workflow Process Managers run on Unix-based servers.
- Java must be installed on every server that hosts a Workflow Process Manager.

### *To point the server to the virtual hosts file*

- 1 Navigate to Administration - Server Configuration > Servers.
- 2 Select the Siebel Server for which you want to enable load balancing.
- 3 Click the Components tab below the Siebel Servers list.
- 4 In the Components list, locate and select Workflow Process Manager.
- 5 Click the Parameters tab below the Components list.
- 6 In the Parameters list, enter values for the parameters that are listed in the following table.

Parameter	Value on Restart
Enable internal load balancing (Alias: EnableVirtualHosts)	True
Session manager load balancing configuration file (Alias: VirtualHostsFile)	Path and filename of the virtual hosts file; for example, d:\sea80\admin\lbconfig.txt

When you restart the server, the values you entered for these parameters in their Value on Restart fields will appear in their Value fields as well.

## Turning On Load Balancing

You activate load balancing by creating a setting that specifies the Virtual Server as the Load Balancer. The setting is optional and is configurable for each application.

- If you specify the setting, the Store-and-Forward message service provides the VirtualServer as the Load Balancer string when invoking the Sync Agent's Connect method.
- If you do *not* specify the setting, Store-and-Forward Messaging connects to the Handheld Sync server component as set in the Mobile Device table. The Load Balancer is not used.

To activate load balancing, follow this procedure.

### *To turn on load balancing*

- 1 Navigate to Administration - Mobile > Application Administration > Settings.
- 2 In the Settings list, create a new record. The following table displays an example.

Field	Value
Name	SFMVirtualHost
Value	VirtualServer

These values must match the values specified in the lbconfig.txt file. See [“Setting Up the Virtual Hosts File” on page 40](#).

## About 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. The SMQReceiver dispatches the messages in the InboundQ to each user's inbound queue and starts a task to process inbound messages in any Workflow Process Manager server component. The queues for each user are managed by their respective Queue Managers.

The architecture of multiple queue managers is illustrated by Figure 5.

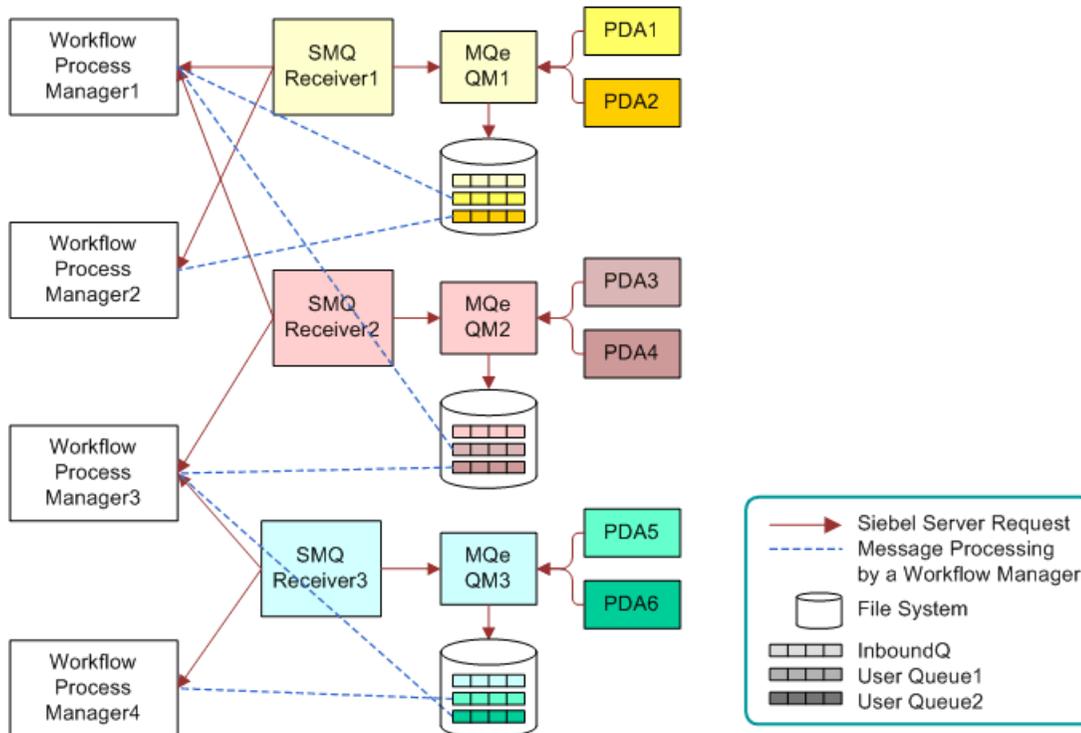


Figure 5. Multiple Queue Managers Architecture

## Setting Up Multiple Queue Managers

If you use multiple Queue Managers, then you must create multiple subsystems.

Your Siebel Handheld application comes with one default subsystem called SmqReceiverSubsys. 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 35. For information about specifying a subsystem for an SMQ Receiver, see [“Enabling the SMQReceiver”](#) on page 34.

# Process of Installing and Configuring Handheld Applications

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

To install and configure the Siebel Handheld applications, perform the following tasks:

- [“Preparing for Installation of Siebel Handheld Applications” on page 44](#)
- [“Installing Siebel Handheld Applications” on page 46](#)
- [“Setting Up a Network Connection for Handheld Devices” on page 46](#)
- [“Synchronizing and Registering the Handheld Application for Store-and-Forward Messaging” on page 47](#)
- [“Optional Task: Tuning Handheld Application Parameters” on page 47](#)

## Preparing for Installation of Siebel Handheld Applications

Prior to installing a Siebel Handheld application on a handheld device, you must first edit and verify the setup.ini parameters. Following this step, you may then install the Siebel Handheld application on the handheld device. See the Siebel Handheld Guide for the Siebel Handheld application you are using.

You must partition the user Handheld devices in advance and assign them to specific Queue Managers. It is recommended that you duplicate the Siebel Handheld installation packages in separate directories and modify the setup.ini files to point to different Queue managers in advance.

**CAUTION:** You statically assign users to Queue Managers. The relationship, once made, is constant. Therefore, plan carefully when deploying devices to make sure that end users are as evenly spread as possible across the available Queue Managers, both in terms of the number of devices and in terms of expected load (allowing for other attributes such as time zone).

**NOTE:** Store-and-Forward Messaging is supported for Direct Server Synchronization mode, but not for Companion Synchronization.

### *To verify and edit the setup.ini parameters*

- 1 Open the application’s setup.ini file.

For information on working with the setup.ini file, see the Siebel Handheld Guide for the Siebel Handheld application you are using.

- 2 Change or verify the following parameter:

- SyncURL = http://[server IP address]/servicece\_enu/

The SyncURL parameter is required to specify the URL to perform Siebel Handheld synchronization. This value is saved in registry on device. You can find this parameter in the [Siebel] section of the setup.ini file. The server IP address must be specified with the Web server's IP address.

**3** Change or verify the following parameter values in the [Wireless Messaging] section.

- QueuingSoftware = MQE

QueuingSoftware can be MQE or NONE (if NONE is used, the Wireless Messaging feature is disabled)

- SMQCompression = GZIP

[Optional parameter] SMQ Compression setting. The valid values could be GZIP or blank. If this parameter is blank, no compression is applied.

- SMQEncryption = RC4

[Optional parameter] SMQ Encryption setting. The valid values could be RC4 or blank. If blank, no encryption is applied.

If MQE is selected for QueuingSoftware, the following parameters need to be specified. If you are implementing load balancing, you must assign individual handheld devices to different Queue Managers. When a device sends messages to the Siebel server, the messages arrives in its pre-assigned Queue Manager's InboundQ first. For more information, see ["About the Siebel Load Balancer" on page 39](#).

- MQeServerQMgr = ServerQM

Name of the MQe server Queue Manager to which you are assigning this handheld device. The default value is ServerQM. It should be the same as the subsystem parameter "SmqQueueManagerName" in the subsystem specified in the Queue Info field in Store-and-Forward Applications View.

- MQeServerQueue = InboundQ

MQe server Queue name. The default value is InboundQ. It should be the same as the subsystem parameter "SmqQueueName" in the subsystem specified in the Queue Info field in Store-and-Forward Applications View.

- MQeServerAddr = [MQe Server Name]

Address of the MQe server hosting the Queue Manager to which you are assigning this handheld device. The server domain name or IP address of the machine running the Queue Manager. Make sure the domain name or IP address is resolvable from the handheld devices in the deployed environment.

- MQeServerPort = 8080

Number of the port to which the assigned Queue Manager is listening. The default value is 8082. It should be the same as the subsystem parameter "SmqServerPortNumber" in the subsystem specified in the Queue Info field in Store-and-Forward Application View.

## Installing Siebel Handheld Applications

Install the Siebel Handheld applications by following the instructions contained in the Siebel Handheld Guide for the Siebel Handheld application you are using.

## Setting Up a Network Connection for Handheld Devices

After installing the Siebel Handheld application on the handheld device, you must set up a network connection for the handheld device so that the device can communicate with the queue manager by way of an IP network through the HTTP protocol.

**NOTE:** If you plan to connect the handheld device to the queue manager through a wireless network, such as a wireless LAN 802.11 network or a wireless WAN network (for example, a GPRS or CDMA network), or a VPN, follow the instructions provided by the device manufacturer, the wireless modem manufacturer or the wireless carrier.

### Adding a Network to the PDA Connection Settings

This procedure applies when the queue manager is configured on a computer in your company network. If you plan to use Microsoft ActiveSync to connect your device to a laptop and then to the company trusted network, or if you plan to connect the device to the company network directly, then follow these steps.

#### *To add a network to the PDA connection settings*

- 1 Tap Start, Settings, then tap the Connections tab.
- 2 Tap the Connections icon, then tap the Advanced tab.
- 3 Tap the Select Networks button.  
The Network Management screen appears.
- 4 Select My Work Network from the first drop-down list, and tap OK.
- 5 Tap the Network Card button.
- 6 Make sure the top item is set to Work.

### Adding an Exception to the PDA Connection Settings

If the MQeServerAddr contains points (for example, 10.11.12.13 or sfm.mycomp.com), then you must add exceptions to the PDA connection settings.

#### *To add an exception to the PDA connection settings*

- 1 Tap Start, Settings, then tap the Connections tab.
- 2 Tap the Connections icon, then tap the Advanced tab.
- 3 Tap the Exceptions button.

- 4 Add the appropriate exception.

