



Siebel Pharma Handheld Guide

Siebel Innovation Pack 2014
November 2104

ORACLE®

Copyright © 2005, 2014, Oracle. All rights reserved.

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

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

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

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

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

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

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

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

Contents

Chapter 1: What's New in This Release

Chapter 2: Overview of Siebel Pharma Handheld

| | |
|---|----|
| Siebel Mobile Products | 17 |
| About Siebel Handheld Applications | 19 |
| Overview of Siebel Handheld Application Synchronization | 20 |
| Synchronization Methods and Architecture for Siebel Handheld Applications | 21 |
| Overview of Siebel Handheld Client Configuration | 23 |

Chapter 3: Developing Siebel Handheld Applications

| | |
|---|----|
| Overview of Siebel Handheld Application Architecture | 26 |
| Unsupported Features for Siebel Handheld Client | 26 |
| About Defining User Functionality in the Siebel Handheld Client | 28 |
| Configuration Guidelines for the Siebel Handheld Client | 28 |
| About Creating Siebel Handheld Application Projects | 30 |
| Designing Screens and Views for Siebel Handheld Applications | 32 |
| Working With Different Handheld Device Specifications | 35 |
| Configuring User Interface Elements in the Siebel Handheld Client | 36 |
| Configuring List Applets for the Siebel Handheld Client | 37 |
| Configuring Fields to Display as Pop-Up Text Areas in List Applets | 38 |
| Configuring Alpha Tabs in List Applets for the Siebel Handheld Client | 38 |
| Configuring Form Applets for the Siebel Handheld Client | 39 |
| Configuring Form UI Applets for the Siebel Handheld Client | 40 |
| Using Auto Pop-Up Lists in the Siebel Handheld Client | 41 |
| Configuring Tree Applets for the Siebel Handheld Client | 42 |
| About Home Page Applets in Siebel Handheld Applications | 44 |
| About Buttons in Siebel Handheld Applications | 44 |
| About Multi-Value Groups in Siebel Handheld Applications | 44 |
| About the Menu Bar in Siebel Handheld Applications | 45 |
| About Toolbars in Siebel Handheld Applications | 45 |
| About the Status Bar in the Siebel Handheld Application | 46 |

- About Hyperlinks in the Siebel Handheld Application 46
- Using the COM Interface with Siebel Handheld Applications 46
- Configuring Calendar Settings 46
 - User Properties for Calendar Settings 47
 - Configuring Calendar Color Codes and Descriptions 47
 - Configuring the Goto View Button 48
- Configuring Calendar Buttons 49
 - Dynamic Drilldown from the Daily Calendar View to Med Ed Event 51
- Configuring Find Enhancements 53
 - Configuring Best Times for Enhanced Find 54
 - Creating New Class Definitions 54
 - Creating Pick Applets 55
 - Creating or Modifying Pick Lists 57
 - Modifying the Account Business Component 60
 - Extracting Non UI Fields During Synchronization 62
 - Modifying Applets 62
 - Modifying Business Components for Extracting Popup Applets 74
 - Extracting Contact Find and Account Find Popup Applets 75
 - Setting Key Values in the Auxiliary Configuration File 76
- Developing Tasks for Task UI in Siebel Handheld Applications 77
 - Designing a Task for Siebel Handheld Applications 77
 - Creating a Task for Siebel Handheld Applications 78
 - Example of Creating a Task for Siebel Handheld Application 79
- Scripting in the Siebel Handheld Application 80
- Compiling the Siebel Handheld Application 81
- Installing and Setting Up End User Translation Functionality 81
 - Performing Setup for Initial String Translation 82
 - Distributing Translated Strings 83
- Configuring Printing from the Siebel Handheld Application 83
 - Defining Documents for Siebel Handheld Applications 84
 - Designing Print Applets for Siebel Handheld Applications 84
 - Configuring Print Applets for the Siebel Handheld Application 85
 - Configuring Print Buttons for the Siebel Handheld Application 85
 - Creating Print Templates for the Siebel Handheld Application 86
 - Configuring a Printer Settings Dialog Box for the Siebel Handheld Application 86
 - Configuring Multiple Form and Multiple Quantity Printing for the Siebel Handheld Application 87
- Configuring Siebel Handheld Applications to Open Other Applications 88
- Configuring Signature Capture Capabilities in Siebel Handheld Applications 89

| | |
|---|----|
| Enabling Signature Capture | 89 |
| Invoking Signature Capture Dialog Box and Display | 90 |
| Signature Capture Dialog Box and Signature Display | 91 |
| Example of Configuring Signature Capture in Activities More Info View in Siebel Handheld Applications | 91 |
| Adding a Signature Capture Button in Siebel Tools | 93 |
| User Properties for Signature Capture Applets | 94 |
| Testing the Siebel Handheld Application | 94 |

Chapter 4: Siebel Handheld Application Administration

| | |
|---|-----|
| Overview of Siebel Handheld Application Administration Screens | 97 |
| Setting Up Users in Siebel Handheld Applications | 99 |
| Assigning User Responsibilities to Siebel Handheld Application Users | 99 |
| Creating Siebel Handheld Application Views | 99 |
| Specifying Views for the Siebel Handheld Application | 100 |
| About Business Component Filters | 100 |
| About Business Object Filters | 102 |
| About Thread Throttling in Siebel Handheld Applications | 103 |
| About Settings in the Administration Screens of Siebel Handheld Applications | 104 |
| Configuring Application Settings for Siebel Handheld Applications | 105 |
| Configuring Component Level Settings for Siebel Handheld Applications | 111 |
| Administering Barcode Settings for Siebel Handheld Applications | 112 |
| Enabling Barcode Scanning in Siebel Handheld Applications | 112 |
| Setting New Record Behavior for Barcode Scans | 113 |
| Mapping a Button for Socket In-Hand Scan Card | 114 |
| Creating New Barcode Definitions for Siebel Handheld Applications | 114 |
| Enabling Applications Views for Barcode Scanning | 119 |
| Enhancing the Behavior of the Barcode Scan | 119 |
| Example of Creating a New Barcode Definition in the Siebel Handheld Application | 122 |
| About BatchSync for Siebel Handheld Applications | 124 |
| Process for Administering Batch Sync for Siebel Handheld Applications | 125 |
| Enabling BatchSync for Siebel Handheld Applications | 125 |
| Downloading a Database Extraction | 126 |
| Reviewing Batch Synchronization Status | 127 |
| Running a Batch Extract | 129 |
| Running a Batch Transaction Process | 129 |
| Load Balancing Support for Synchronization | 130 |
| Setting the ApplicationDirectory Parameter | 131 |

- Enabling Encryption for Siebel Handheld Applications 131
 - Enabling Application Lockout for a Siebel Handheld Application 132
 - Enabling Authentication for a Siebel Handheld Application 132
 - Enabling Local Database Encryption for a Siebel Handheld Application 133
- Storing Asynchronous Data 133
- About Monitoring Synchronization Sessions for Siebel Handheld Applications 134
 - Monitoring Synchronization Sessions for Siebel Handheld Applications 135
 - Obtaining Error Details from Synchronization Sessions in Siebel Handheld Applications 136
 - Obtaining Business Component Information from Synchronization Conflicts in Siebel Handheld Applications 136
 - Obtaining an Audit Trail for Synchronization Sessions in Siebel Handheld Applications 137
 - Responding to Synchronization Errors in Siebel Handheld Applications 137
- Organization Enabled LOV Support 138
 - Enabling Organization LOV Support in the Siebel Handheld Client 139
 - Example of Configuring, Enabling, and Using Multiple Organization LOVs in Handheld Applications 139
- Process of Administering Siebel Pharma Handheld Applications 142
 - Managing Sample Records in Siebel Pharma Handheld Applications 143
 - Administering Disclaimer Text in Siebel Pharma Handheld Applications 144
 - Configuring Distributor Names in Siebel Pharma Handheld Applications 144
 - Performing a Signature Audit in Siebel Pharma Handheld Applications 145
 - Verifying Signature Capture in Siebel Pharma Handheld Applications 146
 - Configuring User Properties for Closing Inventory Periods in Siebel Pharma Handheld Applications 148
 - Configuring User Properties for Signature Capture Verification in Siebel Pharma Handheld Applications 149
 - Validation Logic of the Sign and Submit Buttons in Siebel Pharma Handheld Applications 150
 - Validating the DEA Number in Siebel Pharma Handheld Applications 157
 - Preparing Receipts for Calls in Siebel Pharma Handheld Applications 157

Chapter 5: Data Filtering for Siebel Handheld Applications

- Developing Data Filters for Siebel Handheld Applications 159
- Constructing Business Component and Business Object Filters for Siebel Handheld Applications 160
- Business Object Filters for Siebel Handheld Applications 161
- Recommendations for Managing Business Object Filters for Siebel Handheld Applications 161

| | |
|--|-----|
| Companion Synchronization Deployments | 162 |
| Direct Server Synchronization Deployments | 162 |
| Examples of Default Business Object Declarations | 162 |
| Designating Visibility | 164 |
| Overriding the Popup Visibility | 165 |
| Multiple Visibility Methods | 165 |

Chapter 6: Deploying Siebel Handheld Applications

| | |
|---|-----|
| Overview of Deployment for the Siebel Handheld Application Client | 167 |
| Installing the Siebel Server for Direct Server Synchronization | 169 |
| Installing the Components for Direct Server Synchronization Using the Siebel Server Installer | 169 |
| Siebel Server Component Parameters for Direct Server Synchronization | 170 |
| Configuring the Siebel Server for Direct Server Synchronization | 171 |
| Configuring Logging Levels for Direct Server Synchronization | 171 |
| Configuring Logging Levels for Companion Synchronization | 172 |
| Optimizing Siebel Server Process Management for Direct Server Synchronization | 173 |
| Performance and Scalability of Synchronization for Handheld Applications | 173 |
| Providing Synchronization Security with Secure Sockets Layer | 174 |
| Process of Installing the Siebel Handheld Application | 175 |
| Editing the Siebel Handheld Application Installation Configuration File | 175 |
| Enabling Back Up of Siebel Handheld Application Database | 180 |
| Installing Print Templates in the Siebel Handheld Install Directory | 181 |
| Setting Up Siebel Handheld Application Installation on a CompactFlash Card | 182 |
| Changing the SyncURL on Handheld Devices | 182 |
| Installing the Siebel Handheld Application from External Media | 183 |
| Distributing Siebel Handheld Application Patches | 183 |
| Deploying Patches to Handheld Applications with Direct Server Synchronization | 184 |
| Deploying Handheld Application Patches with Companion Synchronization | 186 |
| Creating Handheld Application Upgrade Kits for Companion Synchronization PDA | 188 |
| Deploying Attachments to Handheld Applications with Direct Server Synchronization | 188 |
| Deploying Handheld Application Attachments with Companion Synchronization | 190 |
| Downloading Patches to the Handheld Device | 192 |
| Setting up PatchAgent on Windows 2003 | 192 |
| Using PatchAgent to Distribute New Print Templates to the Handheld Device | 193 |

Chapter 7: Synchronization Conflict Handling and Recovery with Siebel Handheld Applications

| | |
|---|-----|
| Preventing Synchronization Transaction Conflicts with Handheld Applications | 195 |
| Extended Pick Processing in Siebel Handheld Applications | 195 |
| Insert Failure Processing in Siebel Handheld Applications | 198 |
| Recovering Error Data in Siebel Handheld Applications | 199 |

Chapter 8: Installing the Siebel Handheld Application

| | |
|---|-----|
| Process of Installing, Reinstalling, or Upgrading a Siebel Handheld Application | 201 |
| Uninstalling the Siebel Handheld Application | 201 |
| Installing the Siebel Handheld Application | 202 |

Chapter 9: Working with Siebel Handheld Applications

| | |
|---|-----|
| Components of the Siebel Handheld Application Interface | 205 |
| Navigating the Siebel Handheld Application Interface | 207 |
| Navigating a List of Records in the Siebel Handheld Application | 208 |
| Entering Data with the Siebel Handheld Application | 209 |
| Finding and Querying Data in the Siebel Handheld Application | 210 |
| About Find in Siebel Handheld Applications | 210 |
| About Queries in Siebel Handheld Applications | 210 |
| Printing with the Siebel Handheld Application | 210 |
| Exporting Data with the Siebel Handheld Application | 211 |
| Setting User Preferences in the Siebel Handheld Application | 212 |
| Backing Up and Restoring Data with the Siebel Handheld Client | 212 |
| Synchronizing Data on Your Handheld Device | 214 |
| Using Synchronization with the Siebel Handheld Application Client | 214 |
| Using Companion Synchronization with the Siebel Handheld Client | 215 |
| Using Direct Server Synchronization with the Siebel Handheld Application Client | 217 |
| About Synchronization Options | 218 |
| Troubleshooting Synchronization with the Siebel Handheld Application Client | 219 |
| Guidelines for Improving Siebel Handheld Application Performance | 220 |

Chapter 10: Using Siebel Pharma Handheld

| | |
|--|-----|
| Logging in to the Siebel Pharma Handheld Application | 226 |
| Viewing the Homepage for Siebel Pharma Handheld | 226 |

| | |
|--|-----|
| About the Contacts Screen | 227 |
| Managing Contacts with Siebel Pharma Handheld Applications | 227 |
| Adding Contacts in Siebel Pharma Handheld | 228 |
| Adding Contact Addresses in Siebel Pharma Handheld | 229 |
| Adding and Editing License Information in Siebel Pharma Handheld | 230 |
| Specifying the Best Time to Call on a Contact in Siebel Pharma Handheld | 230 |
| Adding Shared Notes About a Contact in Siebel Pharma Handheld | 231 |
| Indicating Affiliations Between Contacts and Accounts in Siebel Pharma Handheld | 231 |
| Viewing Prescription Data in Siebel Pharma Handheld | 232 |
| Viewing and Creating Alerts with Siebel Pharma Handheld Applications | 233 |
| Viewing Events in Siebel Pharma Handheld | 233 |
| Viewing and Creating Assessments in Siebel Pharma Handheld | 234 |
| About the Accounts Screen | 235 |
| Managing Accounts with Siebel Pharma Handheld Applications | 235 |
| Adding Accounts in Siebel Pharma Handheld | 236 |
| Adding Account Addresses in Siebel Pharma Handheld | 236 |
| Specifying Best Times to Call Accounts in Siebel Pharma Handheld | 237 |
| Adding Notes About Accounts | 238 |
| Indicating Affiliations Between Accounts and Professionals in Siebel Pharma Handheld | 238 |
| Viewing Sales Data in Siebel Pharma Handheld | 239 |
| Viewing Product Focus in Siebel Pharma Handheld | 239 |
| Planning and Recording Calls in Siebel Pharma Handheld Applications | 241 |
| Scenarios for Calls in Siebel Pharma Applications | 242 |
| Call Management Procedures in Siebel Pharma Handheld Applications | 243 |
| Creating Contact Calls in Siebel Pharma Handheld | 244 |
| Recording Contact Calls in Siebel Pharma Handheld Applications | 247 |
| Recording a Contact Call for a Different Contact at the Time of Signature Capture | 252 |
| Creating Account Calls in Siebel Pharma Handheld | 253 |
| Recording Account Calls in Siebel Pharma Handheld Applications | 255 |
| Creating General Activities in Siebel Pharma Handheld Applications | 257 |
| Adding General Activities from Activities in Siebel Pharma Handheld | 257 |
| Adding General Activities Using Calendar in Siebel Pharma Handheld | 258 |
| Managing Samples in Siebel Pharma Handheld Applications | 259 |
| Scenario for Managing Samples in Siebel Pharma Handheld | 260 |

| | |
|---|-----|
| Keeping the Samples Records in the Server and the Siebel Pharma Handheld Application Synchronized | 261 |
| How Siebel Samples Work with Calls | 262 |
| About Inventory Periods in Siebel Pharma Handheld | 263 |
| About Establishing an Initial Inventory Count in Siebel Pharma Handheld | 263 |
| About Samples Views in Siebel Pharma Handheld | 266 |
| Creating a Samples Order in Siebel Pharma Handheld | 267 |
| Creating a Samples Transfer in Siebel Pharma Handheld | 268 |
| Acknowledging Receipt of a Samples Transfer in Siebel Pharma Handheld | 269 |
| Adjusting a Samples Transaction in Siebel Pharma Handheld | 270 |
| Setting Up MedEd Events in Siebel Pharma Handheld Applications | 271 |
| Creating a MedEd Event in Siebel Pharma Handheld | 271 |
| Inviting Professionals to MedEd Events Using Siebel Pharma Handheld | 272 |
| Creating Expense Reports in Siebel Pharma Handheld Applications | 273 |

Appendix A: Troubleshooting

| | |
|---|-----|
| Troubleshooting Siebel Handheld Application Installation | 275 |
| Troubleshooting Siebel Handheld Application Synchronization | 276 |
| Companion Synchronization Checklist | 276 |
| Direct Server Synchronization Checklist | 277 |
| Resynchronizing Problems | 278 |
| Synchronization Timeout Parameters | 278 |
| Siebel Handheld Application Log Files | 279 |
| Direct Server Synchronization Log and Output Files | 280 |
| Companion Synchronization Log and Output Files | 280 |
| Handheld Device Log | 281 |
| Using SQLTrace with Siebel Handheld Applications | 282 |
| End User Error Messages | 283 |
| Web Server Timeout Errors | 283 |
| Handheld Device Timeout Errors | 283 |
| Siebel Handheld Application Performance | 284 |

Appendix B: Screens and Views

| | |
|-------------------|-----|
| Screens and Views | 287 |
|-------------------|-----|

Appendix C: Business Components and Classes

| | |
|---|-----|
| Business Components for Siebel Pharma Handheld | 291 |
| Business Component Classes for Siebel Pharma Handheld | 297 |

- Applet Classes for Siebel Pharma Handheld 301
- Siebel Pharma Handheld Objects that Must Not Be Modified 308

Appendix D: User Properties

- User Properties for Siebel Pharma Handheld Applications 309

Appendix E: Print Tagging Language

- Overview of Print Tagging Language for Siebel Handheld Applications 333
- Print Mode Tags 334
 - Absolute Tag 334
 - Relative Tag 334
- Data Output Tags 335
 - Applet Tag 335
 - Cell Tag 337
 - Picture Tag 339
 - Static Text Tag 341
- Print Formatting Tags 341
 - Background Tag 341
 - Comment Tag 342
 - Divider Tag 343
 - DrawRect Tag 344
 - EndOfLine Tag 345
 - EndOfPage Tag 345
 - Footer Tag 346
 - Format Tag 346
 - Frame Tag 347
 - Header Tag 348
 - Landscape Tag 349
 - List Tag 349
 - LineTo Tag 351
 - Margin Tag 352
 - PageBreak Tag 353
 - Portrait Tag 353
 - Title Tag 354
- Get and Set Tags 354
 - GetCount Tag 355
 - GetDate Tag 356
 - GetDirect Tag 356
 - GetField Tag 357
 - GetRegistry Tag 358

| | |
|------------------------------------|-----|
| GetTime Tag | 359 |
| GetTotal Tag | 359 |
| SetVariable Tag | 360 |
| Control Positioning Tags | 361 |
| GetX Tag | 361 |
| GetY Tag | 362 |
| Offset Tag | 362 |
| SetX Tag | 363 |
| SetY Tag | 364 |
| Data Manipulation and Logic Tags | 365 |
| Concat Tag | 365 |
| Conditional Tags | 366 |
| sql (WHERE) Tag | 367 |
| Variable Parameter Tags | 369 |
| Using Variables in Print Templates | 369 |

Appendix F: Print Configuration Settings

| | |
|---|-----|
| Overview of Print Configuration Settings for Siebel Handheld Applications | 373 |
| Print Configuration Settings | 373 |
| MarginBottom | 374 |
| MarginLeft | 374 |
| MarginRight | 374 |
| MarginTop | 375 |
| PaperHeight | 375 |
| PaperSize | 375 |
| PaperWidth | 375 |
| PrinterBaudrate | 376 |
| PrinterCompressed | 376 |
| PrinterDensity | 376 |
| PrinterDither | 377 |
| PrinterDraftMode | 377 |
| PrinterFormFeed | 377 |
| PrinterHandshake | 378 |
| PrinterOrientation | 378 |
| PrinterPort | 379 |
| PrinterType | 379 |
| PrinterUserSelect | 380 |
| Default Printing Settings | 381 |

Appendix G: COM Extensibility

| | |
|---|-----|
| About COM Extensibility | 383 |
| Overview of the SCEC Framework | 383 |
| Receiving Event Notifications | 384 |
| Invoking Siebel Methods (Action Requests) | 386 |
| Obtaining and Using Action Request Interface References | 389 |
| Process of Implementing a SCEC | 390 |
| Obtaining the SCEC Support Files | 390 |
| Generating and Configuring a SCEC | 391 |
| Modifying SCEC Files | 392 |
| Deploying SCEC Files to a PDA Device | 401 |
| Configuring SCEC User Properties in Siebel Tools | 402 |
| Configuring SCEC Registry Settings | 403 |

Index

1

What's New in This Release

What's New in Siebel Pharma Handheld Guide, Siebel Innovation Pack 2014

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

NOTE: Siebel Innovation Pack 2014 is a continuation of the Siebel 8.1/8.2 release.