For example, enter 10.\*/\* or \*.mycomp.\*/\*

## Synchronizing and Registering the Handheld Application for Store-and-Forward Messaging

If you use ActiveSync to connect your device to the Siebel Server to perform handheld synchronization, then after the synchronization is completed, and before you tap OK, it is recommended that you reconnect the handheld device to set up a new ActiveSync session. This is because the initial registration and initialization step sometimes fails unless a new ActiveSync session is launched. For more information about handheld synchronization, see the Siebel Handheld Guide for the Siebel Handheld application you are using.

After handheld synchronization, the Siebel Handheld application automatically starts and registers with the Siebel Server through Store-and-Forward Messaging. The registration requires that the Siebel Handheld application performs a successful initialization step (INIT) with the Siebel Server Handheld Sync component.

### Optional Task: Tuning Handheld Application Parameters

You can find the following parameters in the device registry. They are used by the Siebel Handheld application and can be modified. Siebel Client Registry settings are set during the client install. Some of their values can be configured by modifying the setup.ini before installation.

**NOTE:** Modifying the registry settings for Siebel Handheld after installation is not recommended. It is recommended that you modify the setup.ini parameters prior to installation instead.

Table 7 describes the Handheld application parameters.

Table 7. Handheld Application Parameters

Parameter	Description
MaxReturnedRecords	(Optional) The maximum number of records returned by an asynchronous query. The default value is 500 records. You may increase or decrease the value by adding the name/value pair through device registry settings. This value is used by SMQ server to set the search result limit. If the number of search result is more than the setting, no records are returned.
SMQPollingInterval	(Optional) Time interval, in seconds, that a Siebel Handheld application contacts a server queue manager to check for messages. The default value is 30 seconds. You may increase or decrease the value by adding this parameter in setup.ini or adding the name/value pair through device registry settings. If you decrease the value, it causes the application to contact the server more often. It means you may get messages faster, but application performance could degrade for non-messaging related operations.  <b>CAUTION:</b> Change this value only when it is absolutely needed.
SMQRegisterTimeOut	(Optional) Timeout for registration acknowledgement. If the application registration does not happen during the specified time frame, a message box would pop up asking if the user would like to wait longer. The default value is 30 seconds. You can change the default value by adding this parameter in setup.ini or adding the name/value pair through device registry settings. When you are working on a very slow network and find a lot of message boxes popping up before a successful registration, it is recommended you increase the value. Otherwise, you should avoid setting this optional parameter.

## Configuring a Business Object and Associated Business Components for Store-and-Forward Messaging

Before you start to configure the Business Object (BO) and its Business Components (BCs) for Store-and-Forward Messaging, make sure that:

- The BO and associated BCs are defined in the corresponding Siebel Handheld application definition
- The Siebel Handheld application supports the BO and BCs

If you plan to modify the sample workflows for the configuration, in the “[SMQ Outbound Activity to One Recipient](#)”, change the following input arguments in Generate Data Message [Step 3 on page 82](#)

- Change MessageObject to the name of the desired Business Object

- Change BusComp to the name of business component to be extracted

[Step 3 on page 82](#) is used to extract data messages based on the Siebel Handheld application definition.

For a Store-and-Forward Messaging enabled Siebel Handheld application, you can find its Message Objects from Site Map > Administration - Store-and-Forward > Store-and-Forward Applications. In addition, make sure the input parameter Primary Row Id of the GenDataMsg() method to be Row Id of the Primary Business Component for the BO.

Alerts are displayed in the Wireless Alerts List View of the corresponding Siebel Handheld application to indicate:

- That inbound messages have arrived at the device
- The time when the messages were received.

To customize the alert text information to reflect the configured BO, change the input argument AlertText in the Generate Data Message [Step 3 on page 82](#).

The “[SMQ Outbound Activity to One Recipient](#)” takes the input process property Recipient. The value of this property is used to determine message recipients. Currently the property value is set as the field Primary Owned By of the Action BC by the “[SMQ Outbound Activity to Primary Owned By](#)” workflow. This property Recipient is configurable for other BO or BC to determine the message recipients.

No change is required for the inbound workflow, “[SMQ Process User Queue](#)”.

## About Setting Up Third-Party Wireless Gateway Servers

IBM WebSphere Everyplace Connection Manager 5.0 has been certified with Siebel Store-and-Forward Messaging. IBM WebSphere Everyplace Connection Manager provides the following features and flexibility to deploy Siebel Store-and-Forward Messaging.

- Additional security over wireless or wired networks
- Seamless cross-network roaming
- Reduced data transmission over wireless networks

For more information, see the related documents provided by IBM.

## 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 Handheld application. Troubleshooting information provided here includes the following topics:

- “[Siebel Store-and-Forward Messaging Server-Side Troubleshooting Checklist](#)” on page 50

- [“Siebel Store-and-Forward Messaging Server-Side Common Error Messages” on page 52](#)
- [“Siebel Store-and-Forward Messaging Client-Side Troubleshooting Checklist” on page 54](#)
- [“Siebel Store-and-Forward Messaging Client-Side Common Error Messages” on page 56](#)

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

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

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

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

```
list task for comp SMQReceiver
```

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

- 4 Make sure the queue directory tree on the queue manager host computer is created and the Siebel Handheld application has registered.

**NOTE:** In the example below, <USERNAME> stands for the user name of the Handheld application user. <ddd> stands for any three digits.

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

```
<DataDir>\MQe\ServerQM\Queues\ServerQM
  AdminQ
  AdminReplyQ
  InboundQ
  [InboundQ.<USERNAME>.<ddd>] // created when the application registers
```

[<USERNAME>. <ddd>.SFQ] // created when the Siebel server responds to the registration

When a Siebel Handheld application starts the first time, the application sends a registration message to the Siebel Server. The registration message first reaches the queue manager's InboundQ directory. The SMQReceiver moves the registration message to the InboundQ.UserQueueName directory, for example, InboundQ.<USERNAME>.<ddd>. Here, <USERNAME> stands for the mobile user name. The registration message is then processed by the Workflow Process Manager and a server response message is generated and placed into the user's outbound queue, for example, <USERNAME>.<ddd>.SFQ. You can verify whether the two directories have been created for a user and whether there are messages over there.

You can check application registration from the Administration—Store-and-Forward screen. Go to User List > Device Administration, query for the user and verify whether a device name shows up in the applet. After a proper registration, the device address column in the Store-and-Forward Device Addresses view shows the device queue name for the particular user.

- 5 Check messages sent from or received on 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 received from or sent to the Siebel Handheld application. See [“To review all message history records” on page 62](#).
- 6 Check that the message objects are populated in the Siebel Store-and-Forward Application View. If they are not populated:
  - a Navigate to Administration – Mobile.
  - b Click the Application Administration view tab.
  - c Select the application name. For example, Siebel Service for CE.
  - d Click the Setting view tab, perform the following query: SchemaChangeNotification in the name field and see if it exists and if the value is: SMQ Message Service. If not, add it and remove the hhsync directory under the siebsrvr directory.
  - e Restart the Siebel server and try again.

If message objects are populated, it may be due to some server components not running.

If no, check the server-side log file to see if there are any error messages.

- 7 Examine the log files such as SMQReceiver log files, WfProcMgr log files, or the Siebel Handheld application 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 Workflow Process Manager,
  - change evtloglvl SMQMsgSvc=5 for comp WfProcMgr

```
change evtloglvl Englnv=5 for comp WfProcMgr
change evtloglvl StpExec=5 for comp WfProcMgr
change evtloglvl PrcExec=5 for comp WfProcMgr
change evtloglvl ObjMgrSqlLog=4 for comp WfProcMgr
```

- For Siebel Service Handheld application,

```
change evtloglvl HandheldSyncLog=5 for comp ServiceCEObjMgr_enu
```

- 8 Test the outbound workflow. Navigate to Activities > Activity List > All Activities, and create a new activity. Step off the record and go back to the activity, assign the owner of the activity to the mobile user. This event triggers the SMQ Outbound Activity to Primary Owned By workflow and a data message should reach the user's Siebel Handheld application if the Handheld device is connected to the queue manager. Server outbound messages are put in the [UserQueueName].SFQ directory temporarily until the Handheld device 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.
- 9 Sometimes, the workflow runtime events cannot be triggered after activation. Restarting the Siebel Server can fix this problem.
- 10 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.
- 11 SMQ Receiver must be enabled after the queue manager is set up and running.

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

Some common server-side error messages, and their solutions, are described in [Table 8](#).

Table 8. Server-Side Common Error Messages

Symptom or Error Message	Diagnostic Steps or Cause	Solution
MQe Internal Error: RemoveMsg: Message inside of Inbound Queue is not deleted successfully.  Error Code: 121	Error Code 121 means there is no matching message in the queue.  Multiple Receivers or Tasks may be trying to process or dispatch from the same queue at the same time.	Check svrmgr to find out how many SMQReceivers are running in the server.  Reconfigure as needed to make sure that there is only one SMQReceiver for each application.

Table 8. Server-Side Common Error Messages

Symptom or Error Message	Diagnostic Steps or Cause	Solution
ObjMgrBusServiceLog Error 1 000008e0440515c0:0 2006-03-02 00:05:23 (srmsyncsv.cpp (1214)) SBL-SRQ-00103: Unable to find definition for component WfProcMgr	This error may appear in the SMQReceiver log file.  The server components may not be synchronized to the internal table.	To synchronize the server components to the internal table, perform the following steps:  <b>1</b> In the Administration - Server Configuration > Synchronize screen, click Synchronize.  <b>2</b> Restart the SMQReceiver component.
JVM Init Exception: Can't load JVM dll: libjvm	This error may appear in the SMQReceiver log file on Solaris and Linux systems.  There may be an incorrect value in LD_LIBRARY_PATH.	Check the value of LD_LIBRARY_PATH by running one of the following commands:  ■ env  ■ echo \$LD_LIBRARY_PATH  On AIX, check \$LIBPATH.

Table 8. Server-Side Common Error Messages

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

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

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

- 1 Verify that the following parameters are in the setup.ini file. [Table 9](#) lists the default values.

Table 9. Parameter Values in the setup.ini File

Parameter	Default Value
MQeServerAddr	[hostname] or IP address of the computer hosting the queue manager
MQeServerCmd	/smq/ServerQM
MQeServerPort	8080
MQeServerQMgr	ServerQM
MQeServerQueue	InboundQ
QueuingSoftware	MQE. For IBM MQe only.
SMQCompression	The type of compression used for Store-and-Forward Messaging for Mobile Web Client. The only value currently supported is GZIP.
SMQConnectOnSend	Controls whether Message Transport software automatically attempts to connect to the server when a message is placed in the client's outbound queue. Default value is FALSE.
SMQConnectOnStart	Controls whether Message Transport software automatically attempts to connect to the server when a new client session is started. Default value is FALSE.
SMQEncryption,	The type of encryption used for Store-and-Forward Messaging for Mobile Web Client. The only value currently supported is RC4.
SMQLogMessages	When TRUE, copies of messages are logged as XML files. Default value is TRUE.
SMQPollingInterval	Specifies how often the Message Transport software polls for new messages. Default value is 30 seconds.

If any of the MQe parameter values do not match the output of the SiebelSMQServlet, then modify the setup.ini file and reinstall the Siebel Handheld application with the correct parameters. For more information, see ["Adjusting the Store-and-Forward Configuration Parameters"](#) on page 37.

- 2 Examine the Siebel Handheld application log file to find more information for troubleshooting. For information on working with the application log file, see the Siebel Handheld Guide for the Siebel Handheld application you are using.

## Siebel Store-and-Forward Messaging Client-Side Common Error Messages

Some common client-side error messages, and their solutions, are described in [Table 10](#).

Table 10. Client-Side Common Error Messages

Symptom/Error Message	Diagnostic Steps/Cause	Solution
Unable to initialize Wireless Messaging. Cannot connect to the server. Please ensure that you have an active connection and try again.	<p>Verify that the device has an active network connection.</p> <ul style="list-style-type: none"><li>■ Start a Pocket Internet Explorer on the device.</li><li>■ Check whether the Pocket Internet Explorer can access a server URL, such as the following:  <code>http://[hostname]/service_enu/s.swe</code></li></ul> <p>Also, a network connection by way of an ActiveSync session may have domain name resolution issues.</p>	<p>To resolve a network connection domain name, use a name that is resolvable from the ActiveSync host computer.</p> <p>If the MQeServerAddr contains points (for example, 10.11.12.13 or sfm.mycomp.com), then add an exception to the PDA connection settings. For more information, see <a href="#">“Adding an Exception to the PDA Connection Settings”</a> on page 46.</p>

Table 10. Client-Side Common Error Messages

Symptom/Error Message	Diagnostic Steps/Cause	Solution
<p>Wireless Messaging registration is pending. Would you like to continue to wait for server confirmation.</p>	<p>The device can send the first registration message to the server queue manager. However, the Siebel server cannot receive the message from the inbound queues, generate a response message, and send the response message back to the Siebel Handheld application.</p>	<p>Review the server-side troubleshooting check list to see whether an error is recorded in the log files. See <a href="#">“Siebel Store-and-Forward Messaging Server-Side Troubleshooting Checklist” on page 50</a>.</p>
<p>Unable to Initialize Wireless Messaging Timed Out while waiting for registration response from the server. Please try again or contact your system administrator.</p>	<p>If SFM device registration fails, the application will prompt for a retry.</p>	<p>Retry the synchronization, then wait 10 seconds for the acknowledgement message.</p> <p>If no acknowledgement arrives, the application asks whether to wait. Do one of the following:</p> <ul style="list-style-type: none"> <li>■ Tap Yes. The system waits another 10 seconds and prompts again. The maximum wait time is 10 minutes.</li> <li>■ Tap No. The system displays an error message and completes the synchronization.</li> </ul>



# 4

## Administration of Siebel Store-and-Forward Messaging

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

- [“Overview of Siebel Store-and-Forward Messaging Administration” on page 59](#)
- [“Process of Administering Applications with Siebel Store-and-Forward Messaging” on page 60](#)
- [“Process of Setting Up Siebel Store-and-Forward Messaging on a Siebel Handheld Application” on page 68](#)
- [“Process of Configuring Asynchronous Query for Siebel Handheld Applications” on page 74](#)

### Overview of Siebel Store-and-Forward Messaging Administration

Siebel Store-and-Forward Messaging administration screens allow you to administer mobile users, their Siebel Handheld devices, detailed Store-and-Forward Messaging message information, and timeout parameters. Store-and-Forward Messaging administration consists of three functional components, User Administration, Message Administration, and Message History.

- **User Administration.** Displays mobile user, user device address, mobile application and message information for each user.
- **Message Administration.** Displays how messages are constructed and allows an administrator to specify message timeout parameters.
- **Message History.** Provides key administrative information about each inbound and outbound message.

This chapter presents an overview of tasks you perform when administering Siebel Store-and-Forward Messaging. You administer Siebel Store-and-Forward Messaging for Handheld from the Administration—Store-and-Forward screen in a Siebel Web client.

**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 25](#).

# Process of Administering Applications with Siebel Store-and-Forward Messaging

Siebel Store-and-Forward Messaging supports Siebel Handheld applications which run on Pocket PC devices. For more information on Siebel Handheld, see the Siebel Handheld Guide for the Siebel Handheld application you are using. For more information on supported devices, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

In the Administration—Store-and-Forward > Store-and-Forward Applications view, the applications listed are identical to the Siebel Handheld applications displayed in the Administration-Mobile > Application Administration view. If you plan to configure a Siebel Handheld application and enable it to run Siebel Store-and-Forward Messaging, perform the following steps:

- “Enabling Applications for Store-and-Forward Messaging” on page 60
- “Administering Messages in Store-and-Forward Messaging” on page 60
- “Administering Handheld Devices for Store-and-Forward Messaging” on page 61
- “Reviewing and Deleting Store-and-Forward Messaging History” on page 62
- “Managing Outdated MQe Queues and Handheld Synchronization Extracts” on page 67
- “Moving PDA Users to a Different MQe Queue Manager and Inbound Queue” on page 67

## Enabling Applications for Store-and-Forward Messaging

**CAUTION:** The following procedure must be performed before the Pocket PC device with the newly configured application synchronizes to the handheld synchronization server for the first time.

### *To enable an application for Store-and-Forward Messaging*

- 1 Navigate to Administration - Mobile > Application Administration.
- 2 Click the Settings view tab, and create a new record with the following information in the Name and Value fields:
  - Name = SchemaChangeNotification
  - Value = SMQ Message Service

**NOTE:** The application now appears in the Administration—Store-and-Forward > Store-and-Forward Applications view.

## Administering Messages in Store-and-Forward Messaging

Once a Siebel Handheld application is enabled for Store-and-Forward Messaging, you must administer the data message definition from the Administration—Store-and-Forward screen.

### *To administer data messages*

- 1 Navigate to Administration—Store-and-Forward > Message List, and select an application.
- 2 The following Administration views and sub-views need to be populated for Store-and-Forward function:
  - **Message Objects.** Each data message object is based on a business object defined in the Siebel Handheld application. A message object consists typically of a record in the primary business component and all the descendant records in its descendant message components.
  - **Message Components.** A message component consists of various fields. Not all field values are included in the message components.
  - **Fields.** Only the fields displayed in the message fields are constructed in a data message.

You can populate these views manually, or they will be automatically populated during the first handheld synchronization process.

## Manually Populating Message Objects, Components, and Fields

For detailed information on how to manually populate these views, see [“Creating Store-and-Forward Messaging Message Objects” on page 26](#).

## Automatically Populating Message Objects, Components and Fields

If a Siebel Handheld application has been enabled for Store-and-Forward Messaging and you have successfully installed an instance of the Siebel Handheld application on a Pocket PC device, the first handheld synchronization process populates the message objects, message components and fields automatically. This initial synchronization process may take longer than usual. For information on Siebel Handheld application synchronization, see the Siebel Handheld Guide for the application you are using.

## Administering Handheld Devices for Store-and-Forward Messaging

When a Siebel Handheld application with Store-and-Forward Messaging enabled is launched the first time, the handheld device sends a registration message to register itself on the Siebel server. When the Siebel Handheld application is started the first time, make sure that the device has a wired or wireless network connection with the queue manager and the Siebel server. After the handheld device is registered, the mobile user and device information are populated. You can review the mobile user and device information from the Administration—Store-and-Forward > User List view.

### *To examine information about a registered handheld device*

- 1 Navigate to Administration - Store-and-Forward > User List.
- 2 In the Persons list, highlight a user, then click the Device Administration view tab.

**3** Review the information.

The following table defines the fields in the Device Administration view.

Field	Description
Address Type	The queuing system used by the Siebel Handheld application, which is IBM MQe.
Device Address	The device's inbound queue name.
Device Name	Description of the registered handheld device.
Mobile Application	The name of the Siebel Handheld application.
Sync Node ID	An ID used by Handheld Sync Server to uniquely identify a device.

## Reviewing and Deleting Store-and-Forward Messaging History

You can review message histories in the Store-and-Forward Message History view for all the inbound and outbound messages. You may also review an individual user's inbound and outbound messages in the Personal Message History view. These views allow you to access characteristics of an inbound or outbound message, including the unique message ID, the application used for the message, the message direction - either from server or from client, 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 62](#)
- ["To delete all old message history records" on page 65](#)
- ["The following procedure describes how to review message history records for a specific user's inbound and outbound messages." on page 66](#)
- ["To delete a user's old message history records" on page 66](#)

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

### ***To review all message history records***

- 1** Navigate to Administration - Store-and-Forward > Store-and-Forward Message History.

**2** Review the information in the Store-and-Forward Message History view.

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

Field	Description
Application	The name of the Siebel Handheld application.
Direction	The direction the message has traveled. Either from Server or From Client.
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. For information about message type codes and descriptions, see <a href="#">"About Message Types" on page 64</a> .
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 Data Message Timeout Parameters for Store-and-Forward Messaging”</a> on page 27.</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 Handheld application.</li> <li>■ <b>Client Received.</b> The Handheld 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 Data Message Timeout Parameters for Store-and-Forward Messaging”</a> on page 27.</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.

## About Message Types

Store-and-Forward Messaging messages fall into the following categories:

- **Outbound Messages.** Generated on a Siebel Server for delivery to a Siebel Handheld application.
- **Inbound Messages.** Generated on a Siebel Handheld application for delivery to a Siebel Server.

This topic describes the message types and their associated codes. These message types and codes appear in the Store-and-Forward Message History view's Message Type fields. Some of these message types are specific to From Server (outbound) messages only, some are specific to To Server (inbound) messages only, and some can be used by both (bi-directional).