What's New in Siebel Pharma Handheld Guide, Version 8.1/8.2

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

2

Overview of Siebel Pharma Handheld

This chapter introduces Oracle's Siebel Pharma Handheld application. The following topics are covered in this chapter:

- "Siebel Mobile Products" on page 17
- "About Siebel Handheld Applications" on page 19
- "Overview of Siebel Handheld Application Synchronization" on page 20
- "Synchronization Methods and Architecture for Siebel Handheld Applications" on page 21
- "Overview of Siebel Handheld Client Configuration" on page 23

Siebel Mobile Products

Several mobile applications are included in the 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 disconnected client and supporting server-side software, respectively. These applications provide intelligent transaction routing, data synchronization, and Store-and-Forward Messaging (SFM) capabilities.

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

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

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

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

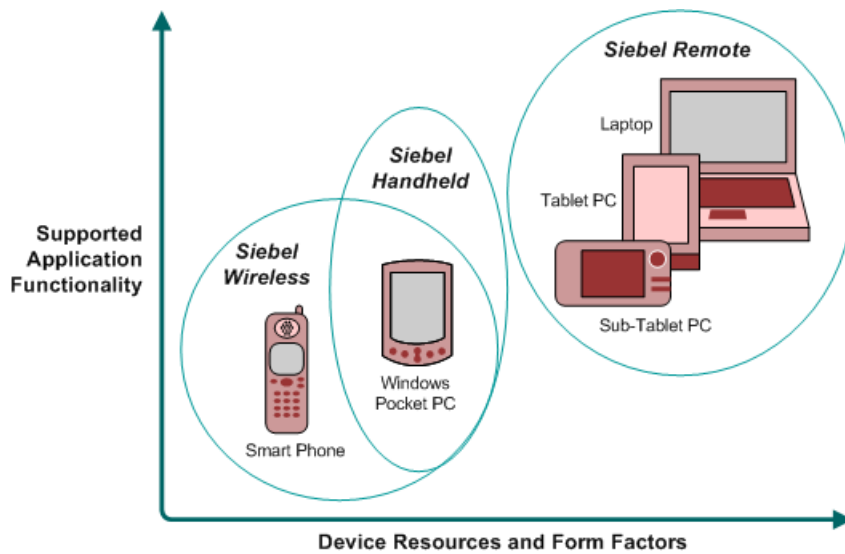


Figure 1. Siebel Mobile Product Platforms

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

Table 1. Siebel Mobile Product Software Components and Supported Devices

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

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

Table 2. Siebel Mobile Product Supported Applications and Related Documentation

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

About Siebel Handheld Applications

Oracle's Siebel Handheld applications provide a focused subset of functionality in comparison to the Siebel Web Client. The Siebel Handheld Application Client differs from the Siebel Web Client and the Siebel Mobile Web Client in that it uses a selected subset of views, accommodates smaller screen sizes, and has a reduced memory capacity. For a list of functional differences, see [Chapter 3, "Developing Siebel Handheld Applications."](#)

Some features that are available on the Siebel Mobile Web Client are not available on the Siebel Handheld application. Scripting and Siebel engines (for example, Pricing and Reporting) are not supported on Siebel Handheld applications. For a list of features that are not supported, see ["Unsupported Features for Siebel Handheld Client" on page 26.](#)

Some supported features of the Siebel Handheld Client are as follows:

- **Application Development with Siebel Tools.** Siebel Handheld applications can be configured in Siebel Tools using the Web Applet Designer. You can take your existing Siebel application and use this as the basis for the Siebel Handheld application, or you can create a new application by using the Siebel Tools application shipped on the Siebel Handheld application DVD. For more information about application development with Siebel Tools, see [Chapter 3, “Developing Siebel Handheld Applications.”](#)
- **Data filtering.** Filtering allows you to specify a subset of data to be used on the handheld device. Filtering is an important feature of a Siebel Handheld application deployment because of the relatively large size of the average enterprise database in comparison to the relatively limited memory capabilities of handheld devices. It is important that you allocate sufficient time in the project plan to create and test the handheld filters. For more information about data filtering, see [“Developing Siebel Handheld Applications” on page 25.](#)
- **Installation options.** The Siebel Handheld application can be installed on Windows-powered devices in one of two ways. The first is to create a partnership between a desktop or laptop PC and the handheld device using Microsoft ActiveSync. The second is to install the Siebel Handheld application from external media, for example, a Cabinet (CAB) file. Handheld application patches can be remotely deployed using PatchAgent. For more information about installation options, see [Chapter 8, “Installing the Siebel Handheld Application.”](#)
- **Patch updates.** Siebel Handheld applications automatically apply the latest software patches to the handheld device when needed. Companion Synchronization allows users to synchronize the Siebel Handheld application database to the local database on Companion PC and invokes Siebel Remote to synchronize the local database with the Siebel Server database.

PatchAgent automates the deployment of patches to handheld devices as part of the synchronization process. For more information about PatchAgent, see [“Installing the Siebel Handheld Application from External Media” on page 183.](#)
- **Backing up and restoring data.** The Siebel Handheld application provides database backup functionality. A database restore utility is also included with the application. For more information about database backup and restoration, see [“Enabling Back Up of Siebel Handheld Application Database” on page 180.](#)

Overview of Siebel Handheld Application Synchronization

The Siebel Handheld Application Synchronization (Sync) Client synchronizes data between the Siebel Handheld application database and the Mobile Web Client database or Siebel Server database.

[Figure 2](#) illustrates the Siebel Handheld synchronization infrastructure.

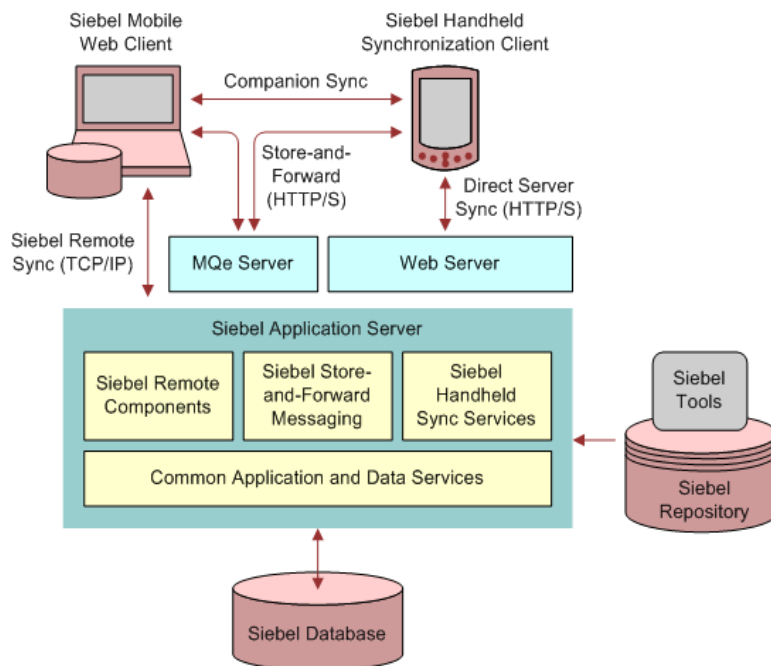


Figure 2. Siebel Handheld Synchronization Infrastructure

The synchronization process does the following:

- Updates the Siebel Server database with changes made to the Siebel Handheld application database
- Updates the database on the handheld device with changes made to the Siebel Server database
- Downloads metadata changes, such as object definitions for new or modified screens or views
- Allows users to select filters to limit the amount of data that is downloaded
- Automatically applies the latest software patches to the handheld device when needed

Synchronization Methods and Architecture for Siebel Handheld Applications

The following methods can be used to synchronize data on the handheld device with data in the Siebel Server database: Companion Synchronization (CS), Direct Server Synchronization (DSS) and Direct Server Synchronization by Proxy (DSSvP). The architecture of these methods is discussed in the following sections.

Handheld Application Synchronization

A Handheld Application Synchronization Server component that is running on the Siebel application server can accept concurrent synchronization requests from multiple synchronization clients. When connecting directly to a server, the Siebel Handheld Application Synchronization Client synchronizes its data directly to a server database. Companion Synchronization allows users to synchronize the Siebel Handheld application database to the local database on a companion PC device and invokes Siebel Remote to synchronize a local database with the Siebel Server database.

In Direct Server Synchronization mode, the Siebel Handheld Application Synchronization Client uses the HTTP (or HTTPS) protocol over a network connection to communicate with the Siebel Handheld Application Synchronization Server. Users must configure their handheld devices for communication over the desired network.

A companion synchronization server runs on a companion personal computer with a Mobile Web Client installation and supports synchronization with a data source specified in a client configuration; the database resides on the companion PC.

Companion Synchronization

Companion Synchronization allows users to synchronize the Siebel Handheld application and database with a Siebel Remote database on a Mobile Web Client.

Microsoft ActiveSync is used to connect the handheld device and the companion PC through a serial or USB connection. The user launches Siebel Handheld application synchronization from companion PC or the device. The user can change the default filters using Siebel Handheld application synchronization.

Direct Server Synchronization

Direct Server Synchronization allows users to synchronize their Siebel Handheld application and database directly with the Siebel Server. The user connects the handheld device to the Siebel application server through a network connection that uses the HTTP protocol. The connection runs through a IIS Web Server to the application server, where the synchronization server components are installed and running.

Direct Server Synchronization via Proxy

Direct Server Synchronization via Proxy (DSSvP) allows users to synchronize their Siebel Handheld application and database directly with the Siebel Server. The user connects the handheld device to the Siebel application server through ActiveSync and the network connection established by the companion PC. The connection runs through a IIS Web Server to the application server, where the synchronization server components are installed.

Overview of Siebel Handheld Client Configuration

Figure 3 illustrates the Siebel Handheld client application architecture.

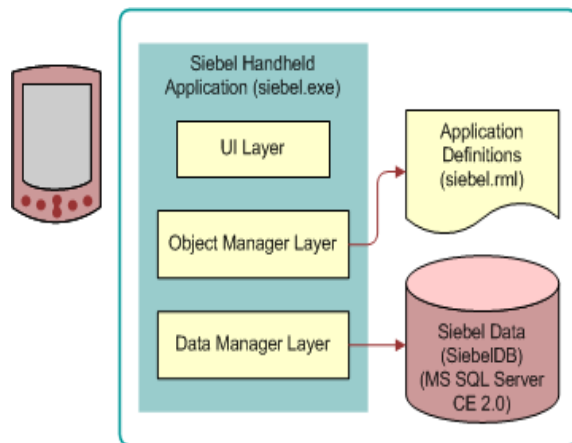


Figure 3. Siebel Handheld Client Application Architecture

The Siebel Handheld executable, siebel.exe, runs on the handheld client in a disconnected state, and includes the user interface, object manager, and data manager.

The siebel.exe accesses the siebel.rml file that contains the application definitions.

The siebel.rml file is the Siebel Handheld equivalent of the repository file used in desktop applications, because it provides the definitions of the Siebel Handheld screens, views, and applets.

The final component of the Siebel Handheld Client architecture is the Siebel Handheld application database. The Siebel Handheld application includes a relational database management system designed to run on the Microsoft Windows-powered platform.

Complete the following steps to configure your Siebel Handheld Client.

To configure the Siebel Handheld Client

1 Define the user functionality.

Determine which user interface elements (such as screens, views, toolbars, applets, find, and help) will be necessary for your Siebel Handheld Client users. For more information about defining user functionality, see [“About Defining User Functionality in the Siebel Handheld Client”](#) on page 28.

2 Configure the Siebel Handheld Client application user interface.

The process for configuring user interface elements in the Siebel Handheld Client is the same as that for other Siebel clients. For more information about configuring user interfaces for Siebel Handheld applications, see [“Configuring User Interface Elements in the Siebel Handheld Client” on page 36](#) and *Configuring Siebel Business Applications*.

3 Identify all eScript or VB script.

Siebel Handheld applications do not currently support scripting, including eScript or VB script. Scripting used in your Siebel application can conflict with your Siebel Handheld application configuration.

4 Compile the new configuration.

For more information about compiling the configuration, see [“Compiling the Siebel Handheld Application” on page 81](#).

5 Test the new configuration.

Test the configuration, repeating the previous steps, to address any configuration errors.

After completing the configuration for the Siebel Handheld Client, you are ready to install it on the handheld devices. To install the Siebel Handheld application configuration on a handheld device, see the instructions in [“Installing the Siebel Handheld Application” on page 202](#).

3

Developing Siebel Handheld Applications

This chapter describes how to set up Siebel Handheld applications. The following topics are covered in this chapter:

- [Overview of Siebel Handheld Application Architecture on page 26](#)
- [Unsupported Features for Siebel Handheld Client on page 26](#)
- [About Defining User Functionality in the Siebel Handheld Client on page 28](#)
- [Configuration Guidelines for the Siebel Handheld Client on page 28](#)
- [About Creating Siebel Handheld Application Projects on page 30](#)
- [Designing Screens and Views for Siebel Handheld Applications on page 32](#)
- [Working With Different Handheld Device Specifications on page 35](#)
- [Configuring User Interface Elements in the Siebel Handheld Client on page 36](#)
- [Using the COM Interface with Siebel Handheld Applications on page 46](#)
- [Configuring Calendar Settings on page 46](#)
- ["Configuring Calendar Buttons" on page 49](#)
- ["Configuring Find Enhancements" on page 53](#)
- [Developing Tasks for Task UI in Siebel Handheld Applications on page 77](#)
- [Scripting in the Siebel Handheld Application on page 80](#)
- [Compiling the Siebel Handheld Application on page 81](#)
- [Installing and Setting Up End User Translation Functionality on page 81](#)
- [Configuring Printing from the Siebel Handheld Application on page 83](#)
- [Configuring Siebel Handheld Applications to Open Other Applications on page 88](#)
- [Configuring Signature Capture Capabilities in Siebel Handheld Applications on page 89](#)
- [Testing the Siebel Handheld Application on page 94](#)

Overview of Siebel Handheld Application Architecture

The process for configuring a Siebel Handheld application has a few differences, described in this chapter, from the process for configuring other types of Siebel applications.

NOTE: Familiarize yourself with the Siebel application configuration process described in *Configuring Siebel Business Applications* before you attempt to configure the Siebel Handheld application.

The five layers in the Siebel Handheld application architecture are comparable to those used on the Siebel Web Client:

- **Application layer.** The application layer starts and closes the application.
- **User Interface layer.** The user interface layer renders the display and interacts with actions of the user.
- **Object Manager layer.** The object manager layer provides a consistent object behavior and interaction between all business objects within the application.
- **Data Manager layer.** The data manager layer maintains an object-oriented abstraction of the native data stored in the data repositories for the benefit of the object manager.
- **Database layer.** The database layer includes the data that Siebel Handheld application users access.

Your Administrator configures an application on the Siebel client using Siebel Tools and compiles the Siebel repository (.srf) file. During the synchronization process, information specific to the Siebel Handheld application is extracted from the Siebel repository file. This information is used to create a repository metalanguage (.rml) file, which is downloaded to the handheld device.

The repository metalanguage file contains the layout information for the handheld device but not the database layer. The database layer is generated in the transport file called dbfile.txt, which is downloaded and imported to the Siebel Handheld application database (siebelDB.sdf). For more information about Siebel Tools, see *Using Siebel Tools*.

Unsupported Features for Siebel Handheld Client

The differences between configuring user interface elements for the Siebel Web Client and the Siebel Handheld Client are shown in [Table 3](#).

Table 3. Configuring Siebel Client User Interface Elements

| Client | Siebel Tools | SWT Template Files | Specialized Algorithms |
|------------------------|--------------|--------------------|------------------------|
| Siebel Web Client | Yes | Yes | No |
| Siebel Handheld Client | Yes | No | Yes |

The following are not supported in Siebel Handheld applications:

- Chart and explorer applet types are not supported.
- Group boxes are not supported. For more information about group boxes, see [“Group Boxes” on page 40](#).
- Alarm Manager is not supported. Therefore, if an alarm check box is added to an applet and the check box is selected, no alarm sounds.
- The base time zone is determined by the system settings on the Siebel Server and cannot be changed on the handheld device.
- Siebel Browser and Siebel Server side scripting are not supported.
- Siebel Workflow is not supported.
- Siebel Personalization.

See the following topics, which outline, more specifically, other unsupported features.

Functions

Siebel Handheld applications support most of the calculated fields and operations used in the Siebel Web applications with a few exceptions. See the following list of unsupported functions.

- AccountId
- BCHasRows
- DockingNodeId
- DockingNodeName
- DockingNodeRouteId
- EAILookupExternal
- EAILookupSiebel
- EXISTS

NOTE: For example, the following expression on multi-value fields is not supported by Handheld Applications: `IF(EXISTS([Multi-valued Field Name] IS NOT NULL), 'Y', 'N')`.

- GetHQInstanceId
- GetHQInstanceName
- GetNumBCRows
- GetProfileAttr
- GetXAval
- IsDocked
- LanguageName
- LocaleName
- LookupMessage

- LookupTranslation
- NOT
- ParentFieldValue. This function works as a calculated field but not in, or with, a search specification.
- RepositoryId
- RepositoryName
- RowIdToRowIdNum
- ToolsLanguage

The NOT key word is not supported on Booleans. Use the != operator instead. For example, "Active" != "Y".

The RowIdToRowIdNum function is supported differently in Siebel Handheld applications than in the Web applications. In Web applications, RowIdToRowIdNum converts a row ID to a numeric value. In Siebel Handheld applications, the row ID itself is returned.

About Defining User Functionality in the Siebel Handheld Client

The first step in configuring a Siebel Handheld Client is to determine which user interface elements are necessary. Siebel Handheld Client users have different requirements from the Web Client users. Keep in mind the differences in display size, memory capacity, and input methods between handheld devices and larger computers, such as desktop computers.

The goal is to create applications that are easy for your users to use and which take advantage of the strengths of the handheld device platform. Because of the processing speed, memory limitations, and form factor differences, include only those user interface elements that are necessary for users to complete their job responsibilities.

Configuration Guidelines for the Siebel Handheld Client

This topic contains general guidelines to use when you configure the Handheld application. Use these guidelines when you design any new objects for the Siebel Handheld Client. This approach facilitates a logical separation of the Siebel Handheld Client user interface elements from the Siebel Web Client user interface elements.

Identify User Activities

Consider the following when identifying user activities:

- To conserve memory (and thereby improve performance) and ease navigation, identify business processes that are required by users and develop applications that support these processes.

- If more than one type of user needs a Siebel Handheld application, it is preferable to divide the application into multiple responsibilities rather than give all possible users access to all available screens and views. Responsibilities are fully configurable by the application developer.
- Do not include additional functionality that is not required to reside on the device.

Limit the Number of Screens and Views

Consider the following when introducing new screens and views:

- Keep the number of views in your application to 30 or fewer. Determine the critical business processes that the Siebel Handheld application will support and pick only the views that are necessary and are compliant with your business requirements.
- Limit the number of views within a screen to 12 or fewer to keep the Show drop-down list concise.
- Limit the number of screens to six or fewer to keep the Screen drop-down list concise.
- Keep screen names to about 15 characters or fewer, so that they fit in the Screen drop-down list. The number of characters is a general guide, because characters vary in width.
- Keep view names to about 30 characters or fewer so that they fit in the Show drop-down list. The number of characters is a general guide, because characters vary in width.

Limit the Number of Applets for Each View

Organize data in multiple applets in a single view to use Tabbed Views. This allows the user to access critical data via a single click.