The message types and their associated message codes are listed and described in [Table 11](#).

Table 11. Message Types

Message Code	Direction	Comments
ACK	Inbound and Outbound	Acknowledgement message. Indicates that a message has been successfully received and processed. Contains the original message ID and the Transaction ID if the original message was a TXN message.
DAT	Outbound	Data message. Contains record-level data from a single business component and its related child business components. May also contain a message ID and a list of recipient users. The data is in XML format. The data can be a Deep message (the default), which provides a parent record and all child and grandchild records; or a Surface message, which contains only the parent level.
ERR	Inbound and Outbound	Error message. Informs the target Siebel Handheld application that a message was not successfully handled on the server side or that another error condition occurred. Contains information about the error, including the error code, original message ID, transaction ID, conflicting value (if any), BusComp field that caused the problem, BusComp name, and BusComp Row Id.
INV	Inbound and Outbound	Invoke message. Contains a command to invoke a business service's method.
PNG	Outbound	Ping message. Used to determine whether a Siebel Handheld application is connected with the Siebel Server. Returns the original message ID in reply to a Ping request.
REQ	Inbound	Request message. Requests a record or a set of records. These can be a specific BusComp record, a set of BusComp records, or the Pick data for a picklist.
TXN	Inbound and Outbound	Transaction message. Delivers a change specified by a remote user to the Siebel system. The change is a record-level action (Associate, Delete, Insert, or Update) specified in the message. The message identifies the affected record by Business Object, Business Component, and record Row ID.

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

***To delete all old message history records***

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

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

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

- 4 Click Delete All Messages.

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

#### ***To review a user's message history records***

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

The table in ["To review all message history records" on page 62](#) describes the fields in the Message History list.

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

#### ***To delete a user's old message history records***

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

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

- 5 Click Delete All Messages.

## Managing Outdated MQe Queues and Handheld Synchronization Extracts

To conserve storage space, MQe queues and handheld synchronization extracts that are no longer needed can be removed from your company's system. For help with removing your obsolete MQe queues and outdated handheld synchronization extracts, create a service request (SR) on My Oracle Support. Alternatively, you can phone Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers remain the same and are listed on My Oracle Support.

**CAUTION:** Do *not* attempt to remove obsolete MQe queues and outdated handheld synchronization extracts yourself. Doing so could cause unpredictable results or accidental data loss.

## Moving PDA Users to a Different MQe Queue Manager and Inbound Queue

A high-volume Siebel Handheld Store-and-Forward Messaging environment can have multiple MQe queue managers and multiple MQe generic inbound queues. At installation, each PDA is assigned to a specific queue manager and generic inbound queue. To bypass a problem such as a server outage, PDA users can be moved from the original queue manager and inbound queue to another queue manager and inbound queue.

This topic sets forth a way to switch PDA users to a secondary server, if multiple MQe servers are present and the primary server fails. This topic describes how to move a handheld device from one MQe queue manager and inbound queue to another, where the following circumstances apply:

- There is one primary MQe server, with a backup server to which you can move users if the primary server fails.
- You have access to the registry editor, either from a Siebel application or from CE Tools, which can run on a desktop computer and access the PDA device registry through ActiveSync.
- The SMQ server has been configured, and Workflow Manager and HandHeldSync are running.

If your company network is set up in a different configuration, then you can vary the steps of this procedure accordingly.

### *To move a user to a different MQe queue manager and inbound queue*

- 1 Launch the registry editor and access the PDA device registry.

- 2 In the PDA device registry, change the values of the fields in the following table.

Field	Comments	Example Original Value	Example New Value
MQeServerAddr	MQe server	Server1	Server2
MQeServerQMgr	MQe queue manager	ServerQM1	ServerQM2
MQeServerQueue	Inbound queue	InboundQ1	InboundQ2

Change only the fields that are required.

**NOTE:** It is not necessary to reregister the PDA, because the device has been registered before and the user is already in the mobile user list.

- 3 Send an SMQ message to the server and verify, on the server side, that a queue named *InboundQ.Client Queue Manager Name* is created, for example:

InboundQ.ServerQM2

The workflow retrieves the message from the queue, processes the message, and creates the outbound queue.

When you change the server, the client continues to push messages to and pull messages from the server that is specified in the device registry.

## Process of Setting Up Siebel Store-and-Forward Messaging on a Siebel Handheld Application

The Siebel Service Handheld application is configured by default to run Siebel Store-and-Forward Messaging.

To set up Siebel Store-and-Forward Messaging on any other Siebel Handheld application, you must perform the following tasks:

- [“Preconfiguring the Siebel Handheld Application” on page 69](#)
- [“Adding Store-and-Forward Messaging Views to the Handheld Application” on page 70](#)
- [“Adding Alter Sound User Properties to the Siebel Handheld Application” on page 70](#)
- [“Adding the Siebel Store-and-Forward Messaging Toolbar to the Handheld Application” on page 71](#)
- [“Adding Application Find to the Handheld Application” on page 71](#)
- [“Compiling Changes Made to the Handheld Application” on page 72](#)
- [“Updating the User’s Responsibilities and Views to Use Store-and-Forward Messaging” on page 72](#)
- [“Adding Settings to Enable Store-and-Forward Messaging” on page 73](#)

- [“Setting up the Siebel Handheld Application Setup.ini File for Store-and-Forward Messaging” on page 74](#)
- [“Installing the Siebel Handheld Application” on page 74](#)

## Preconfiguring the Siebel Handheld Application

There are two steps to follow when preconfiguring your Siebel Handheld application. They are listed below:

### *To modify the Siebel Tools .cfg file*

- 1 Open the Siebel Tools configuration file: [install directory]\tools\bin\enu\tools.cfg.
- 2 Make sure the Siebel Tools database connection information is correct.
- 3 In the [Siebel] section, verify that ClientConfigurationMode = All.
- 4 Save and close the file.
- 5 Open Siebel Tools.

### *To preconfigure an application*

- 1 Launch Siebel Tools and navigate to View > Options > Object Explorer.
- 2 In the Development Tools Options box, make sure the following items are checked:
  - Applet
  - Application
  - Business Component
  - Business Object
  - Find
  - Screen
  - Toolbar
  - View
  - Workflow Process
- 3 Click OK.
- 4 In the Object Explorer, click the Types tab, then click Application.
- 5 Query the application list for your Siebel Handheld application.  
For example, Siebel HH LS Medical.

Select the application, then from the Application level menu, select Tools > Lock Project.

## Adding Store-and-Forward Messaging Views to the Handheld Application

The following task details how to add Store-and-Forward Messaging Views to your Siebel Handheld application.

### *To add Store-and-Forward Messaging views to the application*

- 1 In the Siebel Tools Object Explorer, navigate to the Screen object.
- 2 Create a new record, or copy the SHCE Service Hidden Views Screen and modify it according to the steps below.
- 3 Add the following views to the hidden views screen created in the previous step:
  - SHCE Async Query List view
  - SHCE Wireless Alerts List view

These two views have been created by default and are available for use. You need to configure an Asynchronous Query Results view and an inaccessible view for unsupported business components in the tasks which follow. Make sure you have added these newly created views into the hidden views screen, as detailed in this task. For more information, see [“Process of Configuring Asynchronous Query for Siebel Handheld Applications” on page 74](#).

- 4 In the Object Explorer, expand the Application list then choose Screen Menu Item. Add the newly created screen.

## Adding Alter Sound User Properties to the Siebel Handheld Application

Different sounds can be associated with the five different alert levels by specifying the .wav file in the application user properties.

### *To add alter sound user properties to a Siebel Handheld application*

- 1 In the Siebel Tools Object Explorer, choose Application > Application User Properties.
- 2 Select your Siebel Handheld application from the Application list.  
For example, Siebel HH LS Medical.
- 3 Insert the following new records into the Application User Properties list:
  - WiMeAlertMessage\_1\_Sound
  - WiMeAlertMessage\_2\_Sound
  - WiMeAlertMessage\_3\_Sound
  - WiMeAlertMessage\_4\_Sound
  - WiMeAsyncMessage\_Sound

- 4 Set the value for each user property to the name of a .wav file that exists on your particular handheld device.

Available .wav files are determined by individual device manufacturers. Consult the documentation that accompanied your handheld device.

## Adding the Siebel Store-and-Forward Messaging Toolbar to the Handheld Application

The Store-and-Forward Messaging Toolbar, which appears in the upper right corner of the Siebel Handheld application when Store-and-Forward Messaging is enabled, has been configured by default, but must be added to the Siebel Handheld application. The tool bar contains the following icons that flash when an event has occurred.

- **Inbound Message Icon.** Indicates that a message has arrived at the Siebel Handheld application.
- **Asynchronous Query Icon.** Indicates that asynchronous query results have been received from the server.
- **Outbound Message Icon.** Indicates that an outbound message has been placed in the application's outbound queue and awaits delivery.

To add the tool bar to the Siebel Handheld application, you need to perform the following task:

### *To add the Store-and-Forward Messaging tool bar to the Siebel Handheld application*

- 1 Open Siebel Tools, and in the Object Explorer, choose Application > Toolbar.  
**NOTE:** If you do not see this choice in the Object Explorer, you have either not made the correct changes to the tools.cfg file in ["To add the Store-and-Forward Messaging tool bar to the Siebel Handheld application" on page 71](#), or you have not checked all the items that need to be exposed in ["Preconfiguring the Siebel Handheld Application" on page 69](#).
- 2 Select your application from the Application drop-down list.  
For example, Siebel HH LS Medical.
- 3 Insert a new record in the Application Toolbar list with the following value: SHCE Top.
- 4 Recompile your project.

## Adding Application Find to the Handheld Application

Any Find object configured for asynchronous query must be added to the application object.

### *To add application find to application*

- 1 In the Siebel Tools Object Explorer choose Application > Find.

- 2 Select your application from the Application list.  
For example, Siebel HH LS Medical.
- 3 Insert a new record in the Application Find list for the asynchronous queries that you want to support.

## Compiling Changes Made to the Handheld Application

You must compile changes made to any objects you change or create.

### *To compile Siebel Tools changes to the Siebel Handheld application*

- 1 In Siebel Tools, select your Siebel Handheld application in the Application list.
- 2 Right-click, and select Compile Selected Objects.
- 3 Click browse to specify the new .srf file to the appropriate location.  
For example, Siebel Industry Applications use [install directory]\objects\enu\siebel\_sia.srf and Siebel Business Applications use siebel.srf by default.
- 4 Make sure there are no applications running and that the database icon is not in your status area of the system task bar (if it is, right-click the database icon, and choose Exit).
- 5 Select the object shown and click Compile.
- 6 Replace the server .srf file with your newly compiled .srf file.

## Updating the User's Responsibilities and Views to Use Store-and-Forward Messaging

You must update and the responsibilities to the newly created or newly altered views.

### *To update the user's responsibilities and views for Store-and-Forward Messaging*

- 1 Log into a Siebel Web Client as a system administrator.
- 2 Navigate to Administration - Application > Responsibilities.
- 3 Add SADMIN and handheld users to the Siebel Handheld user's responsibility by doing the following:
  - Query for the responsibility.  
For example, Medical Sales - HH CE.
  - Insert SADMIN, and the user ID of the handheld users in the Users applet.
- 4 Add the new views to the handheld user's responsibility by doing the following:

- Query for the responsibility.  
For example, Medical Sales - HH CE.
- Insert the following new records into the Views applet:
  - SHCE Async Query List view
  - SHCE Wireless Alerts List view

**NOTE:** If you cannot insert the new records into the views applet, you can add these new views through [Site Map > Administration-Application > Views](#).

- 5 Query for the SMQ User responsibility, then insert SADMIN into the Users applet.
- 6 Select the Organization for the SMQ User responsibility and verify whether the Siebel Handheld user's organization is included.  
For example, Medical Products MD ENU is the user SADMIN's organization.

## Adding Settings to Enable Store-and-Forward Messaging

New settings are necessary to properly extract the Store-and-Forward Messaging views.

### *To add settings to enable Store-and-Forward Messaging*

- 1 Navigate to Administration - Mobile > Application Administration > Settings.
- 2 Query for your Siebel Handheld application.  
For example, Siebel HH LS Medical.
- 3 Create the following new Settings records:

Name	Value
NoDatabaseBusComp	<name of unsupported business component>, Sequence = 1
NoExtractBusComp	SHCE Async Query, Sequence = 1
NoExtractBusComp	SHCE Async Query Results, Sequence = 2
NoExtractBusComp	SHCE Wireless Alerts, Sequence = 3
SchemaChangeNotification	SMQ Message Service, Sequence = 1

For example, the Solution business component, for the SHCE Service application. For information on configuring unsupported business components, see [“Configuring Unsupported Business Components for Asynchronous Query” on page 79](#).

For information on entering information for the Siebel Handheld application, see [“Entering Information for the Store-and-Forward Messaging Application” on page 25](#).

## Setting up the Siebel Handheld Application Setup.ini File for Store-and-Forward Messaging

To enable Store-and-Forward Messaging for your Siebel Handheld application, you need to make certain changes to the setup.ini file of the Siebel Handheld application.

### *To set up the setup.ini file*

- 1 Make any changes to the [WirelessMessaging] section of the setup.ini file as shown in [“Preparing for Installation of Siebel Handheld Applications”](#) on page 44.
- 2 Add the settings for the SyncURL, MQeServerAddr, and MQeServerPort parameters. See [“To verify and edit the setup.ini parameters”](#) on page 44.

For more information about installing the Siebel Handheld application, see [“Process of Installing and Configuring Handheld Applications”](#) on page 44.

## Installing the Siebel Handheld Application

For instructions on installation of the Siebel Handheld application, see the Siebel Handheld Guide for the Siebel Handheld application you are using.

## Process of Configuring Asynchronous Query for Siebel Handheld Applications

Asynchronous Query has been configured, by default, for the Siebel Service Handheld application. Perform the following tasks to configure Asynchronous Query functionality for any other Siebel Handheld application.

- [“Configuring the Find Object in Siebel Tools”](#) on page 74
- [“Creating the Asynchronous Query Results Applet”](#) on page 77
- [“Creating the Asynchronous Query Results View”](#) on page 78
- [“Adding Each New View to the Hidden Screens View”](#) on page 78
- [“Adding Each New View and Responsibilities to the Application”](#) on page 79
- [“Configuring Unsupported Business Components for Asynchronous Query”](#) on page 79

## Configuring the Find Object in Siebel Tools

The Find dialog on the handheld device, which contains the functionality for local find and server find (asynchronous query), is configured in Siebel Tools. For information on configuring Find Objects, see *Siebel Search Administration Guide*.

You start the configuration process by adding a Find object. There are two views for each Find object. One view is rendered when a user performs a local find. The other view is for asynchronous query only and is based on the business object: SHCE Async Query.

When the user taps the Details button in the Asynchronous Query List applet, the Siebel Handheld application identifies the Find object of the highlighted row and renders the view that is related to the SHCE Async Query business object. This view, and its applet, display the asynchronous query results for the Find object.

When the user clicks on the Find button (local find) in the Find dialog, the view that is not related to the SHCE Async Query business object is rendered. The visibility of this view is used for Asynchronous Query as well.

### ***To configure the Find object***

- 1 In the Siebel Tools Object Explorer, click Find, then query for the Find object you wish to enable for asynchronous query.

If the Find object does not exist, create a new record.

Some fields of the Find object are explained in the table below:

Property	Description
Applet	Describes the applet that is rendered when the user clicks on the local Find button. When a user performs a local find (by tapping the Find button in the Find dialog on the Siebel Handheld application), the user is navigated to the specified applet and a local query is performed.
Display Name	The name that is displayed in the Find dialog Look In drop-down list. For example, by default the Siebel Service Handheld application contains Orders, Assets, Products, and Solutions.

### To add Find fields

- 1 In the Siebel Tools Object Explorer, click Find > Find Field.

The available fields in the Field property associated with the business component of the applet specified in the Applet property. For example, the Order-SHCEService Find object uses SHCE Service Orders List Applet - Find. The business component of that list applet is Order Entry - Orders.

The necessary fields for the Find Field are explained below.

Field	Description
Display in Entry	Indicates that the field is available in the Find Dialog when the parent Find Object is selected in the Look In drop-down list. In other words, the user can enter query text for this field.
Display in Results	Indicates that data is retrieved for this field during an asynchronous query. This property is only relevant for asynchronous query (server find), as the data returned for local find is dictated by the available columns in the applet specified in the Applet property of the Find object.
Display Name	The name displayed in the Find Dialog Look In drop-down list. For example, the default Siebel Service Handheld contains Orders, Assets, Products, and Solutions.

### To add Find views

- 1 In the Object Explorer, click Find > Find View.
- 2 Add the view.

**NOTE:** Only two views should be added: one for local find, and one for asynchronous query (server find).

**For Local Find.** A view that contains the applet specified by the parent Find Object Applet property should be added. When a local find is performed in the client Find Dialog, the user is navigated to this view. For example, if a local query is done on Orders in the Service Handheld application, the user is navigated to SHCE Service Asset List View - Find, which contains the applet SHCE Service Assets List Applet - Find that was specified in the parent Find Object local properties. Note that the visibility of this view is used by the asynchronous query feature when requesting server information. For example, if the visibility is designated as Personal, then the server query only returns Personal results.

**For Server Find.** When a user taps the Asynchronous Query icon in the upper right corner of the screen, the SHCE Async Query List view appears. The SHCE Async Query List view has been configured and is available for use. From the SHCE Async Query List view or applet, the user can highlight a query and tap the Details button. This operation brings up an Asynchronous Query Results view or applet, which should be configured and added into the application. For information on configuring the Asynchronous Query Results applet and view, see [“Creating the Asynchronous Query Results Applet” on page 77](#) and [“Creating the Asynchronous Query Results View” on page 78](#).

## Creating the Asynchronous Query Results Applet

Once the Find object has been configured, it is necessary to create the applet that is used to display the results to the user. When the user clicks the Server Find button in the Find dialog box, a record is added to the SHCE Async Query business component, and can be viewed in the SHCE Async Query List view by clicking the Asynchronous Query icon. The Asynchronous Query icon is only visible when Store-and-Forward Messaging is enabled. See [“Using Asynchronous Query with the Siebel Service Handheld Application” on page 20](#) for an example. When the server returns the results of the asynchronous query, they are placed in the SHCE Async Query Results business component, which is a child of SHCE Async Query business object. The user can access these results by tapping the Details button in the SHCE Async Query List view.

### To create an applet to display asynchronous query results

- 1 Create a new list applet, or copy an existing Asynchronous Query Result applet, such as SHCE Service Async Query Results Asset List applet.

The new or copied applet must have the following attributes:

- The applet must be a list applet.
- The business component property must be SHCE Async Query Results. In other words, all results applets are based on this business component.
- This applet should have the following fields checked:
  - No Insert
  - No Delete
  - No Update.
- When mapping applet fields to business component fields (Data 1, Data 2, ... Data 6 of SHCE Async Query Results) it is necessary to make sure that the business component field corresponds the correct Find object field. Returned asynchronous query results are populated in sequence of the Find object fields that have the 'Display in Results' flag set. In other words, Data 5 maps to the 5th find object field with the flag. See the following example:

Find Field Name	Sequence Number	Display in Results	Display in Entry	Data Column Map
A	0	Y	Y	Data 1
B	1	Y	N	Data 2
C	2	N	Y	None (not displayed)
D	3	Y	Y	Data 3
E	4	Y	Y	Data 4
F	5	N	Y	None (not displayed)
G	6	Y	N	Data 5

- 2 Use the Edit Web Layout feature in Siebel Tools to configure the user interface. This determines the order the query result columns appear in the user interface.

## Creating the Asynchronous Query Results View

If an Asynchronous Query Results view has not been created, create it by following these steps.

### To create the view object

- 1 In the Siebel Tools Object Explorer, navigate to the View object and create a new record using the guideline listed below.  
**NOTE:** The property business object for the view needs to be SHCE Async Query.
- 2 Add the Async Query Results applet created in [“Creating the Asynchronous Query Results Applet” on page 77](#) to this view.
- 3 Add the following user properties to the view to implement the pseudo popup behavior. Doing this makes the view behave like a popup rather than a view.

Name	Value
SHCEHideNavBar	Y
SHCEHideToolBar	Y
SHCEPseudoPopup	Y

See the SHCE Service Async Query Results Asset List view for more information. If you copy the SHCE Service Async Query Results Asset List view, make sure you change the fields for the Visibility Applet and the Thread Applet and point them to the newly created Asynchronous Query Results applet.