Limit the Number of Columns and Fields for Each Applet to Minimize Scrolling

Consider the following when designing applets:

- Design applications that contain as few screen and view hierarchy levels as possible. In a Web-based application, you can have views with many applets, and the user toggles between the applets.
- Include the most important fields and columns that a user needs to see at a glance before scrolling. For example, if you have a form applet that is a parent, include only needed fields to avoid scrolling.
- If you are using a list applet, determine how wide it needs to be.
- Design each applet so that it only contains columns and fields that are required for end-user tasks.
- Consider moving required and editable fields into visible area of the applet where users can enter data and do not have to scroll. Because there is less screen space on a handheld device than on a laptop or desktop, the most important data must be immediately visible.
- The Date column can be formatted to show only the date instead of Date and Time where Time is not relevant. This saves the user from having to scroll.

- To minimize horizontal scrolling, limit the number of columns displayed in a list applet to no more than ten.
- There is no limit on the number of fields in form applets of single-applet views. However, to minimize scrolling, keep the number of fields to 20 or fewer.
- Use a form applet for the parent for each parent-child view.
- In a multi-applet view, limit the number of fields in form applets to five or fewer. Add additional fields only if the field width is short. For example a check box field.
- Reduce the number of fields if the fields are multi-line, for example, a Comments box that contains three lines of text.
- Do not include read-only check boxes in form applets. It is very difficult for users to discern that the check box is not editable.
- Do not add pick applets or set the Runtime property to true for Read-Only fields that have a picklist value associated to them. This will improve overall extraction time, synchronization time and dbfile.txt file size since the process will not be forced to extract all the picklist values for these fields.
- Keep query names to about 15 characters or fewer, so that they fit in the Queries drop-down list. For example, North American Organization is too long for a query name, so change it to a shorter name, such as N. American Org. The number of characters is a general guide because characters vary in width (for example, W is wider than i).
- For each screen, create a view called My *(asterisk) or All * as a single list applet view. For example, for the Activities screen, create a My Contacts view that has the Contacts list applet only.

Design Data Filters

Consider the following when designing data filters:

- Limit the size of the dbfile.txt file to less than three megabytes (MB). The RDBMS on the handheld device is approximately three times the size of dbfile.txt. If the data files are so large that they cannot be imported into the database with the available memory, users cannot successfully synchronize their data. For information about how to design data filters to limit the amount of data that is synchronized to the handheld device, see [Chapter 5, “Data Filtering for Siebel Handheld Applications”](#).
- When working with assessment creation functionality, you need to name your templates in such a way that they will filter only the needed templates that will be downloaded to the device. Without properly setting up filters, all templates will be downloaded to the device.

About Creating Siebel Handheld Application Projects

The Siebel Handheld application projects appear in the list of projects in Siebel Tools; when Project is selected in the Object Explorer, the list of projects is displayed in the Object List Editor.

Use a naming convention that allows you to identify applications, screens, views, and applets belonging to a particular Handheld application. This allows you to locate all the object definitions in a Handheld application by querying on the name in the Object List Editor. For example, all of the Handheld application screens, views, applets, toolbars, and menus use either an SHCE prefix (for example, SHCE Sales Account List view) or a CE suffix. Use the Object List Editor to query for views that contain the prefix SHCE or the suffix CE; doing so displays all of the Handheld application views.

If you create a new Siebel Handheld application, create a new suffix or prefix to identify the name of the application and each screen, view, and applet name. For example, you might use PPC_SHCE for an application.

Designing Screens and Views for Siebel Handheld Applications

Views in Siebel Handheld applications can display two or more applets simultaneously. You can configure screens and views to display applets in two modes, as described in [Table 4](#).

Table 4. Applet Display Modes

| Mode | Treatment of Applets | Maximum Number of Applets |
|------------------------|---|---|
| Traditional Two-Applet | <p>If the screen or view has only one applet, the applet uses the entire display area. If the screen or view has two or more applets, the first two applets in the frame array of a view definition are displayed, and the user toggles to see the other applets. The first applet is always displayed.</p> <p>Users can use the application-level menu to toggle between applets. You can also provide a toggle button as a visual cue to the user that there are additional applets. Add a toggle button to each applet except the first applet. Identify the button with a Toggle caption and set the <code>methodInvoked</code> property to <code>ToggleApplet</code>.</p> <p>For information about how to use the application-level menu, see the online help for the Siebel Handheld application.</p> | Two |
| Tabbed Applet | <p>Only one applet is displayed. All other applets are layered as tabs at the bottom or top of the display. Tabbed applet mode enables the user to maintain context and switch between applets easily. You can configure a context bar to show the parent applet information.</p> <p>Users can navigate to other applets in the following ways:</p> <ul style="list-style-type: none"> ■ Tap the applet. ■ Tap the tab that represents the applet. ■ Use the application-level menu to select the applet. For information about how to use the application-level menu, see the online help for the Siebel Handheld application. | <p>Six in portrait orientation and square screens.</p> <p>Eight in landscape orientation.</p> |

NOTE: Use only unique display names in the Screens and Show drop-down lists. The Siebel Handheld application uses the first screen or view display name that matches the selected name in the drop-down list. If two screens have the same display name, a search for the screen always returns the first display name found.

Configuring Tabbed Applets for the Siebel Handheld Client

The applet display mode is controlled by the application user property `DefaultViewType`. For more information about this user property, see [DefaultViewType User Property on page 311](#).

To override the effect of the `DefaultViewType` setting, set the value of the view-level user property `ViewType`. For more information about this user property, see [ViewType User Property on page 332](#).

To configure tabbed applet behavior for an application

- In Siebel Tools, set the value of the application user property `DefaultViewType`.

For more information about this user property, see [DefaultViewType User Property on page 311](#).

To configure tabbed applet behavior for a view

- 1 In Siebel Tools, set the value of the view-level user property `ViewType`.

For more information about this user property, see [ViewType User Property on page 332](#).

NOTE: `ViewType` overrides the `DefaultViewType` property in portrait orientation. In landscape orientation and on square screens, the default applet display mode is `Tabbed`.

- 2 If you do not want to show the context bar that displays the parent applet information at the top of a view, set the `DisableParentBar` user property to `TRUE`.

For more information about this user property, see [DisableParentBar User Property on page 313](#).

Screen Allocation for Traditional Two-Applet Mode

The amount of screen space available for applets is determined by the type of applets in the view. Two applets are stacked above one another. A parent form applet in a two-applet view displays multiple fields on a screen. Users use the scroll bar to navigate. The form applet dynamically resizes if there are fewer than five fields and, therefore, does not waste screen display space with empty lines. The maximum number of fields that are displayed at one time is configurable by setting the `Max Parent Applet Size` preference in the User Preferences dialog. For more information about user preferences, see [“Setting User Preferences in the Siebel Handheld Application” on page 212](#).

Drill-Down Only Views

Some views can be drill-down only views—that is, they can be accessed only by navigating from another view. These views are not accessible through the Show drop-down list. However, when the user navigates to the view, the view title appears in the Show drop-down list. To configure a drill-down only view, set the view title as you do for any other view and set the `ScreenMenu` property to `FALSE`. For more information about drill-down only views, see *Configuring Siebel Business Applications*.

Applet Focus Behavior

When navigating between views, sometimes you might need to override the default behavior for giving focus to an applet

The GotoView method is used to navigate from one view to another. A button control is added to the applet in the view you want to navigate from, the button's methodInvoked property is set to GotoView and the button's View property is set to the view that is to be created. When the GotoView method is executed, it builds the specified view. If the view includes a parent and child applet, GotoView gives focus to the child applet. This is the desired behavior in most instances when you are simply navigating from one view to another.

If the desired behavior is to navigate to a new view and, additionally, to create a new record, then you must use the GotoViewNewRecord method. This method calls two methods; first it calls GotoView, and then it calls NewRecord, which creates a new record in the new view.

In some instances the default behavior of these methods does not produce the intended behavior. For example, in the Orders view of the Service Requests screen, when you tap the Create New Order button, it takes you to the Order Details view, which contains a parent form applet (Service Request Order) and a child list applet (Service Order Entry Line Item). In this instance, when the Create New Order button's GotoViewNewRecord method is executed, you want a new order to be created.

However, the default behavior is to give focus to the child applet rather than creating a new record in the parent applet. Therefore, you need to override the default behavior of GotoView with the control user property OverrideDefaultApplet, and set the value to the applet that you want to give focus. For more information about this user property, see [OverrideDefaultApplet User Property on page 317](#). For more information about the View user property, see [View User Property on page 331](#).

For example, if you create a button labeled Create Service Request Order, you must define the user properties shown in [Table 5](#):

Table 5. User Properties

| Name | Value |
|-----------------------|------------------------------------|
| View | SHCE Service Order Line Items View |
| OverrideDefaultApplet | 0 |

Views with Associated Print Templates or Reports

Print templates are text files that you design and create for printing. After creating a print template, you associate it with an applet. When a user chooses to print from a view, the print template that is associated with the applet is automatically selected and used for printing. Follow these guidelines when creating print templates:

- The print template name for a particular display applet is registered in the Mail Template property in the Applet object associated with the applet.
- The template must be placed in the Templates directory.
- The template name that goes into the Mail Template property must not include a path name or extension. For example, if the full path of the template is \Program Files\Siebel Handheld\Templates\InvoiceTemplate.txt, enter only the root name, InvoiceTemplate, in the property field.
- The template file itself must have a .txt extension on the device.

For more information about printing and print templates, see the following:

- [“Configuring Printing from the Siebel Handheld Application” on page 83](#)
- [“Installing Print Templates in the Siebel Handheld Install Directory” on page 181](#)
- [“Printing with the Siebel Handheld Application” on page 210](#)
- [Appendix E, “Print Tagging Language”](#)
- [Appendix F, “Print Configuration Settings”](#)

Working With Different Handheld Device Specifications

Siebel Handheld applications can run on many handheld devices. The capabilities of handheld devices vary, depending on the following characteristics:

- Operating system (OS)
- Processor
- Storage memory
- Screen orientation
- Screen resolution

Siebel Handheld applications are aware of the devices on which they run. The applications do not need to be compiled specifically for particular devices. Handheld applications are device aware in the following respects:

- **OS:** Siebel Handheld applications can identify the OS of the device on which they are running.
- **Storage memory:** Siebel Handheld applications can determine the amount of storage memory available on the device on which they are running. Handheld applications use this information to determine if sufficient storage memory is available to download the dbfile.txt file.

Handheld applications do not download the file if available storage memory is insufficient. The applications request the user to apply more restrictive filters, or to make more memory available on the device.

- **Screen orientation:** Views in Siebel Handheld applications are displayed differently depending on the screen orientation of the device.
 - **Portrait orientation**—In this screen orientation one or two lists or forms are displayed on each screen. When the screen displays two lists or forms, the user can maximize and minimize the lists or forms.

- Landscape orientation and square screens—In this screen orientation one list or form is displayed on each screen. The user can use a tab and an icon button to show one list or form, and simultaneously hide the list or form that is currently displayed.

Dialog boxes and other UI elements are also aware of screen orientation. All UI controls are automatically placed in an area that is not affected by changes to screen orientation. Any fields that users can edit are automatically placed in a location that is not obscured when the Software Input Panel (SIP) is displayed.

These automatic features are not enabled in form UI applets and task applets. When you develop these applets for Siebel Handheld applications, you must manually lay out UI items so that they are not obscured by changes to screen orientation or SIP display.

- **Screen resolution:** Siebel Handheld applications can run on both high-resolution and low-resolution displays.

Configuring User Interface Elements in the Siebel Handheld Client

When configuring user interface elements on the Siebel Handheld Client, also consider the processing speed and memory constraints of handheld devices. The constraint on processing speed affects the performance of a handheld device. The performance of a handheld device is also related to the number of screens and views downloaded to the handheld device. Because handheld devices are not meant to provide the same functionality as larger devices, such as laptops, the number of screens and views must be kept to a minimum on the handheld device.

Due to size constraints on handheld devices, the Siebel Handheld Client displays user interface elements differently from the Siebel Web Client. Consider these differences when configuring the Siebel Handheld Client application. For example, Siebel Handheld Client does not support an alphabetical index.

The following topics are related to configuring user interface elements in the Siebel Handheld Client:

- [“Configuring List Applets for the Siebel Handheld Client” on page 37](#)
- [“Configuring Fields to Display as Pop-Up Text Areas in List Applets” on page 38](#)
- [“Configuring Alpha Tabs in List Applets for the Siebel Handheld Client” on page 38](#)
- [“Configuring Form Applets for the Siebel Handheld Client” on page 39](#)
- [“Configuring Form UI Applets for the Siebel Handheld Client” on page 40](#)
- [“Using Auto Pop-Up Lists in the Siebel Handheld Client” on page 41](#)
- [“Configuring Tree Applets for the Siebel Handheld Client” on page 42](#)
- [“Configuring Tabbed Applets for the Siebel Handheld Client” on page 33](#)
- [“About Home Page Applets in Siebel Handheld Applications” on page 44](#)
- [“About Buttons in Siebel Handheld Applications” on page 44](#)
- [“About Multi-Value Groups in Siebel Handheld Applications” on page 44](#)
- [“About the Menu Bar in Siebel Handheld Applications” on page 45](#)

- [“About Toolbars in Siebel Handheld Applications” on page 45](#)
- [“About the Status Bar in the Siebel Handheld Application” on page 46](#)
- [“About Hyperlinks in the Siebel Handheld Application” on page 46](#)

Configuring List Applets for the Siebel Handheld Client

You configure list applets and list columns for Siebel Handheld applications the same way you configure them for other Siebel applications.

Recommended Strategy for Configuring List Applets

The recommended strategy when configuring the Siebel Handheld application is to use the list applets for record navigation and to rely on form applets to provide the record details.

The following are additional guidelines to follow when designing list applets:

- Configure specific list applets for your Handheld application, rather than reusing Siebel Web Client list applets.
- Remove all but the essential list columns. Because there is limited screen area to display list columns, omit all unnecessary columns to minimize horizontal scrolling.

You can either remove the columns from the Handheld application or set the Show in List property on the column to FALSE. These two methods are different in the following ways. If the columns are removed from the Handheld application, the data is not downloaded to the handheld device. This keeps the Handheld application data to a minimum and, in turn, optimizes the synchronization process performance. If the Show in List property on the column is set to FALSE, the data is still downloaded to the device but is not shown by default. This second approach, though without the synchronization optimization benefit, does allow convenience for the user who wants to customize visible columns. Take these concerns into account when setting up users.

- Reorder the remaining list columns so that the most frequently used columns are furthest to the left.
- Reduce the default width of the list columns so that more columns can be viewed at one time.
- Use hyperlinks to ease navigation by enabling users to drill into a form applet.

These changes can minimize the amount of horizontal scrolling and column reordering that users must do. Limiting the number of list columns to those that are essential minimizes the amount of data downloaded to the device, resulting in faster synchronization times and more economical use of device memory.

NOTE: In Siebel Handheld applications, an asterisk (*) on a form applet, or in a column heading of a list applet, indicates a required field. Administrators specify which fields are required.

Use Siebel Tools to modify, or rearrange List columns. For more information about the List column options, see *Configuring Siebel Business Applications*.

Configuring Fields to Display as Pop-Up Text Areas in List Applets

You can configure fields to display as pop-up text areas in list applets, so that Handheld users can enter new text or edit existing text in a field. Fields that are configured to display as pop-up text areas in list applets contain an (...) ellipses button. Handheld users must click the ellipses button to access the text area.

To configure a field to display as a pop-up text area in list applets

- 1 Set the text length of the field in the business component to 100 or more.

If the field text length is set to anything less than 100, then the (...) ellipses button will not appear on the Handheld device for that field, and users will not be able access the text area.

- 2 For multi-value fields, which are read-only in Handheld applications:

- Set the text length of the field and the destination in both the parent and child business components to 100 or more.
- In the List Column of the applet, set HTML Type to Field and set Runtime to TRUE.

If the field text length is set to anything less than 100 in either the parent or child business component, then the (...) ellipses button will not appear on the Handheld device for that field, and users will not be able access the text area.

Configuring Alpha Tabs in List Applets for the Siebel Handheld Client

You can configure alpha tabs in list applets for Siebel Handheld applications to act on one or more list applet columns and to appear as a row of tabs at the top of the applet. Alpha tabs have the following characteristics:

- Can be configured for any field in the list applet provided the field is set up as the Alpha Tab Search Field user property in Siebel Tools.
- Can only be configured for one field at a time in the list applet.
- Can be configured to display either in single letter format (ungrouped) or in multiple letter (grouped) format.
- Can facilitate localizable tab definitions based on grouped and ungrouped alpha tab format. For example, alpha tabs can be configured for additional accented characters (in the case of some European Languages) and for English character ranges or local character ranges (in the case of some Asian Languages).
- Can include an ALL grouping which will show all records in the list applet by default the first time a user navigates to the view after launching the application.
- Can disable, if configured, any filters applied to the view the first time the view loads.
- Are supported in portrait orientation, landscape orientation, and on square screens.

To configure alpha tabs in list applets

- 1 In Siebel Tools, configure the AlphaTab Named Search property on the List object.

For more information about this user property, see [AlphaTab Named Search User Property on page 309](#).

- 2 Decide whether you want your alpha tabs to display in grouped or ungrouped format. The default behavior is to show grouped alpha tabs.

To display ungrouped alpha tabs, set the value of the DisableAlphaTabGroups user property for the applet to TRUE.

For more information about this user property, see [DisableAlphaTabGroups User Property on page 312](#).

By default, your Siebel Handheld application ships with grouped and ungrouped versions of alpha tabs for each language. For example:

```
IDS_ALPHATAB_GROUPED_SPEC "A-C|D-E|G-I |J-L|M-O|P-R|S-U|V-Z|*"
IDS_ALPHATAB_GROUPED_DISPLAY "ABC|DEF|GHI |JKL|MNO|PQR|STU|VWXYZ|AI I "
IDS_ALPHATAB_UNGROUPED_SPEC "A|B|C|D|E|F|G|H|I |J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|*"
IDS_ALPHATAB_UNGROUPED_DISPLAY "A|B|C|D|E|F|G|H|I |J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|AI I "
```

_SPEC represents the range for the search specification and _DISPLAY represents the strings that will be displayed on each tab. Currently, you cannot override these specifications.

Configuring Form Applets for the Siebel Handheld Client

The Siebel Handheld application uses a specialized algorithm to format and display form applets. Due to the display limitations on the device, only one pair of label and field controls are displayed on each row of the form, regardless of the layout shown within Siebel Tools.

Layout Sequence

When configuring form applets for the Siebel Handheld application, the Applet Web Template Items in the Applet Web Template determine the controls that appear on a form applet. The layout of form applets on the Handheld application is determined by the HTML Sequence field set on control objects.

The HTML Sequence field determines the screen layout of a Handheld application. Controls are ordered from top-to-bottom based on their HTML Sequence property values. Label and control pairs that do not fit on the same line as other label and control pairs wrap to the next line.

Labels

Keep field labels in applications short (approximately 12 characters or fewer, depending on the character width). Use abbreviations where possible. Labels that are too long are truncated.

Group Boxes

Group boxes are not supported and are not displayed in form applets, even if they are added in the Web Applet Designer. Therefore, reword the field labels to include group box information if necessary. For example, you can have two group boxes on the form, one labeled Ship To and another labeled Bill To. Each group box includes a field named Address. Because group boxes are not supported, the Ship To and Bill To labels are lost, and two fields with the identical Address label remain. Therefore, you must rename the Address labels “Ship To Addr” and “Bill To Addr” or some other label that distinguishes them.

Radio Buttons

You can configure any static or dynamic picklist control in form applets to display as a set of radio buttons, instead of as a normal picklist. To do this, you must set field type to DT_RADIO.

Configuring Form UI Applets for the Siebel Handheld Client

Form UI is a technique of applet design which enables Siebel Handheld application developers to create applets that are larger than normal form applets. An applet designed for form UI can have a greater height and a greater width than the applet display area. Form UI enables application developers to create large applets which can resemble a paper form.

You create form UI applets as normal form applets, but with the FormUI user property set to TRUE. For more information about this user property, see [FormUI User Property on page 314](#).

Location of UI Controls

The location of the UI controls on normal form applets is determined automatically. For form UI applets, you can define the location of each control. This enables you to design the applet in the way that you want the applet to appear to users. You can place UI controls in any location on the applet. This enables you to create forms that are easier for your users to use, or reflect your current business processes. For example, you can design an applet that resembles the layout of one of the paper forms that your business currently uses.