## Adding Each New View to the Hidden Screens View

A screen that contains the SHCE Async Query List applet as well as the SHCE Wireless Alerts List view should already be added to the Application > Screen Menu Item in the Tools Object Explorer for your application. The new views should be added to this screen. By adding the screen in Application > Screen Menu Item, the definitions will be extracted during the synchronization process.

**NOTE:** To have the screen appear in the drop-down screen menu, you have to add the screen to the Application > Page Tab. However, since the default buttons provide navigation to the default views, this hidden screen is excluded from the Page Tabs.

For an example, see the SHCE Service Hidden Views Screen.

## Adding Each New View and Responsibilities to the Application

You can perform this task by logging into a Siebel Web Client as a system administrator.

### *To add the new views*

- 1 Navigate to Administration - Application > Views.
- 2 Add the new views.

### *To add the views to the responsibilities*

- 1 Navigate to Administration - Application > Responsibilities (or Responsibility List).
- 2 Add the views to the responsibilities.

## Configuring Unsupported Business Components for Asynchronous Query

An unsupported business component is one that is not supported on the client but is available on the server. You can configure the Siebel Handheld application to support asynchronous query on these business components.

The Applet property of the Find object is mandatory and dictates the available Find fields that can be added. The available fields are the business component fields of the specified Find object. It is therefore necessary to create a new view based on the unsupported business component. For more information, see [“Configuring the Find Object in Siebel Tools” on page 74](#). Although this view is inaccessible to the end user, it provides the visibility information for the server asynchronous query operation. You must still configure the Async Query Results view separately. In summary, there are two Find views for each Find object for an unsupported business component. One view, the inaccessible view, is based on the Find object’s business component (such as Accounts, Contacts, Solutions, and Orders), which may be an unsupported business component on the Siebel Handheld application. The other view is the Async Query Results view that you configure separately. The Async Query Results view is used to display the results of asynchronous queries.

The following two sections provide an example of configuring an inaccessible applet and view for the Solutions business component, which is an unsupported business component in the Siebel Service Handheld application.

## Example of Configuring an Inaccessible Applet

The following is an example for configuring an inaccessible applet.

## Creating a New Applet: SHCE Sales Solutions Find Applet

### *To create a new applet*

- 1 In Siebel Tools, create a new applet, then specify the business component to the one you wish to use.

For instance, Solution.

Make sure that you have built a Web template. It need not contain any fields. The sync server will not extract the applet if there is no Web template. For information on Web templates, see *Siebel Developer's Reference*.

## Creating a New View: SHCE Sales Solutions Find View

### *To create a new view*

- 1 Set the applet from the previous task as the view's default and only applet.
- 2 Specify the visibility on the view that you wish to use for asynchronous queries against this business component.
- 3 Make sure to add this view to the user's responsibility. In the Administration - Application view, add the view and then add the responsibility.

For more information, see ["Adding Each New View and Responsibilities to the Application"](#) on page 79.

- 4 Create a new Find object called: Solution-SHCESales.  
See ["Configuring the Find Object in Siebel Tools"](#) on page 74.
  - a Specify the applet as the one defined in the SHCE Sales Solutions Find Applet.
  - b Choose the fields to expose in entry and in the results.
- 5 Follow the instructions in the previous sections to do the following:
  - a Add the Find object to the application. See, ["Adding Application Find to the Handheld Application"](#) on page 71.
  - b Add the view to the hidden screen you created earlier in the application. See, ["Adding Each New View to the Hidden Screens View"](#) on page 78.
  - c Add the view to the application list of views and add appropriate responsibilities.
- 6 Update synchronization settings to avoid extracting the data for the new view. If this is not done, then the data for this unsupported business component will be extracted unnecessarily.
  - a In the Siebel Web client, navigate to Administration - Mobile > Application Administration.
  - b Choose the name of the application(s) that use this unsupported business component, and select the Settings tab.
  - c Add a new record where the Name property = "NoDatabaseBusComp" and the Value contains the name of the unsupported business component.

# 5

## Siebel Store-and-Forward Messaging Sample Workflows

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

- [“About Store-and-Forward Messaging Workflows” on page 81](#)
- [“SMQ Outbound Activity to One Recipient” on page 82](#)
- [“SMQ Outbound Activity to Primary Owned By” on page 83](#)
- [“SMQ Outbound Activity to Team” on page 84](#)
- [“SMQ SMS Outbound to one recipient” on page 84](#)
- [“SMQ Delete Old Message History” on page 85](#)
- [“SMQ Process User Queue” on page 85](#)
- [“Transaction Delete Message Workflows” on page 86](#)

### 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*.

## SMQ Outbound Activity to One Recipient

As seen in Figure 6, the SMQ Outbound Activity to one recipient workflow is based on the Action business object, and is an outbound workflow defined for a mobile service technician. The workflow generates a server outbound data message containing the information about a dispatched activity, formats the message, sends the message to a server outbound queue, and updates the message history table.

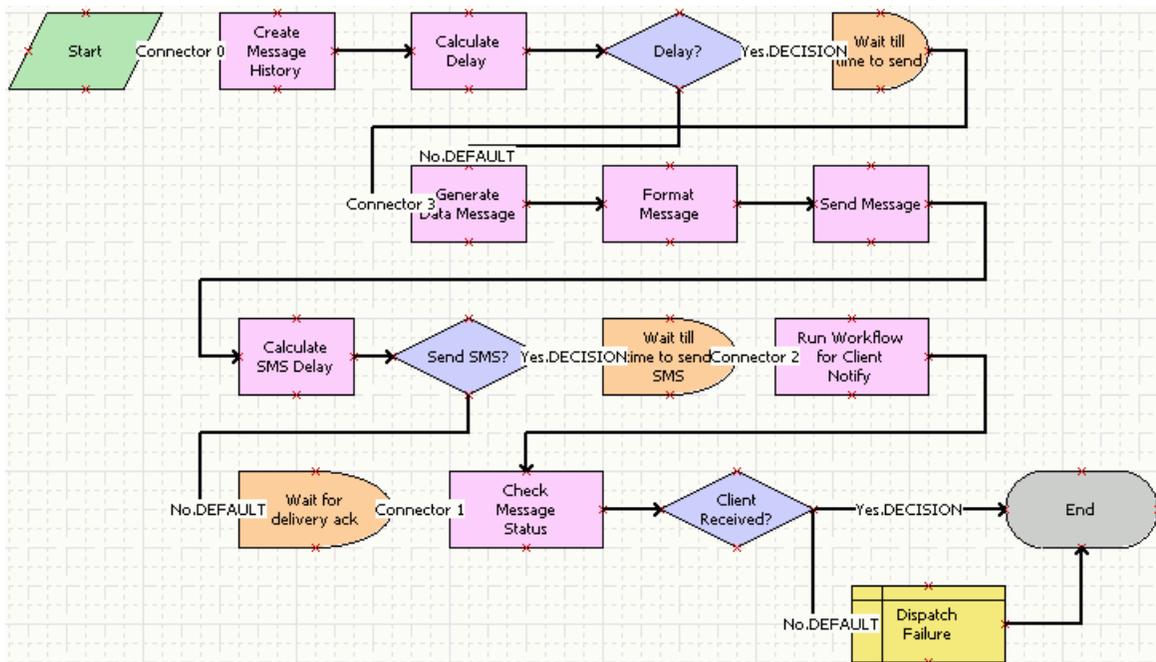


Figure 6. SMQ Outbound Activity to one recipient Workflow

The main process properties include Application Name, Message Object Name, and Message Recipient. The Application Name and Message Object Name must be defined in the Administration-Store-and-Forward > Store-and-Forward Applications view.

This workflow contains the following steps:

- 1 Create Message History: This step creates a message history record in the message history table and sets the message status to Pending.
- 2 Calculate Delay: The workflow may be put to sleep if a Start to Send parameter is specified in the Store-and-Forward Timeout Administration view.  
For information about the Start to Send parameter, see [“Configuring Data Message Timeout Parameters for Store-and-Forward Messaging”](#) on page 27.
- 3 Generate Data Message: This step takes the message object ID as well as the related business object name and extracts the related data to construct a server outbound data message.
- 4 Format Message: This step takes the data extracted and formats it to a well-formed XML document.

- 5 Send Message: This step sends the XML document into the user's server outbound queue.
- 6 Calculate SMS Delay: If a Message Available Notification time is being specified in the Store-and-Forward Message Timeout View, this step calculates the sleep time.
- 7 Wait till time to send SMS: This step waits until the send time for the SMS message to the mobile device.
- 8 Run Workflow for Client Notify: The ProcessName input argument specifies the workflow that notifies the client device. It is set to SMQ SMS Outbound to one recipient as an example to send an SMS message to the client device.
- 9 Wait for delivery ack: This step waits for an ACK (Acknowledgement) message from the MQe queuing system, which indicates that the message reached the targeted handheld device (message state changed to Received).
- 10 Check Message Status: This step checks the message status. If the status of the message has been changed to Client-Received, the message has arrived at the client side.
- 11 Dispatch Failure: If the status of the message is something else besides Client-Received, this step sets the status of the activity record to Dispatch Failure.

## SMQ Outbound Activity to Primary Owned By

As shown in [Figure 7](#), this outbound workflow uses the SMQ Outbound Activity to one recipient workflow as a subworkflow. This workflow is triggered by the WriteRecordUpdated runtime event in the Action business component. When an activity is assigned to a mobile user, or when an activity owned by a mobile user is updated, the workflow sends a server outbound data message to the mobile user with the updated activity and its associated data.

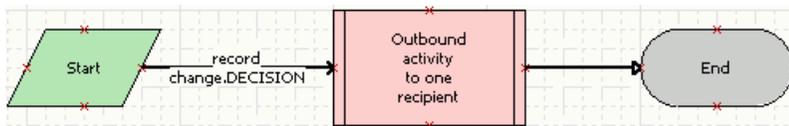


Figure 7. SMQ Outbound Activity to Primary Owned By Workflow

This workflow provides three main process properties to the SMQ Outbound Activity to One Recipient workflow.

- 1 Application Name: set to Siebel Service for CE, which is Service Handheld application.
- 2 Message Object: set to ActionSHCE, which is the business object enabled for Store-and-Forward Messaging for the Service Handheld application.
- 3 Recipient: set to the Primary Owned By field value of an activity record. This value identifies the primary owner of the activity.

In addition, a condition of "[Updated By]<>[Primary Owner Id]) AND [Status]<>LookupValue('Event Status', 'Dispatch Failed')]" is specified on the branch following the start step. This condition filters out two undesirable triggering events, the record changes that are made by the owner himself, and the changes that are made by the workflow itself because the related outbound data messages are not delivered within the Message Delivery Timeout.

## SMQ Outbound Activity to Team

As shown in [Figure 8](#), the SMQ Outbound Activity to Team workflow also uses the SMQ Outbound Activity to one recipient workflow as a subworkflow. Some customers may need to send the same activity record to all the service technicians on the same team. The workflow allows the system to send the same activity message to all the service technicians on the same team.

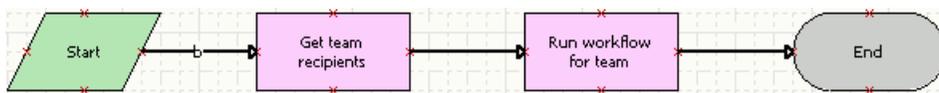


Figure 8. SMQ Outbound Activity to Team Workflow

This workflow shows how it can be achieved in the following two steps.

- 1 Get Team Recipients: The input argument TeamField specifies a multivalue field to derive the set of recipients. Currently the input argument is set to Owned By.
- 2 Run Workflow for Team: The input argument ProcessName specifies the workflow to be run for every technician in the team. This argument is set to SMQ Outbound Activity to One Recipient.

## SMQ SMS Outbound to one recipient

This is an outbound workflow for a service technician. This workflow is invoked by the Run Workflow for Client Notify step of the [SMQ Outbound Activity to One Recipient](#) workflow. It generates a SMS message to notify the client that there is a Store-and-Forward Message awaiting transmission in the SMQ Server. This alerts the technician to set her mobile device to retrieve the message from the server. The workflow is shown in [Figure 9](#).

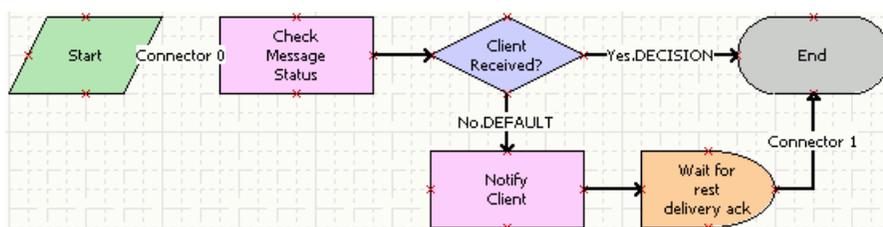


Figure 9. SMQ SMS Outbound to one recipient Workflow

The main process properties include Message Status, Recipient, and SMS Message Body. The workflow contains the following steps:

- 1 Check Message Status: Checks the status of the message in the Store-and-Forward Message History View.
  - If the status is Client-Received, the message has already arrived at the client. In this case, the workflow does not generate a notification message.
  - If the status is not Client-Received, the workflow generates an SMS message to notify the client.
- 2 Notify Client: Sends the SMS message to the client.
- 3 Wait for rest delivery ack: Waits for an acknowledgement from the queuing system to indicate whether the message has reached the mobile device.

## SMQ Delete Old Message History

This workflow deletes a record from the Message History Table. The workflow is shown in [Figure 10](#).



Figure 10. SMQ Delete Old Message History Workflow

The Delete Old Message step has the following main process properties:

- MessageAge. Input value from Delete Message Popup UI.
- Person Id. User name for SMQ Personal Message History Applet Delete Message operation.

## SMQ Process User Queue

As shown in [Figure 11](#), the SMQ Process User Queue is the inbound workflow. This workflow picks up server inbound messages from each user inbound queue and processes them.



Figure 11. SMQ Process User Queue Workflow

This workflow has the following input arguments in the Process User Queue step, which can be customized:

- 1 Application Name: Set to Siebel Service for CE, which is Service Handheld application.
- 2 TimeOut: Time to wait for next inbound message's arrival before the workflow ends. It is currently set to five seconds.

## 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 86](#)
- [“SMQ PreDelete FS Invoice Record” on page 87](#)
- [“SMQ Delete FS Invoice Record” on page 87](#)
- [“SMQ Outbound Delete in Action Business Object” on page 88](#)
- [“Configuring Transaction Delete Workflows for Different Business Components and Business Objects” on page 88](#)

### 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. For more information about message types, see [“About Message Types” on page 64](#).

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 12](#).



Figure 12. 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 13](#).

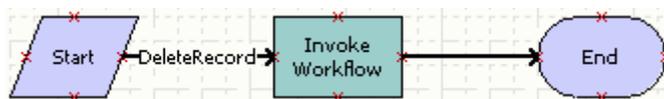


Figure 13. SMQ Delete FS Invoice Record Workflow

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 88.

## 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 14](#).

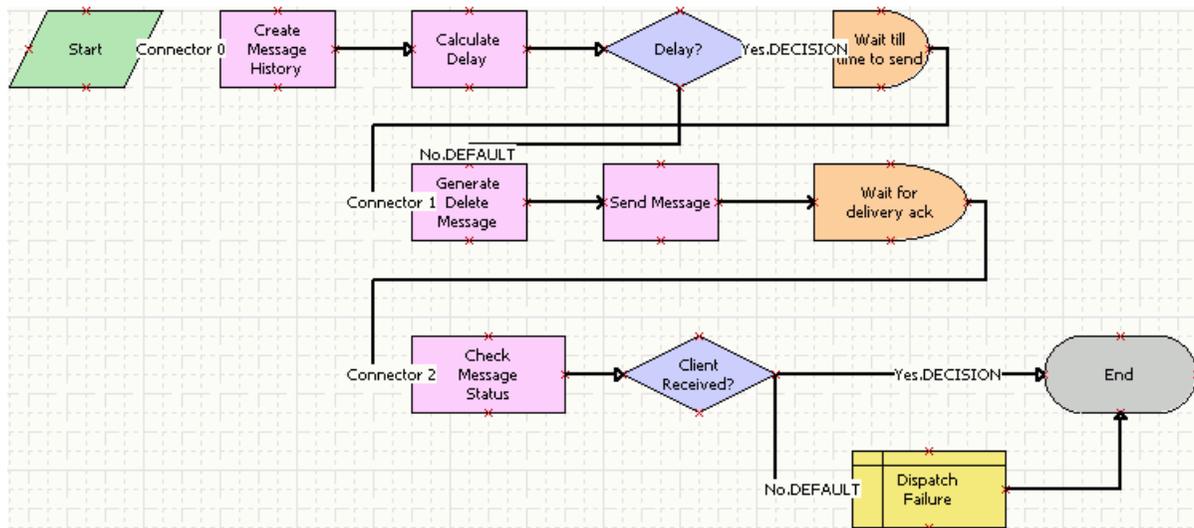


Figure 14. SMQ Outbound Delete in Action Business Object

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 89](#)
- [“Deleting a Record from Another Business Object” on page 89](#)
- [“Deleting a Record from Another Business Component and Business Object” on page 90](#)

## 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, set the MessageObject 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, set the MessageObject 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.



# 6

## Store-and-Forward Messaging for Handheld Business Services

This chapter lists Business Services used in Siebel Store-and-Forward Messaging for Handheld. It includes the following topics:

- “SMQ Message Service” on page 93
- “SMQ Transport Service” on page 94

**NOTE:** Some business service methods relate to specific message types. For information about message types, see “About Message Types” on page 64.

### SMQ Message Service

This business service consists of the main functions for Store-and-Forward Messaging. Table 12 lists the SMQ Message Service methods, and their definitions.

Table 12. SMQ Message Service Methods

Method	Definition
CheckMsgStatus	Queries the status field of a given message.
CreateMessageHistory	Creates a new record in the message history view.
FormatMsg	Formats a message to the proper XML format.
GenDataMsg	Creates a data message based on a message object.
GetTimeout	Calculates any of the three types of time out parameters for a given message object.
PingDevice	Sends a Ping message to check whether the device is within coverage.
ProcessUserQueue	Used primarily in the inbound processing. It processes all available inbound messages for a given user inbound queue.
RegisterMobileDevice	Handles device registration when a registration message is received.
RunWorkflowForTeam	Executes a workflow process for a set of users.
SendMsg	Sends a properly formatted message to the SMQ transport service.
UpdateMsgStatus	Changes a message status to a given value.

## SMQ Transport Service

This business service provides an interface between Oracle’s Siebel applications and the IBM WebSphere MQe Everyplace queuing system. [Table 13](#) lists the SMQ Transport Service methods and their definitions.

Table 13. SMQ Transport Service Methods

Method	Definition
GetQueues	Gets a list of user inbound queues from the queuing system.
DispatchMessage	Moves a message from an application inbound queue to a user inbound queue.
PeekMessage	Retrieves a copy of the first message on a queue with a unique message Id. When called more than once with no RemoveMessage in between, it returns the same message with the same message Id.
ReceiveMessage	Retrieves the first message on a queue, the message is removed from the queue.
RemoveMessage	Removes the first message on a queue. Takes the message Id as an input argument and removes only the message identified by that Id. If the message identified no longer exists, RemoveMessage does not operate.
SendMessage	Sends a message to an outbound queue representing a client device. Provides guaranteed delivery. If SendMessage returns SUCCESS, it is assured that the client will receive the message.
WaitForMessage	Waits for a message on an application inbound queue.