You can also design more complicated forms. For example, you can create a form that has three fields beside each other. Form UI allows the following benefits for UI design for Siebel Handheld applications:

- Flexible location of labels
- Hidden labels
- Flexible location of fields
- Flexible field sizes
- Flexible location of buttons
- Flexible button sizes
- Flexible location of radio buttons

You use the Grid Layout Editor in Siebel Tools to create form UI applets. Before you design the layout of the applet, you must set the FormUI user property to TRUE. In Grid Layout Editor, add the required controls to the applet, then arrange the controls appropriately.

Navigation Scheme

Form UI applets also have a new navigation scheme to enable users to navigate efficiently in a large form applet. There are no scroll bars in form UI applets. Instead, users can tap-and-hold on an area outside of the fields, then drag the form so that a part of the form that they cannot see is displayed.

To enable the new navigation scheme to function, the performance of the screen drawing functionality for form UI applets is significantly better than for other types of applet.

COM Interface

To perform complex processing on a form UI applet, you can write components that use the Component Object Model (COM) architecture. For example, if you want to implement data validation or different UI, you can write a component extension for the applet. For more information about how to write component extensions, see [Appendix G, "COM Extensibility"](#).

UI Behaviors

To reduce the potential for user confusion, many of the automatic UI behaviors which occur in the normal form applet do not occur in form UI applets. For example, the following behaviors are different:

- When the SIP opens, the form UI applet does not resize or scroll automatically to show the insertion point. Instead, the user must drag the applet until the insertion point is visible.
- When you tab to a field that is not visible, the form UI applet does not scroll automatically to the active field. Instead, the user must drag the applet until the active field is visible.

Using Auto Pop-Up Lists in the Siebel Handheld Client

Sometimes when a user opens a view, the user must select a value from a pop-up list to fill in a required field. You can configure the view so that the pop-up list automatically opens when the user opens the view. If you do this, the user does not have to tap the control to open the pop-up list.

To configure a pop-up list to open automatically, use the AutoPopupField user property. For more information about this user property, see [AutoPopupField User Property on page 310](#).

To configure a pop-up list to open automatically, add the user property on the applet with Name set to AutoPopupField.

Depending on the type of applet, this property is set differently:

- For list applets, set the AutoPopupField property to the Business Component field name.
- For form applets, set the AutoPopupField property to the name of the control.

Configuring Tree Applets for the Siebel Handheld Client

As an alternative to List applets, Tree applets with automatic drilldown behavior can be configured for Siebel Handheld applications in much the same way that you configure them for other Siebel applications.

A tree applet is a structured and hierarchical list of nodes of related business components. Tree applets are used to create explorer views where users can expand (by tapping the + button) and collapse (by tapping the - button) nodes in the tree applet and view individual records in particular folders. There are two types of nodes:

- Folder nodes - have a folder bitmap next to them and represent hierarchies within the tree.
- Record nodes - have a document bitmap next to them and represent one record in the node business component.

The hierarchy displayed in a tree applet represents master-detail relationships between records of different business components.

To configure a tree applet

- 1 In Siebel Tools, define a tree applet with the following properties:

| Property Setting | Description |
|--------------------|--|
| Class | The name of a C++ class used to manage the applet. For explorer applets, Class must be set to CSSFrameTree. This is required in order for the tree applet functionality to work. |
| Business Component | The name of the business component whose data is to be displayed. The business component must be defined in the business object specified for any view that uses this applet. |

- 2 Define trees in the tree applet by specifying the following property for each tree:

| Property Setting | Description |
|------------------|---|
| Name | The name of the tree object definition which is basically just a name, usually ending with <i>tree</i> , to which child object definitions in the form of tree nodes can be attached. |

- 3 Define tree nodes under the tree object definition. A tree node corresponds to one folder that the user sees in a tree applet. Important properties to define for each tree node object definition include the following:

NOTE: Trees and tree nodes are usually created and edited in the Applet Designer of Siebel Tools.

| Property Setting | Description |
|-----------------------|--|
| Display Name | The name of the tree node object definition (folder name) as it will appear in the tree applet in Siebel applications. The display name appears to the immediate right of the folder symbol. |
| Applet | The name of the applet that is opened in the right half of the view when the user opens the corresponding folder. Generally a list applet is specified. The applet must be based on a business component that is in the appropriate hierarchical position in the business object. If not specified, nothing opens in the right-hand side of the view, which is desirable in some situations. |
| Position | Specifies the tree node's hierarchical position relative to other tree nodes, and its sequence on its level. The top-level node has a value of 1 and all immediate child nodes of the top-level node have a value of 1.x, where x specifies the node's order relative to other nodes on the same level. In turn, all immediate child nodes of 1.x have a value of 1.x.x and so on. In general, the right most digit in a position specifies its order relative to others on the same level, and all other digits specify the position it attaches to. |
| Business Component | The business component that populates values in this tree node. This must be set to the same business component as is specified by the applet invoked by this tree node. |
| Label Field | The name of the field in the business component that the Siebel application uses to populate names in the record list that appears when the node is expanded by the user. |
| Max Child Items | An integer that sets a limit on the number of child items that appear under the tree node. An arrow appears to let the user scroll further, if there are many child items. |
| Selected Bitmap Index | An integer indicating which bitmap, in a multiple bitmap file, to display when a record is selected. This property should be set to the value 5, which corresponds to the folder symbol. |

- 4 Decide on an operation mode for your applet by setting one of the following user properties in Siebel Tools:
 - To display the applets in tabbed applet mode, set the view-level user property `ViewType` to `Tabbed`.

For more information about this property, see [ViewType User Property on page 332](#).
 - To display the applets in traditional two-applet mode, set the view-level user property `ViewType` to `Traditional`.

- To suppress the second dynamic applet, so that only a single applet is displayed, set the view-level user property `Disable Second Applet` to `TRUE`.

For modes that display two applets, you can also specify how much space you want the tree applet to occupy. For more information about this property, see [Explorer Applet Size User Property on page 313](#).

About Home Page Applets in Siebel Handheld Applications

To configure a Home Page applet, create an applet that has controls of HTML Type Button only. Use the following guidelines in creating your Home Page applets:

- Set the `MethodInvoked` property of these controls to `GotoView`. Add a `View` user property to the control, and set the value of the `View` property to the name of the view that you want to display. For more information about this user property, see [View User Property on page 331](#).
- Set the `Caption` field.
- Set the `HTML Sequence` field according to the order in which the buttons are laid out on the device. The buttons appear on the Home Page applet of the handheld device, laid out in rows of three buttons.
- Set the `class` property of the Applet to `CSSFrameCEHome`.

About Buttons in Siebel Handheld Applications

Set the `HTML Type` field on buttons to `Button`, `MiniButton`, `PushButton`, or `MiniButtonEditNew`. All of these HTML Types map to the same button control on the handheld device.

Buttons with text labels on them are sized to the minimum width required to fit the text on the buttons.

NOTE: You cannot use scripting to augment the button functionality in Siebel Handheld applications.

You can add buttons as long as the method invoked is supported by the class or superclass of the frame or business component. You can also remove buttons from an applet. Be careful when removing buttons, because you might alter the behavior of your application in unintended ways.

About Multi-Value Groups in Siebel Handheld Applications

Multi-Value Groups (MVG) are supported Handheld applications in the same way that they are supported by the Siebel Mobile Web Client. MVG fields and applets are used to input and display data on Handheld applications.

MVG fields have an MVG control button, similar to the ellipsis control (...). Tapping the MVG control—for example, on a parent form or list applet—opens the MVG applet. When the MVG applet is open, you can perform inserts, updates, or deletes on the MVG applet. Any updates performed on the MVG applet automatically appear in the corresponding MVG field, if appropriate.

NOTE: MVG fields are displayed as greyed out or read-only in the parent applet.

For more information about creating and configuring MVGs for your Siebel application, see *Configuring Siebel Business Applications*.

About the Menu Bar in Siebel Handheld Applications

The menu object definition in the Siebel repository that is implemented in Siebel Handheld applications is named SHCE Generic. The default menu bar configuration for the Handheld application includes the following menus: File, Edit, View, and Help. You can remove or rename menus on the menu bar, but you cannot add new menus. You can reorder the menu items by changing the Position property for the menu items. Generally, use the default menu bar configuration for all of your Handheld applications.

About Toolbars in Siebel Handheld Applications

This topic describes how to configure the toolbar for your Siebel Handheld application.

The following lists the supported toolbar functions:

- Back, Forward
- New Record
- New Query, Execute Query
- Next Record, Previous Record, First Record, Last Record
- Minimize or Maximize
- Delete Record

The default toolbar can be configured in Siebel Tools by configuring the SHCE Main toolbar object. SHCE Main is the main toolbar that is downloaded to the handheld device. If a toolbar named SHCE Main does not exist in your repository, a default toolbar, *Main*, is downloaded instead.

The SHCE Main toolbar contains the default buttons for the device: Back, Forward, New Record, New Query, and Execute Query. Change the order of the buttons on the display by modifying the Position property of the toolbar items. Remove a toolbar button by deleting it or making it inactive.

You can add additional default buttons to the toolbar by creating toolbar items whose Command property matches the name of an active bitmap in the SHCE Command Icons bitmap category. Do not add bitmaps to the SHCE Command Icons bitmap category because only those commands, which are already provided, are supported on the device. No additional commands are supported. You can remove a bitmap or make a bitmap inactive if you do not want it to appear in the Customize Toolbar dialog box.

You can change the bitmap for a toolbar by reimporting the bitmap from the SHCE Command Icons bitmap category.

Users can customize their toolbar from the application-level menu. For information about how to use the application-level menu, see the online help for the Siebel Handheld application. For more information about toolbars, see *Configuring Siebel Business Applications*.

About the Status Bar in the Siebel Handheld Application

The status bar is located above the toolbar. The status bar displays the status of an applet with focus, including the applet title and record item count (for example, Contacts: 1 of 13).

About Hyperlinks in the Siebel Handheld Application

Configuring a drilldown, or clicking a hyperlink or dynamic hyperlink in the Siebel Handheld application, is performed the same way as with other Siebel applications. For more information about configuring drilldowns, see *Configuring Siebel Business Applications*.

Using the COM Interface with Siebel Handheld Applications

To perform complex processing on applets, you can write components that use the COM architecture. For example, if you want to implement data validation or a more sophisticated UI, you can write a component extension for the applet. For more information about how to write component extensions, see [Appendix G, "COM Extensibility"](#).

Configuring Calendar Settings

This topic provides information about calendar enhancements such as calendar layout simplification, utilization of the menu bar as a partial toolbar, dynamic color and description coding. It also includes configuration steps and examples of implementing the enhancements.

The following topics are related to configuring calendar settings:

- ["User Properties for Calendar Settings" on page 47](#)
- ["Configuring Calendar Color Codes and Descriptions" on page 47](#)
- ["Configuring the Goto View Button" on page 48](#)

NOTE: The calendar settings are optional and need only be implemented, if required, by your business processes.

User Properties for Calendar Settings

Administrators can set the following user properties to enhance the calendar views for the Siebel Handheld applications:

- SHCEDailySlotHeight
- SHCEHideStatusBar
- SHCEHideToolBar
- SHCEMenuBackButton
- SHCEMenuForwardButton
- SHCEMenuNewButton
- UseSystemLocale

For more information about how to use these properties, see [User Properties on page 309](#).

NOTE: Registry settings take precedence over a user property configured from the user interface.

The administrator can also change the Setup.ini file to add the following Registry Setting parameters. These parameters are used by all calendar views:

```
SHCEMenuNewButton = TRUE
SHCEMenuBackButton = TRUE
SHCEMenuForwardButton = TRUE
SHCEHideToolBar = TRUE
SHCEHideStatusBar = TRUE
UseSystemLocale = TRUE
AuxiliaryConfigPath = \Program Files\Siebel Handheld\Templates\AuxConfig.txt
```

Configuring Calendar Color Codes and Descriptions

The auxconfig file contains a section for each organization and the rule expressions to define the color code and dynamic description for Daily and Weekly calendar views. Color code is rendered as the side bar color in a Daily view and is rendered as text background color of the time indicator in the Weekly view. The color code is a hexadecimal representation of the possible color to be used. The key values that can be defined are:

Default Color: Used if the Color expression does not evaluate to any color code.

Color: Used to show color codes based on the Expression defined.

Day Description: Used for the Daily view and defines which fields to display for Calendar Activity when it is shown in Daily view. The text word-wraps to multi-line display if timeslot spacing is for two or more lines.

Week Description: Used for Weekly view and defines which fields to display on Calendar Activity when it is shown in Weekly view. The number of fields displayed can be reduced due to limited display space.

Table 6 shows an example of a section from the auxconfig.txt file:

Table 6. Key Value Example

| Key Value | Description |
|---|--|
| <1Ls-9n59> | Organization Id. If the administrator defines these Key Value Set under both Organization Id and Organization Name, the Organization Id takes precedence. Avoid duplicate sections for one Organization. |
| Default Color = 0000FF | Specifies what color code should be used for any activity that does not evaluate to a color code for each Color expression which in this example is set to Blue (0000FF) |
| Color = IIf ([Category] = ' Contact Call ' , (IIf([Status]=' Planned' , ' FF0000' , (IIf([Status]=' Submi tted' , ' 00FF00' , ' FFFF00')))) , (IIf([Category]=' Account Call ' , (IIf([Status]=' Planned' , ' FF0000' , (IIf([Status]=' Submi tted' , ' 00FF00' , ' FFFF00')))) , ' 0000FF'))) | Color evaluates the Expression and if the activity is Contact Call or Account Call and has a status of Planned, the color displays in red (FF0000). If the activity has a status of Submitted, the color displays in green (00FF00). If the call has a status other than Planned and Submitted, for example, Synchronized, then the color displays in yellow (FFFF00). All other kinds of activities are displayed in blue (0000FF). |
| Day Descri ption = IIf ([Category]=' Contact Call ' , [Contact Last Name] + ' ' + [Category] + ' ' + [Status] , [Descri ption]) | Day Description evaluates the expression and if the activity is Contact Call, Last Name, Category of Call, and Status is displayed. If the activity is not Contact Call, the Description field is displayed on the calendar. |
| Week Descri ption = IIf([Category]=' Contact Call ' , [Contact Last Name] + ' ' + [Status] , [Descri ption]) | Week Description evaluates the expression and if the activity is Contact Call, Last Name, and Status is displayed. If the activity is not Contact Call, the Description field is displayed on the calendar. |

Configuring the Goto View Button

Click the Goto view button to navigate to a specified view. The view name needs to be defined as a user property for the control.

To configure Goto View buttons on the Daily Calendar view

- 1 Set ClientConfigurationMode = ALL in tools.cfg file in \root\BIN\ENU directory.
- 2 Log in to Siebel Tools.

- 3 Query for Applet Name: SHCE Sales eCalendar Daily Applet - My in Applet.
- 4 Navigate to Applet, then Control.
- 5 Add the following controls to Applet Control:

| Name | Caption | HTML Type | Method Invoked | Type |
|-------------|--------------|------------|----------------|-------------|
| CalNavigate | User defined | MiniButton | GotoView | Push Button |

NOTE: The caption for all buttons can be user-defined. The method invoked is specialized.

- 6 Navigate to Applet, Control, then Control User Property.
- 7 Select CalNavigate in Control User Property.
- 8 Add the following Control User Property Item for CalNavigate:

| Name | Control | Type | Item Identifier |
|-------------|-------------|---------|-----------------|
| CalNavigate | CalNavigate | Control | 134 |

- 9 Add the following User Properties and respective values for the CalNavigate control:

| Control Name | User Property Name | User Property Value |
|--------------|--------------------|--|
| CalNavigate | View | Define the View name that you want to navigate to. |

Configuring Calendar Buttons

This topic provides information about how to configure buttons on the calendar screen. The New Account Call and Med Ed Event buttons are specific to the Siebel Pharma Handheld application.

New Account Call: Creates a new account call and populates the call time with the timeslot selected in the calendar. Additionally, you can define two user properties for the button control to specify which view and Business Object to display. By default, when the user clicks on this button, the SIS HH Pharma Account Call Products Detailed View - CE is displayed.

Med Ed: Creates a new record for Med Ed Event and populates the Start time with the timeslot selected in the calendar. Additionally, you can define two user properties for the button control to specify which view and Business Object to display. By default, when the user clicks on this button, the My Med Ed Events View - CE is displayed.

The following procedures are related to configuring calendar settings for Siebel Pharma Handheld:

- ["To configure new buttons on the Daily Calendar view" on page 49](#)
- ["To set up for Dynamic Drilldown from the Daily Calendar view" on page 51](#)

To configure new buttons on the Daily Calendar view

- 1 Set ClientConfigurationMode = ALL in tools.cfg file in \root\BIN\ENU directory.

- 2 Log in to Siebel Tools.
- 3 Query for Applet Name: SHCE Sales eCalendar Daily Applet - My in Applet.
- 4 Navigate to Applet > Control.
- 5 Add the following controls to Applet Web Template:

| Name | Caption | HTML Type | Method Invoked | Type |
|----------------|---------|------------|----------------|-------------|
| CalNewAcctCall | 'A' | MiniButton | CalNewAcctCall | Push Button |
| CalMedEd | 'E' | MiniButton | CalMedEd | Push Button |

NOTE: The caption for all buttons can be user-defined. The method invoked is specialized.

- 6 Navigate to Applet > Applet Web Template > Applet Web Template Item.
- 7 Select CalNewAcctCall in Applet Web Template.
- 8 Add the following Applet Web Template Item for CalNewAcctCall:

| Name | Control | Type | Item Identifier |
|-------------|----------------|---------|-----------------|
| NewAcctCall | CalNewAcctCall | Control | 132 |

- 9 Select CalMedEd in Applet Web Template.
- 10 Add the following Applet Web Template Item for CalMedEd:

| Name | Control | Type | Item Identifier |
|----------|----------|---------|-----------------|
| CalMedEd | CalMedEd | Control | 133 |

- 11 Add the following User Properties and respective values for the CalNewAcctCall control:

| Control Name | User Property Name | User Property Value |
|----------------|------------------------|---|
| CalNewAcctCall | Account Call View Name | Define the view name that you want to navigate to. For example: SIS HH Pharma Account Call Products Detailed View — CE |
| | Account Call BO Name | Pharma Account Call — CE |

12 Add the following User Properties and respective values for the CalMedEd control:

| Control Name | User Property Name | User Property Value |
|--------------|--------------------|--|
| CalMedEd | Med Ed View Name | Define the view name that you want to navigate to. For example: Invitees List View - CE |
| | Med Ed BO Name | Pharma ME Event |
| | Med Ed BO Name | Pharma ME Event |

13 Perform the following changes to ensure that an activity is created and displays on the daily calendar:

In the Action business component, create the following three new calculated fields:

| Field Name | Value |
|------------------|---|
| Calc Type | Meeting |
| Calc Description | Medical Event |
| Calc End Time | If (ParentBCName()= "Pharma ME Event",ParentFieldValue ("End Date"), Timestamp()) |

14 From the application, navigate to Administration - Mobile > Application Administration > Settings.

15 Add |Calc Type|Calc Description|Calc End Time! to the DefaultFields of the Action business component for the Siebel Pharma Handheld application.

Dynamic Drilldown from the Daily Calendar View to Med Ed Event

The calendar enhancement also enables a dynamic drilldown from the Daily Calendar view to Med Ed Events.

To set up for Dynamic Drilldown from the Daily Calendar view

- 1 Set ClientConfigurationMode = ALL in tools.cfg file in \root\BIN\ENU directory.
- 2 Log in to Siebel Tools.
- 3 Query for Applet Name: SHCE Sales eCalendar Daily Applet - My in Applet.

- 4 Go to Drilldown Object and create a new record with the following values:

| Name | HyperLink Field | View Name | Source Field | Business Component | Destination Field |
|--------------|-----------------|----------------------------|--------------|--------------------|-------------------|
| Med Ed Event | Category | Invitees List View — CE | ME Event Id | Pharma ME Event | Id |

- 5 Select Drilldown Object Named Activity Type and select Dynamic Drilldown destinations.
- 6 Select and copy Professional Call record and change the following fields:

| Name | Field | Value | Destination Drilldown Object |
|--------------|---------------------|--------------|------------------------------|
| Med Ed Event | Calculated Category | Med Ed Event | Med Ed Event |

The calendar enhancements also provides the ability to create an activity for every new Med Ed Event record created. The activity displays on the Daily Calendar view and enables users to drill down from the Daily Calendar view to Med Ed Event.

To auto-create an Activity related to the new Med Ed Event on the Daily Calendar

- 1 Set ClientConfigurationMode = ALL in tools.cfg file in \root\BIN\ENU directory.
 - 2 Log in to Siebel Tools.
 - 3 Select the Pharma ME Event Business Component.
 - 4 Add a new business component User property as follows:
Name: Create Activity
Value = Y
This property creates an activity with:
 - Type = Meeting
 - Description= Medical Event
 - ME Event Id = Populated with Med Ed Id
 - 5 Select the Action Business Component.
 - 6 Select the Field 'Calculated Category' and update the Calculated Value from
LookupName("TODO_TYPE",[Category])
to
IIf([ME Event Id] IS NOT NULL,"Med Ed Event",LookupName("TODO_TYPE",[Category])).
 - 7 Update the Post Default Value for Duration Minutes to 30.
- NOTE:** This cause the system to ignore the User Preference Appointment Duration (in Minutes) for calls created through the calendar.

- 8 Update the predefault Value for Planned from System: Timestamp to Expr: "IIf (ParentBCName () = 'Pharma ME Event', ParentFieldValue ('Start Date'), Timestamp())".
- 9 Compile the SRF changes and restart the service.
- 10 Add a new child LOV Value for Category Meeting of Type = Meeting.
- 11 Synchronize the handheld device with the Siebel Server.

Configuring Find Enhancements

The configuring find enhancements that are available for you to implement include:

NOTE: Configuration of the following Find enhancements is optional and can vary depending on the fields used during the implementation.

- **Enhanced Find for Contacts:** Allow users to define a view that can be used to query and filter records in the Contact List applet. The query can be based on the field defined in the parent (Contact) and child business components.
- **Enhanced Find for Accounts:** Allow users to define a view that can be used to query and filter records in the Account List applet. The query can be based on the field defined in the parent (Account) and child business components.
- **Enhanced Find for Contact Calls:** Allow users to define a popup applet that can be used to query and filter records in the Contact Pick applet for selecting the Contact to call on. The query can be based on the field defined in the parent (Contact) and child business components.
- **Enhanced Find for Account Calls:** Allow users to define a popup applet that can be used to query and filter records in the Account Pick applet for selecting the Account to call on. The query can be based on the field defined in the parent (Account) and child business components.

NOTE: When defining captions or displays, use symbolic strings for multi-lingual implementations. Define new symbolic string if you need to use different values.

The following procedures are related to configuring find enhancements:

- ["Configuring Best Times for Enhanced Find" on page 54](#)
- ["Creating New Class Definitions" on page 54](#)
- ["Creating Pick Applets" on page 55](#)
- ["Creating or Modifying Pick Lists" on page 57](#)
- ["Modifying the Account Business Component" on page 60](#)
- ["Extracting Non UI Fields During Synchronization" on page 62](#)
- ["Modifying Applets" on page 62](#)

Configuring Best Times for Enhanced Find

Best Time is a field from where the user can pick a value in time which to search against the Start and End Time to find contacts. For a Contact, if the specified Best Time falls between Start and End Time, then that Contact is displayed in the result of your search.

You can create new columns in the S_CONTACT and S_ORG_EXT tables if you want to use specialized Best Times field functionality for Enhanced Find.

To configure Best Times field functionality:

- 1 Go to the S_CONTACT and S_ORG_EXT tables.
- 2 Enter the following configuration details:

| For this Field | Add This Value |
|----------------------|----------------|
| Name | X_BEST_TIME |
| User Name | BEST_TIME Ext |
| Type | Extension |
| Nullable | TRUE |
| Physical Type | Time |
| Length | 7 |
| Cascade Clear | Ignore |
| Transaction Log Code | TRUE |
| Status | Active |

Creating New Class Definitions

The new class definition allows users to define a view that can be used to query and filter records in the Find applet. The query can be based on the field defined in the parent (for example, Account) and child business components.

The new class definition supports the Find applet to be set to query mode during initialization and supports refine query where the user can refine the previously entered value in the Find applet. The new class definition also supports query and refine query in the popup applet invoked from the Form applet.

To create new class definitions

- 1 Navigate to Class Object in the Object Explorer.

- 2 Create a New Class Object or copy an existing class such as CSSFrameNewCall and set the following properties for each of the specified Classes:

| Description | Name | DII | Object Type | SuperClass |
|--|----------------------|----------|-------------|---------------|
| Used for Enhanced Contact Find View | CSSFrameFindContact | SPHFR | Applet | CSSFrame |
| Used for Enhanced Account Find View | CSSFrameFindAccount | SPHFR | Applet | CSSFrame |
| Used for Enhanced Contact Find Popup applet in Contact Call View | CSSFramePopupContact | SSCACMFR | Applet | CSSFramePopup |
| Used for Enhanced Account Find Popup applet in Account Call View | CSSFramePopupAccount | SSCACMFR | Applet | CSSFramePopup |
| Used for Multi-Select Pick Applets | CSSFrameMultiplePick | SSEUI | Applet | CSSFrame |

Creating Pick Applets

Users can create new pick applets to specifically support the multi select feature in the pick applet. You can configure pick applets where only one field such as Name or Postal Code is part of the pick map definition.

To create pick applets

- 1 Navigate to HH LS Contact Account Affiliation Pick Applet and copy the record.
 - a Set the following values:
 - Class:** CSSFrameMultiplePick
 - Name:** HH LS Contact Account Affiliation Pick Applet - Multi Select.
 - b Go to Controls and query for 'List' control.
 - c Go to the Control user property and add the following values to the user property:
 - Name:** AutoSelect
 - Value:** TRUE
 - d Go to List Column, delete all List Column records and add a new record called Name with the following values:
 - Display name:** Name
 - HTML Type:** Field

- e Edit Layout to remove the old list columns and add the Name control instead.
- 2 Navigate to HH LS Contact Account Affiliation Pick Applet and copy the record.
 - a Set the following values:

| For this Field | Add This Value |
|--------------------|---------------------------------|
| Class | CSSFrameMultiplePick |
| Name | Find Pharma Address Postal Code |
| Business Component | Pharma Address |

- b Go to Controls and query for 'List' control.
- c Remove values from HTML Display Mode, HTML Type.
- d Go to the Control user property and add the following values to the user property:
 - Name:** AutoSelect
 - Value:** TRUE
- e Go to List Column, delete all List Column records and add a new record called Postal Code with the following values:
 - Display name:** Postal Code
 - HTML Type:** Field
- f Edit Layout to remove the old list columns and add Postal Code instead.
- 3 Navigate to Find Pharma Address Postal Code and copy the record.
 - a Set the following values:

| For this Field | Add This Value |
|--------------------|----------------------------|
| Class | CSSFrameMultiplePick |
| Name | Find CUT Address - CE City |
| Business Component | CUT Address - CE |

- b Go to Controls and query for 'List' control.
- c Go to the Control user property and add the following values:
 - Name:** AutoSelect
 - Value:** TRUE
- d Go to List Column, delete all List Column records and add a new record called City with the following values:
 - Display name:** City
 - HTML Type:** Field

- e Edit Layout to remove the old list columns and add City instead.
- 4 Navigate to Find Pharma Address Postal Code and copy the record.
 - a Set the following values:

| For this Field | Add This Value |
|--------------------|-----------------------------------|
| Class | CSSFrameMultiplePick |
| Name | Find CUT Address - CE Postal Code |
| Business Component | CUT Address - CE |

- b Go to Controls and query for 'List' control.
- c Go to the Control user property and add the following values:
 - Name:** AutoSelect
 - Value:** TRUE
- d Go to List Column, delete all List Column records and add a new record called Postal code with the following values:
 - Display name:** Postal code
 - HTML Type:** Field
- e Edit Layout to remove the old list columns and add Postal Code instead.

Creating or Modifying Pick Lists

The following procedure describes how to create or modify pick lists as part of the Find enhancements.

To create or modify pick lists

- 1 Navigate to the Pick List object definition in the Object Explorer and create a new record with the following values:

| Name | Business Component | Bounded | Long List | No Insert/No Delete/No Update/No Merge |
|--|--------------------|---------|-----------|--|
| Pharma Picklist Pharma Address - CE | Pharma Address | TRUE | TRUE | TRUE |

- 2 Navigate to the Pick List object definition in the Object Explorer and select the Picklist Contact Account Affiliation record:

| Name | Business Component | Bounded | Long List | No Insert/No Delete/No Update/No Merge |
|--------------------------------------|-----------------------------|---------|-----------|--|
| Picklist Contact Account Affiliation | Contact Account Affiliation | TRUE | TRUE | TRUE |

To modify the Contact Business Component

- 1 Navigate to the Contact Business Component.
- 2 Right-click the field object and select New Record.
- 3 Set the field value with the following property set:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Best Time |
| Type | DTYPE_TIME |
| Join | S_CONTACT |
| Column | X_BEST_TIME |

- 4 Create the following Multi Value Links with the following values:

| Name | Destination Business Component | Destination Link | NoAssociate/NoDelete/NoInsert/NoUpdate/NoCopy |
|-----------------------------|--------------------------------|---------------------------------------|---|
| Best Call Times | CSLS Contact Best Times - CE | Contact/CSLS Contact Best Times - CE | TRUE |
| Status | Pharma Professional Call - CE | Contact/Pharma Professional Call - CE | TRUE |
| Contact Account Affiliation | Contact Account Affiliation | Contact/Contact Account Affiliation | TRUE |

- 5 Right-click the field object, select New Record, and then set the following property values:

| Name | Dest Field | Multi-value Link | Multi-valued | Picklist |
|------------|-------------|------------------|--------------|--------------|
| CallStatus | Call Status | Status | TRUE | |
| StartDay | Start Day | Best Call Times | TRUE | Day PickList |

- 6 Navigate to Pick Maps (Child Object of Field) and set the property as:

Field: Start Day

Picklist Field: Value

- 7 Right-click the field object, select New Record, and then set the following property values:

| Name | Dest Field | Multi-value Link | Multi-valued |
|-----------|------------|------------------|--------------|
| StartTime | Start Time | Best Call Times | TRUE |
| End Time | End Time | Best Call Times | TRUE |

- 8 Right-click the Account field, and select Copy Record from the menu.

- 9 Name the new field as Account Find and set the Picklist field property as Picklist Contact Account Affiliation.

- 10 Set the field value with the following property values:

| Field | Picklist Field | Dest Field | Multi-value Link | Multi-valued | Picklist |
|--------------|--------------------------------------|------------|-----------------------------|--------------|--------------------------------------|
| Account Find | Picklist Contact Account Affiliation | Name | Contact Account Affiliation | TRUE | Picklist Contact Account Affiliation |

- 11 Navigate to Pick Maps (Child Object of Field) and set the property as:

Field: Account Find

Picklist Field: Name

- 12 Right-click the Postal Code - CE field, and select Copy Record.

- 13 Name the new field Postal Code Find and set the Picklist field property as Pharma Picklist Pharma Address - CE.

- 14 Set the field value with the following property values:

| Field | Picklist Field | Dest Field | Multi-value Link | Multi-valued | Picklist |
|------------------|-------------------------------------|------------|------------------|--------------|-------------------------------------|
| Postal Code Find | Pharma Picklist Pharma Address - CE | - | - | - | Pharma Picklist Pharma Address - CE |

- 15 Navigate to Pick Maps (Child Object of Field) and set the property as:

Field: Postal Code Find

Picklist Field: Postal Code

- 16 Select the Contact Id field and copy the record.

17 Set the field value with the following property set:

| For this Field | Add This Value |
|----------------|---------------------|
| Name | Last Call Date Find |
| Type | DTYPE_DATE |
| Join | Position Join |
| Column | LAST_CALL_DT |

To force specific non UI fields down during synchronization

- 1 Navigate to the Administration - Mobile screen > Application Administration view.
- 2 Select the Settings view tab.
- 3 Select Siebel ePharma CE Application Name.
- 4 Click Query, enter Contact, and click Execute Query
- 5 Select the DefaultFields Directive for the Contact Business Component and add the following to the end in the Value column:
|Start Time|End Time

Modifying the Account Business Component

You need to modify the Account business component to include the newly created field for the Find Enhancement feature.

To modify the Account business component

- 1 Navigate to Account business component.
- 2 Right-click the field object and select New Record.
- 3 Set the field value with the following property set:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Best Time |
| Type | DTYPE_TIME |
| Join | S_ORG_EXT |
| Column | X_BEST_TIME |

- 4 Create the following Multi Value Links with the following values:

| Name | Destination Business Component | Destination Link | NoAssociate/NoDelete/NoInsert/NoUpdate/NoCopy |
|-----------------|--------------------------------|--------------------------------------|---|
| Best Call Times | Retail Outlet Best Call Time | Account/Retail Outlet Best Call Time | TRUE |
| Status | Pharma Account Call - CE | Account/Pharma Account Call - CE | TRUE |

- 5 Right-click the field object, select New Record, and then set the following property values:

| Name | Dest Field | Multi-value Link | Multi-valued | Picklist |
|------------|-------------|------------------|--------------|--------------|
| CallStatus | Call Status | Status | TRUE | |
| StartDay | Start Day | Best Call Times | TRUE | Day PickList |

- 6 Navigate to Pick Maps (Child Object of Field) and set the property as:

Field: Start Day

Picklist Field: Value

- 7 Right-click the field object, select New Record, and then set the following property values:

| Name | Dest Field | Multi-value Link | Multi-valued |
|-----------|------------|------------------|--------------|
| StartTime | Start Time | Best Call Times | TRUE |
| End Time | End Time | Best Call Times | TRUE |

- 8 Right-click the Postal Code - CE field, and select Copy Record from the menu.

- 9 Name the new field as Postal Code Find and set the Picklist field property as Pharma Account Address PickList - CE.

- 10 Navigate to Pick Maps (Child Object of Field) and set the property as:

Field: Postal Code Find

Picklist Field: Postal Code

- 11 Right-click the Postal Code Find field, and select Copy Record.

- 12 Select My Position Id and copy the record.

13 Set the field value with the following property set:

| For this Field | Add This Value |
|----------------|---------------------|
| Name | Last Call Date Find |
| Type | DTYPE_TIME |
| Join | Account Position |
| Column | LAST_CALL_DT |

14 Name the new field City Find and ensure that the Picklist field property is set as Pharma Account Address PickList - CE.

15 Navigate to Pick Maps (Child Object of Field) and set the property as:

Field: City Find

Picklist Field: City

Extracting Non UI Fields During Synchronization

Fields displayed in the UI and fields that are part of the calculated field are extracted to the device during the synchronization process. However, not all fields in the business component are extracted. Often, you might need to refer to others fields in the user property that are either not exposed in the UI or calculated fields. For example, a user enters a value in the Amount field through the UI. This action updates a value in the Status field, but the Status field is not displayed in the UI and therefore is not automatically extracted to the device. Such fields need to be forced down to the device during synchronization.

To extract specific non UI fields during synchronization

- 1 Navigate to Administration - Mobile > Application Administration view.
- 2 Select the Settings view Tab.
- 3 Select Siebel ePharma CE Application Name.
- 4 Click Query, enter Account, and click Execute Query.
- 5 Select the DefaultFields Directive for the Account business component and add the following to the end of the Value column:

|Start Time|End Time

Modifying Applets

The following procedures describe how to modify applets, add new columns to applets and to include control level user properties to support Auto Select.

To modify applets

- 1 Add a Find button to Pharma Professional List Applet - CE and Pharma Account List Applet - CE:
 - a Set the field value with the following property set:

| For this Field | Add This Value |
|----------------|----------------|
| Display Name | Find |
| MethodInvoked | FindView |
| HTML Type | MiniButton |

- b Navigate to Applet WebLayout.
 - c Change the Application Context to Siebel ePharma CE.
 - d Add a new button after New Call Button.
 - e Save the layout.
 - f Change the Application Context to All Applications.
- 2 Add the PopupFindApplet control user property to Pharma Professional Call Form Applet - CE.
 - a Select ContactLastName control and go to the control user property.
 - b Add a new record with Name as PopupFindApplet and Value as Contact Find Popup Applet.
 - c Add a new control to Applet Web Template in Layout.

| For this Field | Add This Value |
|----------------|---------------------------|
| Name | Find |
| Detail Applet | Contact Find Popup Applet |
| Visible | False |

- 3 Add the PopupFindApplet Control User Property to Pharma Account Call Form Applet - CE.
 - a Select AccountName Control and navigate to the Control User Property.
 - b Add a new record with Name as PopupFindApplet and Value as Account Find Popup Applet.
 - c Add a new control to Applet Web Template in Layout.

| For this Field | Add This Value |
|----------------|---------------------------|
| Name | Find |
| Detail Applet | Account Find Popup Applet |
| Visible | False |

- 4 You must create four new applets, as detailed in the following tasks:

"To create APPLET #1" on page 64

"To create APPLET #2" on page 68

"To create APPLET #3" on page 72

"To create APPLET #4" on page 72

To create APPLET #1

- 1 Create a new Applet called HH Contact Find Form Applet - CE.
- 2 Set the applet with the following property set:

| For this Field | Add This Value |
|--------------------|----------------------------------|
| Name | HH Contact Find Form Applet - CE |
| Business Component | Contact |
| Class | CSSFrameFindContact |
| Title | Contact |
| Type | Standard |
| No Delete | TRUE |

- 3 Navigate to Applet User Prop (Child Object of an Applet) and add the following user properties. The following user properties are set to support multi-select in an pick applet. The Pick Applet Property has values defined as |Field name|, |Dest Field|.

| For this Field | Add This Value |
|-------------------|-------------------------------|
| PickApplet 1 | Account Find, Name |
| PickApplet 2 | Postal Code Find, Postal Code |
| SHCEHideToolBar | TRUE |
| SHCEHideStatusBar | TRUE |

- 4 Navigate to Controls (child object of the applet)
 - a Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Account Find |
| Caption | Affiliations |
| Field | Account Find |

| For this Field | Add This Value |
|----------------|--|
| HTML Type | Field |
| Pick Applet | HH LS Contact Account Affiliation Pick Applet - Multi Select |
| Runtime | TRUE |

- b** Once the Account Control is added, navigate to its user property (Control User Prop), and add the following values:

Name: AutoSelect

Value: TRUE

- c** Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | All |
| Caption | All |
| HTML Type | MiniButton |
| Method Invoke | AllContacts |

- d** Right-click on to the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Best Time |
| Caption | Best Call Time |
| Field | Best Time |
| HTML Type | Field |
| Runtime | TRUE |

- e** Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|---------------------------------|
| Name | Postal Code Find |
| Caption | Postal Code |
| Field | Postal Code Find |
| HTML Type | Field |
| Pick Applet | Find Pharma Address Postal Code |
| Runtime | TRUE |

- f Once the Postal Code Find Control is added, navigate to its user property (Control User Prop) and add the following values:

Name: AutoSelect

Value: TRUE

- g Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|---------------------|
| Name | Last Call Date Find |
| Caption | Last Call Date |
| Field | Last Call Date Find |
| HTML Type | Field |

- h Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Rep Specialty |
| Caption | Specialty |
| Field | Rep Specialty |
| HTML Type | Field |
| Runtime | TRUE |

- i Navigate to its user property (Control User Prop) and add the following values:

Name: AutoSelect

Value: TRUE

- j Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Route |
| Caption | Route |
| Field | Route |
| HTML Type | Field |
| Runtime | TRUE |

- k Navigate to its user property (Control User Prop) and add the following values:

Name: AutoSelect

Value: TRUE

- l Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Start Day |
| Caption | Day |
| Field | Start Day |
| HTML Type | Field |
| Runtime | TRUE |

- m Navigate to its user property (Control User Prop) and add the following values:

Name: AutoSelect

Value: TRUE

- n Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Search |
| Caption | Search |
| HTML Type | MiniButton |
| Method Invoke | SearchContacts |

- o Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|---------------------------|
| Name | With/Without Planned Call |
| Caption | Planned |
| Field | CallStatus |
| HTML Type | Checkbox |
| Runtime | TRUE |

- 5 Navigate to Applet Web Template and create a record with the following values:

| For this Field | Add This Value |
|----------------|---------------------------------|
| Name | Edit |
| Type | Edit |
| Sequence | 1 |
| Web template | Applet Form 4 column (Edit/New) |

- 6 Right-click on Applet name and Edit Web Layout.
- 7 Add the new Controls on to the form.

To create APPLET #2

- 1 Create a new Applet called HH Account Find Form Applet - CE with the following values:

| For this Field | Add This Value |
|--------------------|----------------------------------|
| Name | HH Account Find Form Applet - CE |
| Business Component | Account |
| Class | CSSFrameFindAccount |
| Title | Account |
| Type | Standard |
| No Delete | TRUE |

- 2 Navigate to Applet User Prop (child object of an applet) and add the following user property. The following user properties are set to support multi-select in an pick applet:

| For this Field | Add This Value |
|-------------------|-------------------------------|
| PickApplet 1 | City Find, City |
| PickApplet 2 | Postal Code Find, Postal Code |
| SHCEHideToolBar | TRUE |
| SHCEHideStatusBar | TRUE |

- 3 Navigate to Controls (child object of the applet)

- a Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | All |
| Caption | All |
| HTML Type | MiniButton |
| Method Invoke | AllAccounts |

- b Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Best Time |
| Caption | Best Call Time |
| Field | Best Time |
| HTML Type | Field |
| Runtime | TRUE |

- c Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------------------|
| Name | City |
| Caption | City |
| Field | City Find |
| HTML Type | Field |
| Pick Applet | Find CUT Address - CE City |
| Runtime | TRUE |

- d Once the City Control is added, go to its user property (Control User Prop) and add the following values:

Name: AutoSelect

Value: TRUE

- e Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|------------------|
| Name | Postal Code Find |
| Caption | Postal Code |

| For this Field | Add This Value |
|----------------|-----------------------------------|
| Field | Postal Code Find |
| HTML Type | Field |
| Pick Applet | Find CUT Address - CE Postal Code |
| Runtime | TRUE |

- f Once the Postal Code Find Control is added, go to its user property (Control User Prop) and add the following values:

Name: AutoSelect

Value: TRUE

- g Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|---------------------|
| Name | Last Call Date Find |
| Caption | Last Call Date |
| Field | Last Call Date Find |
| HTML Type | Field |

- h Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Route |
| Caption | Route |
| Field | Route |
| HTML Type | Field |

- i Once the Route Control is added, go to its user property (Control User Prop) and add the following values:

Name: AutoSelect

Value: TRUE

- j Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Start Day |
| Caption | Start Day |

| For this Field | Add This Value |
|----------------|----------------|
| Field | Start Day |
| HTML Type | Field |

- k Once the Start Day Control is added, go to its user property (Control User Prop) and add the following values:

Name: AutoSelect

Value: TRUE

- l Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Search |
| Caption | Search |
| HTML Type | MiniButton |
| Method Invoke | SearchAccounts |

- m Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|----------------|
| Name | Type |
| Caption | Account Type |
| Field | Type |
| HTML Type | Field |

- n Once the Type Control is added, go to its user property (Control User Prop) and add the following values:

Name: AutoSelect

Value: TRUE

- o Right-click on the controls, select New Record, and add the following values:

| For this Field | Add This Value |
|----------------|---------------------------|
| Name | With/Without Planned Call |
| Caption | Planned |
| Field | CallStatus |
| HTML Type | Checkbox |

- 4 Navigate to Applet Web Template and create a record with the following values:

| For this Field | Add This Value |
|----------------|---------------------------------|
| Name | Edit |
| Type | Edit |
| Sequence | 1 |
| Web template | Applet Form 4 column (Edit/New) |

- 5 Right-click on the Applet name and Edit Web Layout.
- 6 Add the new Controls on to the form.

To create APPLET #3

- 1 Copy Applet #1 (HH Contact Find Form Applet - CE) and change the value of the following property set:

| For this Field | Add This Value |
|--------------------|---------------------------|
| Name | Contact Find Popup Applet |
| Business Component | (Leave blank) |
| Class | CSSFramePopupContact |
| Type | Pick List |

- 2 Select Applet Web Template Object and change Web Template name to Popup Query.
- 3 Select Applet and set Popup Dimension Property in the Properties window to 240 X 200.

To create APPLET #4

- 1 Copy Applet #2 (HH Account Find Form Applet - CE) and change the value of the following property set:

| For this Field | Add This Value |
|--------------------|---------------------------|
| Name | Account Find Popup Applet |
| Business Component | (Leave blank) |
| Class | CSSFramePopupContact |
| Type | Pick List |

- 2 Select Applet Web Template Object and change Web Template name to Popup Query.
- 3 Select Applet and set Popup Dimension Property in Properties window to 240 X 200.

- 4 Add the applet user property PopupCommitParent = False.

To set the view property

- 1 Navigate to the view Object definition and query for Pharma Professional Calls View - CE.
- 2 Right-click on the Pharma Professional Calls View - CE view and select Copy Record.
- 3 Change the following property value:
Name: Pharma Professional Find View - CE
Business Object: Contact
- 4 Delete any View Report Records, Thread applet Properties, Add to History = False.
- 5 Navigate to View Web Template Item (Child object of View Web Template).
- 6 Add HH Contact Find Form Applet - CE Applet, and delete any other applets if they exist.

There should be only one applet associated to this view with the following values:

| For this Field | Value |
|-----------------|----------------------------------|
| Name | HH Contact Find Form Applet - CE |
| Item Identifier | 1 |
| Applet | HH Contact Find Form Applet - CE |
| Applet Mode | Edit |

- 7 Repeat [Step 1](#) and [Step 2](#) and change the following property value:
Name: Pharma Account Find View - CE
Business Object: Account
- 8 Repeat [Step 4](#).
- 9 Add HH Account Find Form Applet - CE Applet, and delete any other applets if they exist.

There should be only one applet associated to this view with the following values:

| For this Field | Value |
|-----------------|----------------------------------|
| Name | HH Account Find Form Applet - CE |
| Item Identifier | 1 |
| Applet | HH Account Find Form Applet - CE |
| Applet Mode | Edit |

To add the previously-created views to the screen

- 1 Navigate to Screen Object definition and query for ePharma Professionals CE.

- 2 Navigate to Screen Views (child object of screen).
- 3 Right-click on screen views, select New Record, and set the screen views property with the following values:

| For this Field | Value |
|----------------|---|
| Name | Pharma Professional Find View - CE |
| View | Pharma Professional Find View - CE |
| Type | Aggregate View |
| Sequence | 15 (pick the next unused sequence number) |

Leave all other properties empty.

- 4 Navigate to the Screen Object definition and query for ePharma Account CE.
- 5 Navigate to Screen Views (child object of screen).
- 6 Right-click on the screen views, select New Record, and set the screen views property with the following values:

| For this Field | Value |
|----------------|---|
| Name | Pharma Account Find View - CE |
| View | Pharma Account Find View - CE |
| Type | Aggregate View |
| Sequence | 15 (pick the next unused sequence number) |

Leave all other property empty.

- 7 Add New Views to Views/Responsibilities for the representative.

Modifying Business Components for Extracting Popup Applets

The following procedures describe how to modify the Pharma Professional Call and Pharma Account Call business components.

To modify a business component

- 1 Navigate to the business component you want to modify. For example, Pharma Professional Call - CE or Pharma Account Call - CE.
- 2 In the field list, create a new field.
- 3 Name the new field, as follows:

- Last Name Popup if modifying Pharma Professional Call - CE.
 - Account Popup if modifying Pharma Account Call - CE.
- 4 Set the Calculated property to Y.
 - 5 Select the default value for all other properties.

Extracting Contact Find and Account Find Popup Applets

The following procedures describe how to extract the Contact Find and Account Find popup applets.

To extract Contact Find popup applets

- 1 Navigate to Pharma Professional Call Form Applet - CE.
- 2 In the applet controls, create a new record and set the following properties for the control:

| For this Field | Value |
|----------------|---------------------------|
| Name | Last Name Popup |
| Field Picklist | Last Name Popup |
| Detail Applet | Contact Find Popup Applet |
| Visible | False |

- 3 Select the default value for all other properties.
- 4 Add the new control to Edit Applet Web Template as a Web Template item.
- 5 Set the item identifier to 1315 or any appropriate available number.

To extract Account Find popup applets

- 1 Navigate to Pharma Account Call Form Applet - CE.
- 2 In the applet controls, create a new record and set the following properties for the control:

| For this Field | Value |
|----------------|---------------------------|
| Name | Account Popup |
| Field Picklist | Account Popup |
| Detail Applet | Account Find Popup Applet |
| Visible | False |

- 3 Select the default value for all other properties.
- 4 Add the new control to Edit Applet Web Template as a Web Template item.

- 5 Set the item identifier to 1316 or any appropriate available number.

Setting Key Values in the Auxiliary Configuration File

Administrators can define which view to display when a user clicks the Find button. The View name, BusObjName, and BusCompNames key value are specified in the auxiliary configuration file. If no key value is specified, then by default, the Pharma Professional Find View - CE screen is displayed for Contacts BusComp. The Pharma Account Find View - CE screen is displayed by default for Accounts BusComp.

The following is an example of entries in an auxiliary configuration file (auxconfig.txt):

```
<Xerco Pharmaceuticals (PH ENU)>
Contact View Name = XC Pharma Professional Find View - CE
Account View Name = XC Pharma Account Find View - CE
Contact BusObjName = Contact
Account BusObjName = Account
Contact BusCompNames = Contact
Account BusCompNames = Account
```

NOTE: If the administrator defines key values under both Organization ID and Organization Name, the Organization ID takes precedence. Avoid duplicate sections for one Organization.

A key value set pair provides flexibility to define which fields should be shown for each organization in the enhanced Find screen. The administrator can specify the field names that do not need to be shown for the Organization under the Organization name or Organization Id section in the auxiliary configuration file.

Under the organization section, Contact Field Names is the key value for Contact enhanced Find screen and Account Field Names is the key value for the Account enhanced Find screen. For the values list, the field names that you do not want to display must be separated by a comma.

For example:

```
Contact Field Names = Route, Primary Specialty, Postal Code Find
Account Field Names = City Find, Route, Postal Code Find, Type
```

Administrators can also specify the multi value list names used in the enhanced Find screen in the auxconfig.txt for each organization.

For example:

```
Account Link Names = CUT Address - CE, Best Call Times, Status
Contact Link Names = Contact Account Affiliation, Pharma Address, Best Call Times,
Status
```

Developing Tasks for Task UI in Siebel Handheld Applications

Siebel Task UI enables you to create an interactive tool that can help users to perform tasks. *Siebel Tasks* are tools that display a series of screens, which guide users through the steps required to perform a task. Siebel Tasks are similar to software wizards.

Siebel Tasks can allow the user to navigate backwards and forwards within the task, and can allow users to pause and resume the task.

You can use Siebel Tasks to ensure that your business processes are executed as consistently and efficiently as possible. You can replicate the steps of your business processes in a task. For example, if your business process requires that the contact details of a new account are recorded before an order can be generated for the account, you can construct a task to reflect this requirement. When the users start the task to assign an order to a new account, the first screen they must complete is the contact details screen. The users cannot navigate to the order details screen until they enter all the required data on the contact details screen.

Siebel Tasks can guide inexperienced and intermittent users through the execution of unfamiliar tasks, and in that way increase the efficiency of these users.

The following procedures describe how to implement tasks in your Siebel Handheld applications:

- [“Designing a Task for Siebel Handheld Applications” on page 77](#)
- [“Creating a Task for Siebel Handheld Applications” on page 78](#)
- [“Example of Creating a Task for Siebel Handheld Application” on page 79](#)

For more information about Siebel Task UI, see *Siebel Business Process Framework: Task UI Guide*.

To perform complex processing on a task UI applet, you can write components that use the COM architecture. For example, if you want to implement data validation or a more sophisticated UI, you can write a component extension for the applet.

For more information about how to write component extensions, see [Appendix G, “COM Extensibility”](#).

Designing a Task for Siebel Handheld Applications

Before you create a task for a Siebel Handheld application, you must design the structure and sequential logic of the task.

The design involves doing the following:

- 1 Analyze the task, and divide the task into component steps. The steps that you identify will become individual screens in the task.
- 2 Identify the sequence in which you want users to perform the steps that comprise the task. That is, you must define how the task flows. If an error is detected on the current screen, you might want to remain on the current screen until the error is resolved. In some situations, you might want the sequence to reverse.

- 3 Identify the data that the user must enter or that must be displayed in the task.

Creating a Task for Siebel Handheld Applications

To create a Siebel Handheld application task, perform the following steps:

- 1 Create the task in Siebel Tools. A task for a Handheld application is implemented as a view which contains a series of task applets. You must create an applet for each screen.
- 2 Set the user property of the view to indicate that the view is a Siebel Task.
- 3 You create the applets as form UI applets; that is, with the FormUI property set to TRUE. The applets that you create must meet the following requirements:
 - Users cannot scroll on the applet, so the applet must fit in the display area of the target device.
 - Any fields that users can edit must be in a location that is not obscured when the SIP is displayed.
 - The only UI items that can appear in the bottom 20 pixels of the screen are the navigation buttons.
 - The navigation bar, menu bar, toolbar, and status bar on task applets are hidden. The applet must use the navigation buttons to control all navigation.

For more information about form UI applets, see [Configuring Form UI Applets for the Siebel Handheld Client on page 40](#).

- 4 Add UI controls to the task applet.
- 5 Add one, two, or three navigation buttons to the applets. Use frame methods to associate functionality with the buttons.

To add navigation buttons to an applet, you define the buttons in the user properties for the applet. For example, if you want to configure Back, Cancel, and OK buttons for an applet named OrderTaskProduct, add the following user property to the applet:

```
TaskNavigation="Back, Cancel, Ok(OrderTaskProduct)"
```

Note the following:

- A task applet does not automatically commit data when the user steps off the applet into another task applet in the same view. You must create the required code to ensure that the data is committed.

Alternatively, data is committed when the user steps off the view.

- To perform more complex processing in a task, you can write components that use the COM architecture. For example, if you want to implement data validation or a more sophisticated UI, write a component extension for the Handheld application.

For more information about how to write component extensions, see [Appendix G, "COM Extensibility"](#).

Example of Creating a Task for Siebel Handheld Application

The following example is a simple task for Siebel Handheld applications that users can execute to take a simple order. This task creates three screens, as follows:

- Select Product screen
- Specify Quantity screen
- Confirm Order screen

To create this task, perform the following steps:

- 1 In Siebel Tools, create a normal view named OrderTaskView.
- 2 Specify that the view is a task view, that is, set the TaskUI user property, as follows:
TaskUI = "TRUE"
For more information about this user property, see [TaskUI User Property on page 327](#).
- 3 Create three applets on the view with the following names:
 - OrderTaskProduct
 - OrderTaskQuantity
 - OrderTaskSummary
- 4 For each applet, set the user properties as described in the following table:

| User Property | OrderTaskProduct | OrderTaskQuantity | OrderTaskSummary |
|----------------|---------------------|---------------------|------------------------|
| FormUI | "True" | "True" | "True" |
| TaskBtnMethod1 | "UndoRecord,GoBack" | "UndoRecord,GoBack" | "NewRecord,GotoApplet" |
| TaskBtnBitmap1 | "Delete Record" | "Delete Record" | "More" |
| TaskBtnMethod2 | "NewRecord" | "GotoApplet" | "GotoApplet" |
| TaskBtnTarget2 | — | "TASK1" | "TASK2" |
| TaskBtnBitmap2 | "New Record" | "Go Back" | "Go Back" |
| TaskBtnMethod3 | "GotoApplet" | "GotoApplet" | "GoBack" |
| TaskBtnTarget3 | "TASK2" | "TASK3" | — |
| TaskBtnBitmap3 | "Go Forward" | "Go Forward" | "Goto Last" |

- 5 In Grid Layout Editor, add the required controls to each applet, and arrange the controls appropriately.

NOTE: No data validation is performed in this simple Task UI.

Scripting in the Siebel Handheld Application

The Siebel Handheld application does not execute Siebel Visual Basic (VB) or Siebel eScript scripting.

The Handheld application runs Siebel VB or Siebel eScript scripts and specialized business component logic at synchronization time, rather than in real time on the handheld device. This has significant consequences in applets and controls, because scripts attached to applets and controls are ignored.

CAUTION: Be aware of the following additional issues with respect to scripts and validation:

- **Deactivating scripts.** *Do not* deactivate existing scripts in Siebel Tools to make the handheld device or the synchronization process accept data it is excluding. Errors can occur within the Siebel Web Client or elsewhere from where these scripts are run.
- **Calls to user interface methods.** When a business component script makes a call to a user interface method (such as Msg Box), the script cannot be executed, and the record update, deletion, or addition is rejected during synchronization. The Siebel Visual Basic code must be restricted to the business component level. Applet-level Siebel Visual Basic does not execute and, therefore, must not be written for the Handheld application. As a substitute to using Siebel Visual Basic on the Handheld application, for real-time execution, use the following business component user properties:
 - BC/Field Read only field for making fields or business components read-only dynamically.
 - Pickmap for fields on the picklist.

Do not use message boxes or prompts in the Siebel Visual Basic code used in conjunction with the Handheld application business components.

If the business component is shared between the Siebel Handheld Client and other applications, use the following constructs to determine whether the code is executed by the handheld synchronization process or if the code is executed in real-time by other applications:

```
'ActiveViewName' is only available when the script is called real time on the laptop
If (TheApplication.ActiveViewName<>"") then
  MsgBox"....."
End If
```

ActiveViewName is not available if the business component is accessed by other server processes, such as workflow or business service.

To determine if the business component is called by synchronization for the Handheld application, do the following:

- 1 Add a server script to Application_Start event of the Siebel Handheld application.

To do this, set a custom profile attribute with SetProfileAttr. If the custom attribute is not set, this indicates that:

- a The Application_Start event of the Handheld application did not occur.
- b The business component is called by something other than synchronization process for the Handheld application.

Business component level scripting is executed during the synchronization process for the Handheld application. You might want to enhance error handling if the business component is shared between the Handheld application and the laptop.

Compiling the Siebel Handheld Application

Compiling the repository for the Siebel Handheld application is identical to compiling on the Siebel Web Client. For more information about compiling the repository, see *Using Siebel Tools*. For deployment to production, always perform a full compilation. This assumes that you have organized all of the object definitions for your Handheld application in one project or a limited set of projects.

When you compile your application, a Siebel repository file (.srf) is created and put in the destination directory you specified during the compile procedure. Make the .srf file the source of repository information for the Handheld application test client machine, defining all Siebel applications for that client, when you move it to your local \Siebel\objects directory.

It is strongly recommended that you make a backup of the existing .srf file in your local \Siebel\objects directory before overwriting it with the new one. That way, if you make an error or you want to revert to the original application, you have a backup that you can easily restore. Versioning for the repository is tracked, and if the .cfg, schema, or repository files have changed, the synchronization process will automatically rebuild the .rml and schema. For more information about compiling the Handheld application, see *Using Siebel Tools*.

Installing and Setting Up End User Translation Functionality

End User Translation functionality provides you with the ability to translate the text strings within the Siebel Handheld application. The translation process must happen during the development cycle and once completed your development team can then package the translated strings for distribution.

The translation process requires the generation of a file of strings from the binaries of the Handheld application which are then translated and distributed to the user base. This file, userstr.txt, containing the translated strings is then used to override the strings within the existing binaries for the Handheld application. Therefore, you will be required to build a copy of userstr.txt into the installation media for your Handheld application.

The translation string file is version specific so you must repeat this process every time you upgrade your Handheld application.

The following procedures are related to installing and setting up end user translation functionality:

- [“Performing Setup for Initial String Translation” on page 82](#)
- [“Distributing Translated Strings” on page 83](#)

Performing Setup for Initial String Translation

In order to perform the initial string translation you must perform the procedure in this topic prior to installation:

To set up Siebel Handheld application for initial string translation

- 1 In the installation directory of the Siebel Handheld application, add the following two lines to the setup.ini file:

```
[EndUserLocal i zati on]
UserDefi nedLang=Y
```

- 2 Install the Siebel Handheld application.

During application initialization and the subsequent startup, the file 'userstr.tmp' is generated within the ApplicationDataDirectory (by default this is \Program Files\Siebel Handheld). This file is generated upon startup whenever the UserDefinedFlag is set to Y within the registry of the Handheld application and an existing userstr.txt cannot be found.

- 3 Copy the generated file over to the computer where you will be performing the string translation.
- 4 Translate the strings as required and remove any untranslated strings from the file.

For example, given the following exported strings:

```
33872|HTML\
33873|Tab delimited text file\
33874|Export\
33875|Format: \
33876|Only Active applet\
33877|All applets in view\
33878|Output file name\
33879|Browse...\
33880|And paste output to clipboard\
33881|And open file\
```

Translate the required strings and remove any untranslated strings as follows:

```
33873|Archi vo del imi tado por tabul aci ones\
33874|Exportar\
33875|Formato: \
33876|Só lo el applet acti vo\
33877|Todos los applets de la vi sta\
33878|Nombre del &archi vo de sal ida: \
33880|Y pegar sal ida en el &portapapel es\
33881|Y &abri r el archi ve
```

NOTE: Strings 33872 and 33879 were removed from the userstr.txt, because they were not required to be translated.

- 5 Save the file as userstr.txt in a UNICODE file format and then copy back into the ApplicationDataDirectory on the handheld device.

- 6 After restarting the Siebel Handheld application, the translated strings are read from userstr.txt in place of the strings included in the binaries of the Handheld application.

Distributing Translated Strings

When you have generated and tested the userstr.txt file of translated strings, include this file in the installation media for the Siebel Handheld application.

To distribute translated strings

- Create a new language sub-directory named UserDefinedLanguage, within the installation directory of your Siebel Handheld application, from which user devices are installed and which contains the setup.exe.

This directory with the name UserDefinedLanguage exists alongside the language sub directories shipped with the application. The new userstr.txt file must be placed in this subdirectory.

Upon installation of the Handheld application, the custom file will be placed in the ApplicationDataDirectory of the destination device along with the usual Handheld application files.

Configuring Printing from the Siebel Handheld Application

Users can print from any view in the Siebel Handheld application. However, they are not necessarily printing what they see on their screen. They can print a portion of the data they view or data that is not viewable at the time.

A print template defines the document that is to be printed. After the template has been defined, it must be associated with an applet in a view.

The following procedures are related to configuring printing from the Siebel Handheld application:

- [“Defining Documents for Siebel Handheld Applications” on page 84](#)
- [“Designing Print Applets for Siebel Handheld Applications” on page 84](#)
- [“Configuring Print Applets for the Siebel Handheld Application” on page 85](#)
- [“Configuring Print Buttons for the Siebel Handheld Application” on page 85](#)
- [“Creating Print Templates for the Siebel Handheld Application” on page 86](#)
- [“Configuring a Printer Settings Dialog Box for the Siebel Handheld Application” on page 86](#)
- [“Configuring Multiple Form and Multiple Quantity Printing for the Siebel Handheld Application” on page 87](#)

Defining Documents for Siebel Handheld Applications

Defining documents for printing includes several steps:

- Determine the documents your users need to print.
- Determine the views from which they are most likely to print the documents.
- Lay out the document for printing.

Determining which documents users will need and the views from which they are most likely to print requires that you have a good understanding of the users' day-to-day work.

Whether the existing applets in a view include most of the required data or only a small portion is a secondary consideration. It is not likely that a printout of the existing applets, which are formatted for an electronic PDA interface, will provide an acceptable or usable printed document. In most instances, you need to create additional applets that are specifically used for printing data.

The printed document will be composed of several applets which are added to the view. The applets pull the data from the underlying business components. These applets, in turn, direct the data to the print templates discussed in [Appendix E, "Print Tagging Language."](#)

Designing Print Applets for Siebel Handheld Applications

The print template references applets associated with the view and is how you specify what data appears in the printed document. For information on print templates, see ["Creating Print Templates for the Siebel Handheld Application" on page 86](#). As you design your applets, be aware of the print specifications of the printers used in the field. Portable printers typically print on two-inch or four-inch wide paper. You need to design your applets so that the data fits within the limits of the paper width. Keep your documents simple so that they work with different sizes of paper. The following are some guidelines to use when designing your applets to accommodate the smaller paper sizes.

Form Applets

Data from only one business component can be displayed on each line in your document. When you design your form applets, keep in mind that you can display a maximum of two columns of data in form applets. You can specify that each column of data is preceded by an optional caption column. This means you can have up to four columns displayed in your document—two data columns and two caption columns. Alternatively, you can have no captions, and simply display one or two columns of data.

The ratio between caption and data columns is, respectively, 35 percent and 65 percent. This is a fixed relationship that cannot be customized in the template.

Limitations on multiple columns in printed output are largely hardware dependent. If printing seven fields on a 40-column printer, the physical limitations allow only 4-5 characters for each field $(40 - (7-1))/7 = 4.85$. To print more than four columns, an 80-column printer is recommended. Generally, the capabilities of impact printers are lower than those of thermal printers.

List Applets

When specifying the width of a column, you can either specify it with a unit of measurement or as a percentage of the total page width. For example, you can specify that the first column is 20 mm, and the second column is 40 mm. If the width of the paper you print to is narrower than 60 mm, the column widths automatically adjust to fit the printable width as a proportion of the specified width. Therefore, the first column will always be 33 percent, and the second column will be 67 percent of the paper width. The minimum width for a column is 8 mm, which includes a 3 mm gutter margin between columns.

You can specify any number of columns in your list applet. However, if the width of the column is less than 8 mm, the application ignores the column and does not print it.

On narrow paper widths, all the data from a field might not fit on one line. Data that does not fit can be specified, in the Applet tag, to wrap to the next line and keep wrapping until the data in that field is completely displayed. The printable width of the page is equal to the paper width minus the left and right margins ($\text{PaperWidth} - [\text{LeftMargin} + \text{RightMargin}]$). These parameters are set in the setup.ini file. For more information about setting these parameters, see [Appendix F, "Print Configuration Settings."](#)

Configuring Print Applets for the Siebel Handheld Application

Configure print applets as you do any other applet using Siebel Tools.

To set the properties for each applet in the view:

- 1 Set the HTML Popup Dimension value to 0x0.

The 0x0 setting hides the applet from the user who must not see the applet that is used to generate the document. Exposure of these applets complicates a product that is tightly configured for an effective and efficient workflow on a small mobile device.

NOTE: For compatibility with previous versions, Oracle continues to support the name, `Popup Dimension`, for this property. However, it is recommended that you use `HTML Popup Dimension`.

- 2 Set the name of the Mail Template to be the name of the print template file. For more information about the print template file, see ["Creating Print Templates for the Siebel Handheld Application" on page 86](#). For more information about the Mail Template property, see ["Views with Associated Print Templates or Reports" on page 34](#).

Configuring Print Buttons for the Siebel Handheld Application

Users can use the application-level menu to print documents. For information about how to use the application-level menu, see the online help for the Siebel Handheld application.

You can also include a Print button on the applet to give users a visual reminder that it is possible to print from a particular view.

When you design your applets, keep the following in mind when deciding where to place the print button:

- Add the print button to the applet that is visible to the user. Do not add the print button to the print applet, which the user never sees.

- Consider what the desired behavior is when determining the placement of a print button.

For example, in a parent-child view, you might want to add the print button to the parent applet to ensure that the print button is always enabled. This is not necessarily true if the print button is placed on the child list applet. If there are no items in the list applet, the print button is disabled, and the user cannot print from that view. If you only want the user to print when there are list items, then placing the button on the child list applet is appropriate.

To configure a print button

- 1 From Siebel Tools, add a control to the print applet and specify the following:

HTML Type: Button

Method Invoked: Print

- 2 Set the Display Name property of the button. Generally, this is set to Print.

Creating Print Templates for the Siebel Handheld Application

Once you have identified and created the necessary views, you are ready to create print templates. A print template is a text file that includes instructions for printing a document. It specifies the data to be printed, document layout options, and text formatting. These instructions are specified using a print tagging language that is described in [Appendix E, "Print Tagging Language."](#)

For more information about print templates, see ["Views with Associated Print Templates or Reports" on page 34.](#)

Configuring a Printer Settings Dialog Box for the Siebel Handheld Application

If you want, you can configure a printer settings dialog box to popup once printing is initiated by a user in your Siebel Handheld application. The printer settings dialog box prompts the user to specify printer settings such as printer type, port, paper size, and page orientation.

If you do not configure a printer settings dialog box for your Handheld application, then you must manually configure the printer settings for your Handheld application in the setup.ini registry file. Print configuration settings are described in more detail in [Appendix F, "Print Configuration Settings."](#)

NOTE: The print settings that the user selects in the printer settings dialog box override those set in the registry (setup.ini file). The print settings specified via tags in the print template override those set in the registry or selected by the user in the printer settings dialog box.

To configure a printer settings dialog box

- 1 In the installation directory of your Handheld application, open the setup.ini file.
- 2 In the [Printing] Section of your setup.ini file, add PrinterUserSelect=1 or PrinterUserSelect=2.
PrinterUserSelect=1 - displays a printer dialog box for each copy of each form if multi-form, multi-copy printing is used.
PrinterUserSelect=2 - displays a printer dialog box only once when a form starts printing; the same settings are used for all copies of all forms if multi-form, multi-copy printing is used.
- 3 Install and synchronize your Handheld application. For more information about installing and synchronizing your Handheld application, see ["Process of Installing the Siebel Handheld Application" on page 175](#) and ["Synchronizing Data on Your Handheld Device" on page 214](#).

Configuring Multiple Form and Multiple Quantity Printing for the Siebel Handheld Application

Where multiple print templates or forms are associated with an applet, you can configure your Siebel Handheld application so that the user is given the option of specifying the number of copies, if any, to print for each form associated with that applet.

User can select 0 (to not print that form) or specify more than 0 by clicking the arrow keys.

To configure multi-form, multi-quantity printing

- 1 Configure print applets as you would any other applet using Siebel Tools.
 - 2 Add the print applets to the View from which you want to print.
 - a From the Objects Explorer in Siebel Tools, select View, [My Print View], View Web Template, Base, then the View Web Template Item, where *[My Print view]* is the name of the view from which you want to invoke printing.
 - b In the View Web Template Items List, add a record for each new print applet, specifying the value for Applet by selecting the new print applets you created in [Step 1](#).
- NOTE:** Specify values for Item Identifier to be greater than 2 to ensure that print applets are not visible in the user interface.
- 3 Create print templates as specified in ["Creating Print Templates for the Siebel Handheld Application" on page 86](#).
 - 4 Configure a print button using the PrintBatch method as follows:

- a From the Objects Explorer in Siebel Tools, select Applet, [My Print Applet], then Control, where [My Print Applet] is the applet upon which you want the Print button to appear.
- b In the Control List, add a new record with the following properties:
Name = Print
HTML Type = Button
Method Invoked = PrintBatch
- NOTE:** If the applet already has a print button configured based on previous functionality, just edit the Method Invoked property for that button to PrintBatch.
- c From the Objects Explorer in Siebel Tools, select Applet, [My Print Applet], then Applet User Prop, where [My Print Applet] is the same applet that you specified in part a of [Step 4](#).
- d In the Applet User Properties List, add a PrintForms user property. For more information about this property, see [PrintForms User Property on page 318](#).

Install and synchronize your Siebel Handheld application. For more information about how to install and synchronize your Siebel Handheld application, see [“Process of Installing the Siebel Handheld Application” on page 175](#) and [“Synchronizing Data on Your Handheld Device” on page 214](#).

Configuring Siebel Handheld Applications to Open Other Applications

You can open other applications from hyperlinks within Siebel Handheld applications. To configure a field to be a hyperlink, you must specify the field type as FT_URL in Siebel Tools. For more information about how to configure a field, see [Configuring Siebel Business Applications](#).

When a user taps a hyperlinked field, the content of the field is treated as a URL or as an attachment. Field values that start with *www* or *http* are automatically treated as URLs, otherwise they are treated as attachment files.

You must also configure the application user property, BrowserPath. You must specify the path to the application that you want to use to open a document in a URL field. The BrowserPath user property specifies the location of the executable file for the application in which to open attachments and URLs. For more information about how to use the BrowserPath user property, see [BrowserPath User Property on page 311](#).

The AttachmentDir setting specifies the storage location for your attachment files. For more information about AttachmentDir, see [“Configuring the Attachment Storage Location and Storage Capacity” on page 190](#).

NOTE: Field values are added to the start of the AttachmentDir registry value.

To open an application within a Siebel Handheld application

- 1 Set the BrowserPath application user property to, for example, \windows\pword.exe in Siebel Tools.

- 2 Set the registry setting AttachmentDir to, for example, \Program Files\Siebel Handheld\Attachment.
- 3 From the Siebel Handheld application on your handheld device, tap the URL or attachment to open the file in Pocket Word.

In this example, the field is displayed as a hyperlink. If the value of the URL field is *test.doc*, when the user taps the field value, the file \Program Files\Siebel Handheld\Attachment\test.doc is opened. The file is opened in Pocket Word, that is, \windows\pword.exe.

Configuring Signature Capture Capabilities in Siebel Handheld Applications

You can configure the Siebel Handheld application to capture, view, and print signatures. The following procedures are related to configuring signature capture:

- [“Enabling Signature Capture” on page 89](#)
- [“Invoking Signature Capture Dialog Box and Display” on page 90](#)
- [“Signature Capture Dialog Box and Signature Display” on page 91](#)
- [“Example of Configuring Signature Capture in Activities More Info View in Siebel Handheld Applications” on page 91](#)
- [“Adding a Signature Capture Button in Siebel Tools” on page 93](#)
- [“User Properties for Signature Capture Applets” on page 94](#)

NOTE: This feature allows customers to develop their custom signature capture capability on the Siebel Handheld application. It cannot be used to edit or change the Siebel Pharma Handheld signature capture preconfigured functionality, which allows you to capture signatures for samples disbursed. For more information about configuring Siebel Pharma Handheld signature capture for samples disbursed, see [“User Properties for Signature Capture Verification” on page 147](#).

Enabling Signature Capture

Enabling signature capture requires implementing the required supporting field and column definitions, adding a signature control to the applet and, if required, configuring an applet button through which the signature capture dialog box can be invoked. A field can be configured as a signature control in any list or form applet. Multiple signature controls can be added to a single applet. Once captured, signatures are converted to an ASCII encoding data and stored with the Siebel database as ASCII text.

When enabling Signature Capture, the following basic configuration is required:

- The Business Component field mapped to the signature control must be of Type DTYPE_NOTE.

- The underlying Table Column on which the Business Component field is based on Physical Type LONG.
- The Applet Control must have a control user property called Signature with a value Y. The Control's HTML Type property must be set to Field.

The Applet Control, by default, does not redisplay the signature data if there is data in the underlying business component field. In this case, the field displays as read-only and you are not able to see the signature that was captured.

To redisplay signatures, add another Control user property called ShowData with a value Y. With this additional configuration, the control is still read-only, but you are able to see the signature that was captured. For a full list of user properties available for configuring signature capture, see [“User Properties for Signature Capture Applets” on page 94](#).

Invoking Signature Capture Dialog Box and Display

There are two ways to invoke the Signature Capture dialog box or Signature Display. By default, the basic configuration specified in [“Enabling Signature Capture” on page 89](#) allows you to invoke the signature capture dialog box or display a signature using the (...) ellipses button.

To use a basic configuration to invoke the Signature Capture dialog box or display with the Applet Signature button follow this procedure:

- 1 Create a new Control with properties HTML Type = MiniButton, Method Invoked = StartSignature
- 2 Add a Control user property to the MiniButton control to indicate the signature field in the business component (for example, the field that is being used to store signature in the business component on which the applet is based).

For examples, see [“Example of Configuring Signature Capture in Activities More Info View in Siebel Handheld Applications” on page 91](#) and [“Adding a Signature Capture Button in Siebel Tools” on page 93](#).

Signature Capture Dialog Box and Signature Display

The Signature Capture dialog or display box consists of three controls in the client area: header, signature, and footer. A toolbar displays various buttons depending on the existence of signature data.

Initially, when signature data does not exist or has not yet been captured, the user sees a single line of text within the field control that reads Not Signed. Once the signature capture dialog box or signature display is invoked, the Clear and Cancel buttons are enabled, while the Accept button is disabled. Once the Signature area within the dialog box is populated, the Accept button is automatically enabled. [Table 7](#) describes the functionality around these three buttons.

Table 7. Signature Capture Dialog Box Buttons

| Button | Description |
|--------|---|
| Accept | Tap this button to commit the signature to the database. |
| Clear | Tap this button to reset and clear the signature field. |
| Cancel | Tap this button to discard the signature and close the display. |

Once the signature is committed (for example, signature captured and accepted in the database), the user sees a single line of text *x Signed x* within the applet field control.

The Header and Footer is a line of text in one or more paragraphs. The specific Header or Footer values that appear as well as how they appear in the dialog box is controlled by a set of user properties. See [“User Properties for Signature Capture Applets” on page 94](#).

Example of Configuring Signature Capture in Activities More Info View in Siebel Handheld Applications

The following example provides detailed steps to configure signature capture in Activities More Info View in the SIA Siebel Sales Handheld application. The Activities More Info View has one applet 'SIS HH Activity Entry Applet - CE', which is based on the Action Business Component, which in turn is based on S_EVT_ACT Table. The following example shows how to configure a field within the Action Business Component called Signature, which is based on a new column created from the S_EVT_ACT_X Table.

To enable basic signature capture (for example, no header or footer), follow these steps:

- 1 In Siebel Tools, add an extension column to the table S_EVT_ACT_X.

| For this Field | Add This Value |
|----------------|----------------|
| Name | SIGNATURE |
| User Name | Signature |

| For this Field | Add This Value |
|----------------------|---|
| Type | Extension |
| Nullable | TRUE |
| Physical Type | Long |
| Length | 0 |
| Cascade Clear | Ignore |
| Transaction Log Code | TRUE |
| Status | Active |
| Sequence | 999 |
| Comments | Extension Column for Activity Signature Capture |

- 2 Add a field to Action the business component with the following values:

| For this Field | Add This Value |
|----------------|--|
| Name | Signature |
| Join | X_EVT_ACT_X |
| Column | SIGNATURE |
| Text Length | 16,383 |
| Type | DTYPE_NOTE |
| Comments | Custom Field added for Signature Capture |

- 3 Add a control to the applet. For this example, you add the control to the 'SIS HH Activity Entry Applet - CE'.

| For this Field | Add This Value |
|--------------------------|------------------------------|
| Name | Signature |
| Caption | Signature |
| Caption String Reference | SBL_SIGNATURE-1005113048-35Y |
| Field | Signature |
| HTML Display Mode | EncodeData |
| HTML Row Sensitivity | TRUE |
| HTML Type | Field |
| Runtime | TRUE |
| Text Alignment | Left |

| For this Field | Add This Value |
|----------------|---|
| Visible | TRUE |
| Comments | Custom control added to display and invoke Signature Capture. |

- 4 Add two Control User Properties to the new 'Signature' control with the following values:

| User Property Name | Value |
|--------------------|-------|
| Signature | Y |
| Show Data | Y |

- 5 Add the new control to the Edit Web Template for SIS HH Activity Entry Applet - CE.

You can now invoke the signature capture dialog box by tapping the ellipsis points (...) on the signature field of the applet.

Adding a Signature Capture Button in Siebel Tools

Optionally, you can also configure the signature capture dialog box by placing a button on the applet. The following configuration steps provide details on doing this.

To create a new minibutton control

- 1 Create a new MiniButton control against the 'SIS HH Activity Entry Applet - CE'.

| For this Field | Add This Value |
|--------------------------|--|
| Name | SignatureCapture |
| Caption | Sign |
| Caption String Reference | SBL_SIGN-1005113047-35W |
| HTML Type | MiniButton |
| MethodInvoked | StartSignature |
| Comments | Custom MiniButton to Invoke Signature Capture. |

- 2 Add a Control user property to the MiniButton control to indicate the Signature field in the business component:

Name: SignatureField

Value: Signature

User Properties for Signature Capture Applets

You can use user properties to modify the style and other features of the signature capture applet. For more information about these user properties, see the following sections:

- [BorderStyle User Property on page 310](#)
- [EncryptKey User Property on page 313](#)
- [InkColor User Property on page 315](#)
- [InkWidth User Property on page 315](#)
- [ShowData User Property on page 323](#)
- [Signature User Property on page 324](#)
- [WritingGuide User Property on page 332](#)

You can use other user properties to modify the header and footer display and the save function. For more information about these user properties, see the following sections:

- [ConfirmAccept User Property on page 311](#)
- [MaximumFooterLines User Property on page 316](#)
- [MaximumHeaderLines User Property on page 316](#)
- [SignatureDataField User Property on page 324](#)
- [Signature Date User Property on page 323](#)
- [SignatureFooterField User Property on page 324](#)
- [SignatureFooterSaveField User Property on page 324](#)
- [SignatureHeaderField User Property on page 325](#)
- [SignatureHeaderSaveField User Property on page 325](#)
- [SignatureHeight User Property on page 325](#)
- [Signature Time User Property on page 323](#)

Testing the Siebel Handheld Application

Before deploying your application to your users, be sure to thoroughly test the application by synchronizing with a handheld device.

To test your Siebel Handheld application

- 1 Download the configuration for your Siebel Handheld application to a handheld device so you can test the functionality and verify that you have the desired behavior.

- 2 Check to see how much data is downloaded to your handheld device.

If too many records are downloaded, this increases the length of time it takes to synchronize the handheld device and slows down the application performance.

For more information about checking to see how much data is downloaded to your handheld device, see [“Monitoring Synchronization Sessions for Siebel Handheld Applications” on page 135](#), including the procedure [“To monitor data download” on page 135](#).

NOTE: You can also check the log file on the Siebel Server to see how many records are downloaded for each business component. However, you must turn on logging to capture this information in the log file. For more information about enabling logging for Direct Server Sync, see [“Configuring Logging Levels for Direct Server Synchronization” on page 171](#). For more information about enabling logging for Companion Synchronization, see [“Configuring Logging Levels for Companion Synchronization” on page 172](#).

4

Siebel Handheld Application Administration

This chapter describes how to administer Siebel Handheld applications. The following topics are covered in this chapter:

- [“Overview of Siebel Handheld Application Administration Screens” on page 97](#)
- [“Setting Up Users in Siebel Handheld Applications” on page 99](#)
- [“About Thread Throttling in Siebel Handheld Applications” on page 103](#)
- [“About Settings in the Administration Screens of Siebel Handheld Applications” on page 104](#)
- [“Configuring Application Settings for Siebel Handheld Applications” on page 105](#)
- [“Configuring Component Level Settings for Siebel Handheld Applications” on page 111](#)
- [“Administering Barcode Settings for Siebel Handheld Applications” on page 112](#)
- [“About BatchSync for Siebel Handheld Applications” on page 124](#)
- [“Process for Administering Batch Sync for Siebel Handheld Applications” on page 125](#)
- [“Load Balancing Support for Synchronization” on page 130](#)
- [“Setting the ApplicationDirectory Parameter” on page 131](#)
- [“Enabling Encryption for Siebel Handheld Applications” on page 131](#)
- [“Storing Asynchronous Data” on page 133](#)
- [“About Monitoring Synchronization Sessions for Siebel Handheld Applications” on page 134](#)
- [“Organization Enabled LOV Support” on page 138](#)
- [“Process of Administering Siebel Pharma Handheld Applications” on page 142](#)

Overview of Siebel Handheld Application Administration Screens

This chapter is designed to show you how to administer Siebel Handheld applications. Key tasks are outlined that you can perform using the Administration - Mobile screen. Your organization might follow a different task sequence depending on its business requirements.

Following are the views used for administering Handheld applications. You can access the Administration - Mobile screen through either the Siebel Web Client or Siebel Mobile Web Client:

- **Session Administration.** This view contains a detailed information about user synchronization sessions. Use this view to monitor and investigate Handheld application user synchronization activities.

- **Server Component Administration.** This view allows you to manage server-level settings for synchronization components. These settings are used by the synchronization engine to extract application definition and user data. These settings will override the application-level settings.
- **Application Administration.** This view lists all the Handheld applications available within your enterprise. From this view, you can create new application settings and administer settings that have already been created. The contents of the following views are used by the synchronization engine to extract application definition and user data:
 - **Business Component Filters.** Use this view tab to change synchronization filter settings for business components. If you wish to restrict the contents of selected business components to reduce the data set otherwise visible to your users, you can change filter settings in this view.
 - **Business Object Filters.** Use this view tab to define business object level filters, which restrict data that is downloaded to the Siebel Handheld Client upon synchronization.
 - **Settings.** Use this view tab to define parameter settings for your application.
- **Conflict Administration.** This view allows you to access selected transaction errors.
 - **Audit Trail.** Use this view tab to obtain information about action taken on a particular data conflict. Audit Trail must first be enabled for the conflict data. The business components are Handheld Transaction Error and Handheld Txn Error Message.
 - **Business Components.** Use this view tab to see business component information associated with a particular data conflict.
 - **Error Details.** Use this view tab to get detailed information about specific data conflicts. You are also able to take particular action on any given conflict.
- **Barcode Administration.** This view allows you to create new barcode definitions for your Handheld applications.
- **User Administration.** This view allows you to manage user-specific settings, such as business object and business components filters, that filter the data that display to the user. The data sets contained in these user views are used by the synchronization engine to extract application definitions and user data. The Siebel Handheld application is populated by these data when the user performs synchronization. These settings override the Siebel Server component-level and application-level settings.
 - **User Business Components.** Under this view tab, you can set user-specific synchronization filter directives for business components. These settings override application-level settings.
 - **User Business Objects.** Under this view tab, you can set user-specific filters for business objects. In this setting, you limit the data set that is downloaded, at the time of synchronization, to the Siebel Handheld Client. These settings override application-level settings.
 - **User Settings.** Under this view tab, you can control user-specific settings which override the Siebel Server component-level and application-level settings.
- **Batch Synchronization Administration.** In this view you can extract Siebel Handheld application databases, process-pending transactions, and apply the transactions to the Siebel Server.

- **Barcode Enabling.** In this view you can enable barcode scanning for chosen views in the Siebel Handheld application. For more information about enabling barcodes in your Siebel Handheld application, see *Siebel Sales Handheld Guide*.

NOTE: By default, most of the parameters referenced in this chapter are not automatically available as options in the drop-down menus of your Handheld application, and as a result are not set in the default configuration. If you want to make these and any other new parameters available, then you must add them to your Handheld application by creating a new parameter record for each parameter you want to make available.

Setting Up Users in Siebel Handheld Applications

To set up users in a Siebel Handheld application, you must perform a number of tasks. The tasks that you must perform include the following:

- [“Assigning User Responsibilities to Siebel Handheld Application Users” on page 99](#)
- [“Creating Siebel Handheld Application Views” on page 99](#)
- [“Setting Business Component Filters for Siebel Handheld Application Users” on page 101](#)
- [“Setting Business Object Filters for Siebel Handheld Application Users” on page 102](#)

Assigning User Responsibilities to Siebel Handheld Application Users

You must create user accounts and assign responsibilities and passwords for each user of the Handheld application. The steps are the same as those for creating user accounts, assigning responsibilities and passwords for the Siebel Web Client. For more information about creating user accounts, and assigning user responsibilities and passwords, see the *Applications Administration Guide*.

Creating Siebel Handheld Application Views

The synchronization engine for the Handheld application extracts views based on configuration (SRF) and the responsibility set up in the Application Administration view.

To configure a new view in your Handheld application, follow the list below. Knowledge of Siebel Tools is required to perform these tasks.

- 1 Create the new view in Siebel Tools. For information on creating views, see *Configuring Siebel Business Applications*.
- 2 Add that view to the Handheld application screen. For more information about adding views to screens, see *Configuring Siebel Business Applications*.

- 3 Add the screen to the Handheld application, if required. For more information about adding screens to applications, see *Configuring Siebel Business Applications*.
- 4 Compile the project. For more information about compiling projects, see *Using Siebel Tools*.
- 5 Add the new view to the Handheld application user's responsibility. For more information about adding views to a user's responsibility, see ["To add a Siebel Handheld application view to a user's responsibility" on page 100](#).

To add a Siebel Handheld application view to a user's responsibility

- 1 Launch the Siebel Web Client application that uses the same database that the Handheld application synchronizes with.
- 2 Navigate to the Administration - Application screen, then the Views view.
- 3 Add the new views.
- 4 In the Responsibilities view, associate each view with the appropriate Handheld application responsibility.
- 5 Add appropriate users to the responsibility.

NOTE: For more information about adding views and associating views with responsibilities, see *Applications Administration Guide*.

Specifying Views for the Siebel Handheld Application

The Handheld application uses a small subset of views used in the Siebel enterprise application. Because there are memory resource constraints on handheld devices, it is recommended that you compile an SRF to extract only those views that are used on the device.

The combination of the SRF and the user's responsibility determines which views are extracted. The handheld device only extracts views whose parent screen is included in the Handheld application and are part of the user's responsibility.

About Business Component Filters

Business component filters are used in some cases to restrict the size of extractions that otherwise might be excessive. They suppress the extraction of business component records. Use business component filters with discretion because excessive filtering can cause server data integrity problems.

Business component filters can be specified at either the application level or the user level, but not at the component level. They must not be specified at the user level outside of your test environment, and when tested they must be migrated to the application level.

If you are configuring synchronization for Handheld applications, a suggested practice, is to start with no business component filters at all. Attempt a synchronization and even if it fails due to excessive data extraction volumes, view the extraction results to determine the business components for which the data extraction was excessive. If any business component extraction is found to be excessive, you can reduce the extraction by adjusting the existing search specs specified in Siebel Tools (applets and business components), or Handheld application settings (for example, in DefaultBusObj), and selection filters (business object filters). If nothing else works, a business component filter can be used to force a reduction in the result set for a particular business component.

Setting Business Component Filters for Siebel Handheld Application Users

The Business Components Filters view tab in the Application Administration view contains the synchronization filter settings for business components. If you wish to add or remove business component filter settings for users, you do so in this view.

To add a business component filter setting

- 1 Navigate to the Administration - Mobile screen, then the Application Administration view.
- 2 In the Handheld Applications list, select the appropriate application, then click the Business Component Filters view tab.
- 3 Click New, then enter the required information.

Some fields in the Business Component Filters view are described in the following table:

| Field | Comments |
|--------------------|---|
| Owner | Read only. Contains the Handheld application to which the business component will be associated. For example, Siebel <Application name> for CE. |
| Business Component | A text box. Enter the business component name. |
| Name | A text box. Enter the business component filter name. For example, Filter 1. |
| Query | A text box. Enter the syntax for the query. For example, [Start Date] <=Today() For more information about developing queries, see "Data Filtering for Siebel Handheld Applications" on page 159 |

To remove a business component filter setting

- 1 Navigate to the Administration - Mobile screen, then the Application Administration view.
- 2 In the Handheld Applications list, select the appropriate application, then click the Business Component Filters view tab.
- 3 Select the filter setting record you wish to remove.

- 4 From the Business Component Filters menu, choose Delete Record.

About Business Object Filters

The Siebel Web Client supports named queries on most or all screens. These queries restrict the result set displayed to the user. Business object filters are the Handheld application synchronization equivalent to these named queries. Just like client based queries, they are applied to a specified business object. They also contain search or sort specifications for business components that are part of that business object.

The Default column is the only difference from the query syntax appearing in a BusObject filter for the Siebel Web Client. The Default column is unique to the business object filter mechanism. For Web client queries, users can set a default query for each view in their Preferences screen. If users do not set a default query for each view in their Preferences screen, there is no default query, and so all records are returned by default. This is not recommended for Handheld application deployments as this might result in excessive volumes of data being extracted. Thus, the Default column allows you to decide what filter to use by default if a user does not make a selection.

There is also a client option to revert to default filters selection. There must be at most a single default filter for each business object name. It is not recommended that you set more than one filter as the default for a particular business object. Note however, that the same default choice will be made by all clients.

Business object filters can be specified at either the application level or the user level, but not at the component level. Business object filters specified at the user level override a filter specified at the application level, provided the application-level filter has the identical values for the business object name and filter name. The list of filter choices presented to the user is the merged list of all application filters except those overridden at the user level, plus all user filters.

Setting Business Object Filters for Siebel Handheld Application Users

The Business Object Filters View in the Application Administration view contains the synchronization filter settings for business objects. If you wish to add or remove business object filter settings for users, you do so in this view.

To add a business object filter setting

- 1 Navigate to the Administration - Mobile screen, then the Application Administration view.
- 2 In the Handheld Applications list, select the appropriate application, and then click the Business Object Filters view tab.
- 3 Click New, and then enter the required information.

- 4 The fields in the Business Object Filters view are described in the following table:

| Field | Comments |
|-----------------|---|
| Owner | Read only. Contains the Handheld application to which the business component will be associated. For example, Siebel <Application Name> for CE. |
| Default | Check to allow you to decide what filter will be used by default if a user does not make a selection. |
| Business Object | A text box. Enter the business object name. For example, Account. |
| Name | A text box. Enter the business object filter name. For example, BOFilter 1. |
| Query | <p>A text box. Enter the syntax for the query, for example, as follows:</p> <pre>' Account' . Search = "[Account Status] = LookupValue ('ACCOUNT_STATUS'", "'Active'")" ' Account' . Sort = "Name, Location"</pre> <p>For more information about developing queries, see “Data Filtering for Siebel Handheld Applications.”</p> |

To remove a business object filter setting

- 1 Navigate to the Administration - Mobile screen, then the Application Administration view.
- 2 In the Handheld Applications list, select the appropriate application, and then click the Business Object Filters view tab.
- 3 Select the filter setting record wish to remove.
- 4 From the Business Object Filters menu, choose Delete Record.

About Thread Throttling in Siebel Handheld Applications

When a large number of users attempt to synchronize simultaneously, the load on a single server can become excessive. Thread throttling significantly reduces the total number of simultaneously active transaction processor threads and database extraction threads, thereby increasing the overall throughput for synchronization, without reducing the number of concurrent users. Synchronization performance degrades slightly but will successfully complete for all users.

Thread throttling allows the maximum possible number of concurrent users. It throttles in a manner that maximizes application server usage during any mixture of transaction processor and database extraction threads. Thus, if the Siebel Handheld Application Server supports x concurrent transaction processing threads, it will allow this number of transaction processing threads while disallowing all database extraction threads. Similarly, if the server supports y concurrent database extraction threads, it will allow this number of database extraction threads while disallowing all transaction processing threads. If fewer than x transaction processing threads are currently active, some database extraction threads are allowed to execute concurrently. Similarly, if fewer than y database extraction threads are currently active, some transaction processing threads are allowed to execute concurrently.

For information on thread throttling settings, see [“Configuring Application Settings for Siebel Handheld Applications” on page 105](#).

To enable thread throttling, see [“To add a new application setting” on page 111](#).

About Settings in the Administration Screens of Siebel Handheld Applications

Settings can appear at three levels: Application, Component, or User. Certain settings are only allowed at specific levels.

There are also two types of settings: single-line and multi-line settings:

- For a single-line setting, only one row can contain a unique setting name at a given level. Thus, a single-line setting at the User level can be used only once at that level. For single-line settings, the Sequence column must always contain a value of zero.

For single-line settings, you can leave the sequence column value alone, which will result in a default value of zero being assigned.

- For multi-line settings, the sequence column is necessary to allow multiple rows. The sequence column has no effect on actual processing, but is part of the logical key in the database table. When you create a new record in this database table, the record must have a unique logical key.

For multi-line settings, each line must include a different sequence number to avoid logical key conflicts.

If the combination of setting name and sequence number is not unique for a particular parent record (Application, Component, or User), you will receive an error message when stepping off or saving the record. Multi-line entries must contain the correct key values before they can be saved to the database.

The rules for override are that any setting, whether single line or multi-line, specified at the lower levels overrides that same setting name at higher levels. Thus the settings at the application level apply unless specific settings are overridden for a particular component or a particular user. When an override occurs, it occurs for the complete setting name. Thus, if a particular multi-line setting has three rows at the user level and twelve rows at the application level, the resulting multi-line setting value for that setting will contain only the three rows at the user level. There is also an override mechanism to allow a setting to be removed, rather than replaced. To remove a setting from a higher level, specify the setting name to be removed, with an empty setting value.

Configuring Application Settings for Siebel Handheld Applications

The Administration - Mobile screen contains a Settings list that allows you to change default settings. Some of the settings are described in [Table 8 on page 105](#).

The Handheld setting value field has a limit of 2000 characters. To create a default business object (DefaultBusObj) setting value with more than 2000 characters:

Break the value into multiple lines of DefaultBusObj settings. Enter these settings using the same DefaultBusObj value format:

```
BusObj |Vi si bi l i t y|Query|BusComp_1|query speci fi cati ons_1[|BusComp_n|query
speci fi cati ons_n]
```

When defining multiple lines of DefaultBusObj settings for a single business object, all view modes and filters must match. If a business component is found in multiple DefaultBusObj settings for a single business object, then the business component's query specifications will be concatenated with a space and appended to the previous business component's query specifications. You must be aware of this effect in order to decide if both query specifications must be combined with an OR or AND operative.

The DefaultBCUserProps and DefaultFields settings for a single business component can also be broken into multiple lines of settings if their values exceed 2000 characters. These settings must all be entered using the same value format as well:

```
BusComp|BCUserProp_1[|BCUserProp_n] or BusComp|Fi el d_1[|Fi el d_n]
```

Table 8. Settings for Siebel Handheld Applications

| Setting Name | Definition | Valid Values |
|-----------------------|---|------------------------------|
| AllowRememberPassword | Single-line setting. Specifies whether the Remember Password check box will be displayed on the Siebel Handheld Client Synchronization screen. | TRUE or FALSE |
| ApplicationDirectory | Single-line setting. Specifies the working directory (UNC path) for the Handheld application synchronization component where user data is stored. | \\server name\directory name |
| BatchDownloadStagger | Single-line setting. Enables the setting of the interval (in minutes) during which the polling for ready extracts is staggered. | integer >= 0; default is 30 |
| BatchInitialDownload | Single-line setting. Enables the setting of the earliest time (HH:MM; 24-hour clock) at which the polling for a database extract can commence. The setting must be UTC enabled. | 00:00 is the default value |

Table 8. Settings for Siebel Handheld Applications

| Setting Name | Definition | Valid Values |
|---------------------|--|-----------------------------|
| BatchPollFrequency | Single-line setting. Enables the setting of the amount of time (in minutes) between polling sessions. | integer >= 0; default is 15 |
| ClientSyncDefault | Determines the default synchronization options available to the synchronization client in the Handheld application. It can have the following values: <ul style="list-style-type: none"> ■ Upload which means that the Upload Only option is selected in the batch synchronization dialog on the client. ■ Deferred which means that the Upload Now, Auto Download Later option is selected. | Deferred, Upload |
| DBExtractThreadLoad | Single-line setting. Enables Thread Throttling functionality for database extract threads. It is the load (or weight) that will be assigned to each individual Database Extractor thread. For each running database extracting thread, this load value will be used to determine the total system load, which is then compared against the MaxTotalThreadLoads value. A new thread will only be started if the total system load does not exceed the MaxTotalThreadLoads value. A value of zero disables throttling of Database Extractor threads. | integer >= 0; default is 2 |
| DefaultBCUserProps | Multi-line setting. Reduces the number of User Properties that are extracted to the device metadata, thus decreasing the number of fields for a business component that need to be extracted. These reductions lower the metadata extract size and optimize client performance by removing irrelevant user properties from the Handheld application configuration. If this directive is not specified for a business component, then all user properties will be extracted. | |

Table 8. Settings for Siebel Handheld Applications

| Setting Name | Definition | Valid Values |
|---------------------------|--|--|
| DefaultBusObj | Multi-line setting. Used to force the extraction of a business object and its designated business components. This is typically used when a business component is not extracted because it is not specified via the "DefaultView" directive and is required for the application to function correctly. Another use of this directive is to designate business components that function as RBOs (reverse business objects). | Business Object Name Visibility Level Business Component Name QBE query construct For more information about QBE (query by example) structure, see "Constructing Business Component and Business Object Filters for Siebel Handheld Applications" on page 160. |
| DefaultFields | Multi-line setting. Used to force the extraction of fields for a particular business component. This is typically the case where the business component is extracted via the "DefaultView" directive but because the field is not visible on the applet, it is not extracted. Also when a hidden business component is extracted via the "DefaultBusObj" directive, it is necessary to specify all of the fields that need to be extracted for the business component. If no fields are specified via this directive, then the business component will not be extracted for this case. | Business Component Name Field1 Field2 ... |
| DefaultHandheldInsertMode | Single-line setting. Enables insert failure processing for synchronization. If enabled, synchronization will stop upon encountering a data conflict during an insert operation. Users or administrators must resolve the data conflict before synchronization can resume. This feature is disabled by default. | NoFail or FailOnError |

Table 8. Settings for Siebel Handheld Applications

| Setting Name | Definition | Valid Values |
|-----------------------------|---|-----------------------|
| DefaultHandheldSyncPickMode | Single-line setting. Enables Extended Pick processing for synchronization. If enabled, synchronization stops when a data conflict is encountered during a pick operation. Users or administrators must resolve the data conflict before the synchronization can resume. This feature is disabled by default. | NoFail or FailOnError |
| Enable Application Lockout | Single-line setting. If set to TRUE, locks the Siebel