# Index

## A

### administering views

- responsibilities and views, updating 72
- Store-and-Forward Messaging, adding settings to enable 73

### administration

- administering applications, process of 60
- administration screens, about 59
- Administration-Store-and-Forward screen views 17
- application, enabling 60
- Applications view, about 18
- data messages, administering 60
- Device Administration view 18
- message history views 19
- message history, reviewing 62
- message objects, components, and fields, about automatically populating 61
- message objects, components, and fields, about manually populating 61
- Message Timeout view 18
- outbound data message timeout Parameters, setting up 28
- registered handheld device, examining information about 61

### Administration-Store-and-Forward screen

- about and views 17
- Applications view 18
- Device Administration view 18
- message history views 19
- Message Timeout view 18

### alert sounds, adding 70

### application

- enabling for Store-and-Forward Messaging 60

### application find, adding 71

### application parameters

- tuning 47

### Applications view, about 18

### architecture

- logical architecture diagram 13
- server components 15

### Asynchronous Query

- icon, about 19
- overview 20
- performing 20

### asynchronous query (server find)

- adding 76

### Asynchronous Query Results applet

- creating 77

### Asynchronous Query Results view

- creating 78

### Asynchronous Query, configuring

- Asynchronous Query Results applet, creating 77
- Asynchronous Query Results view, creating 78
- Find fields, adding 76
- Find object, configuring 74
- Find views, adding 76
- Hidden Screens view, adding new views 78
- new applet, creating 80
- new view and responsibilities, adding 79
- new view, creating 80
- process tasks 74
- unsupported business components, configuring 79

## B

### business components

- new applet, creating 80
- new view, creating 80
- unsupported business components, about configuring 79

### Business Object 48

### business services

- SMQ Message Service, methods and definitions 93
- SMQ Transport Services, methods and definitions 94

## C

### client APIs 17

### client inbound queue 17

### client modules

- about 17

### client outbound queue 17

### client-side checklist

- client-side common error messages 56

### client-side checklist, troubleshooting 54

### Companion Synchronization

- support of 12

### compiling changes 72

**component groups**  
 enabling 23  
**component groups, enabling** 24  
**components**  
 populating manually/automatically 61  
**Configuring** 48  
**configuring**  
 See installing and configuring handheld application  
**configuring associated business components** 48  
**connectivity management module** 17

**D**

**data messages**  
 administering 60  
 Applications view, about using 18  
 message objects, components, and fields,  
 about manually populating 61  
 message objects, manually/automatically  
 populating 61  
**delete transactions, workflows for** 86  
**Device Administration view**  
 about and fields 18  
**device registry**  
 tuning parameters 47  
**Direct Server Synchronization mode**  
 support of 12  
**drop-down menu**  
 making screen appear in menu 78

**E**

**Enterprise Application Integration component group**  
 about enabling 23  
**error messages**  
 client-side common error messages 56  
**error messages, common server-side** 52

**F**

**Field Service component group**  
 about enabling 23  
**Field Technician - Handheld**  
 about adding 25  
**fields**  
 manually/automatically populating 61  
**Find object**  
 adding 71  
 configuring 74  
 Find fields, adding 76  
 Find views, adding 76  
**functionality** 12

**H**

**handheld applications**  
 See Siebel Handheld applications  
**Handheld Synchronization component group**  
 about enabling 23  
**Hidden Screens view**  
 adding new view 78

**I**

**IBM MQe queue manager**  
 deploying, about 31  
 MQe servlet, installing 32  
 SMQ subsystems, creating additional 35  
 SMQReceiver, deploying on UNIX  
 systems 35  
 SMQReceiver, enabling 34  
**IBM WebSphere MQe**  
 about and handheld applications 17  
**icons, described** 19  
**inbound messages**  
 icon, about 19  
 message history views, about 19  
 message history, reviewing 62  
**inbound server message protocol** 64  
**inbound workflows, list of** 16  
**information**  
 entering application information 25  
**installing and configuring handheld application**  
 application parameters, tuning 47  
 client-side checklist, troubleshooting 54  
 client-side common error messages 56  
 network connection, setting up 46  
 process tasks 44  
 server-side checklist, troubleshooting 49  
 server-side common error messages 52  
 setup.ini file, setting up for handheld  
 applications 74  
 setup.ini parameters, verifying and  
 editing 44  
 Siebel Handheld applications, about  
 installing 46  
 synchronizing and registering 47  
**installing Store-and-Forward Messaging component groups, enabling** 23, 24  
 information, entering 25  
 internationalization, testing for 31  
 Message Delivery Timeout parameter,  
 configuring 27  
 messaging workflows, configuring 30  
 prerequisites, about 23  
 process tasks 23  
 responsibilities, about adding 25

- Store-and-Forward Messaging Message Objects, creating 26
- workflows, activating and reloading runtime events 30
- internationalization, testing for** 31

**L**

- language**
  - about creating an application for 31
- local find, adding** 76
- logical architecture**
  - diagram 13
  - server components 15

**M**

- Message Administration screen, about** 59
- Message Delivery Timeout parameter, configuring** 27
- message history**
  - reviewing 62
  - views, about 19
- message objects**
  - configuring Message Delivery Timeout parameter 27
- Message Repository screen, about** 59
- Message Service methods** 15
- Message Timeout view, about** 18
- messages**
  - data messages, administering 60
  - message objects, manually/automatically populating 61
  - outbound data message timeout parameters, setting up 28
- messaging workflows, configuring** 30
- mobile users**
  - using Device Administration view 18
- MQe queuing system**
  - SMQ Transport Service 16
- MQe servlet, installing** 32

**N**

- network connection, setting up** 46

**O**

- outbound data message timeout parameters**
  - setting 28
- outbound messages**
  - icon, about 19
  - message history views, about 19
  - message history, reviewing 62
- outbound workflows, list of** 16

**P**

- Pocket PC device**
  - application, enabling before synchronization 60
  - Connection Manager, about using 17

**Q**

- QueueInfo column**
  - about testing for internationalization 31

**R**

- registering**
  - examining registration information 61
  - handheld application and synchronizing 47
- registry settings, tuning** 47
- responsibilities**
  - new view and responsibilities, adding 79
  - updating and views 72
- responsibilities, about adding** 25
- runtime events**
  - reloading and activating workflows 30

**S**

- server components**
  - diagram 15
  - inbound workflows 16
  - outbound workflows 16
  - SMQ Message Service methods 15
  - SMQ Receiver 16
  - SMQ Transport Service 16
- server find, adding** 76
- server inbound queues**
  - about SMQ Receiver Service 16
- Service handheld application**
  - See Siebel Service Handheld application
- setup.ini file**
  - handheld applications, setting up for 74
- setup.ini parameters, verifying and editing** 44
- Siebel Client Registry settings**
  - tuning 47
- Siebel Handheld** 9
- Siebel Handheld administration views**
  - SMQ User responsibility 25
- Siebel Handheld applications**
  - administering applications, process of 60
  - application parameters, tuning 47
  - Asynchronous Query Results applet, creating 77
  - Asynchronous Query Results view, creating 78
  - Asynchronous Query, process of

- configuring 74
    - Asynchronous Query, using with 20
    - client-side checklist, troubleshooting 54
    - client-side common error messages 56
    - Find fields, adding 76
    - Find object, configuring 74
    - Find views, adding 76
    - Hidden Screens view, adding new views 78
    - IBM WebSphere MQe, about using 17
    - installing and configuring tasks, list of 44
    - installing, about 46
    - Message History view 62
    - message history, reviewing 62
    - network connection, setting up 46
    - new applet, creating 80
    - new view and responsibilities, adding 79
    - new view, creating 80
    - registered handheld device, examining
      - information about 61
    - server-side checklist, troubleshooting 49
    - server-side common error messages 52
    - setup.ini file, setting up 74
    - setup.ini parameters, verifying and editing 44
    - SMQ subsystems, creating additional 35
    - Store-and-Forward Messaging, about
      - configuring for 19
    - synchronizing and registering 47
    - unsupported business components, configuring 79
  - Siebel Handheld applications, setting up**
    - application find, adding 71
    - compiling changes 72
    - preconfiguring an application 69
    - process of setting up 68
    - Store-and-Forward Messaging views, adding 70
    - toolbar, adding 71
    - user properties for alert sounds, adding 70
  - Siebel Messaging and Queuing methods** 15
  - Siebel mobile products** 9
  - Siebel Mobile Web Client and Siebel mobile products** 9
  - Siebel Remote**
    - Siebel mobile products, about 9
  - Siebel Server**
    - IBM MQe queue manager, about
      - deploying 31
    - server-side checklist, troubleshooting 49
    - server-side common error messages 52
  - Siebel Service Handheld application** 19
    - Field Technician - Handheld responsibility 25
    - Store-and-Forward Messaging, about
      - configuring for 19
      - Store-and-Forward Messaging, using with 19
  - Siebel Wireless** 9
  - SMQ Message Service**
    - about 15
    - methods and definitions 93
  - SMQ Outbound Activity to One Recipient**
    - configuring 30
    - diagram and input arguments 82
    - workflow, about 16
  - SMQ Outbound Activity to Primary Owned By**
    - configuring 30
    - diagram and input arguments 83
    - workflow, about 16
  - SMQ Outbound Activity to Team**
    - configuring 30
    - diagram and input arguments 84
    - workflow, about 16
  - SMQ Process User Queue**
    - about 16
    - configuring 30
    - diagram and input arguments 85
  - SMQ Receiver Service** 16
  - SMQ subsystems**
    - creating additional 35
  - SMQ Transport Service**
    - about 16
    - methods and definitions 94
  - SMQ User responsibility, about adding** 25
  - SMQReceiver**
    - deploying on UNIX systems 35
    - enabling 34
  - sounds**
    - adding alert user properties 70
  - Store-and-Forward Message History, about** 19
  - Store-and-Forward Messaging Message Objects, creating** 26
  - Store-and-Forward Messaging views**
    - adding to the application 70
  - Store-and-Forward User Message History, about** 19
  - synchronizing**
    - handheld application and registering 47
    - Store-and-Forward Messaging Message Objects, creating 26
  - system administrator**
    - responsibilities 25
- T**
- timeout**
    - Message Timeout view, about 18
    - outbound data message timeout parameters, setting up 28

**timeout, Message Delivery Timeout parameter** 27

**toolbar**

about and icons 71  
 application, adding to 71  
 Siebel Tools .cfg file, modifying 69

**transport service** 16

**troubleshooting**

client-side checklist, troubleshooting 54  
 client-side common error messages 56  
 server-side checklist 49  
 server-side common error messages 52

**U**

**UNIX systems, deploying SMQReceiver on** 35

**User Administration screen, about** 59

**user properties**

adding for alert sounds 70

**V**

**views, administering**

responsibilities and views, updating 72  
 Store-and-Forward Messaging, adding settings to enable 73

**W**

**.wav file, specifying** 70

**wireless network**

about connecting using 46

**Workflow Management component group**

about enabling 23

**workflows**

activating and reloading runtime events 30  
 inbound 16  
 messaging workflows, configuring 30  
 outbound, list of 16  
 sample workflows, about 81  
 SMQ Delete FS Invoice Record workflow 87  
 SMQ Outbound Activity to One Recipient, about 82  
 SMQ Outbound Activity to Primary Owned By, about 83  
 SMQ Outbound Activity to Team, about 84  
 SMQ Outbound Delete in Action Business Object workflow 88  
 SMQ PreDelete FS Invoice Record workflow 87  
 SMQ Process User Queue, about 85  
 transaction delete workflows overview 86  
 transaction delete workflows, about 86  
 transaction delete workflows, configuring custom 88

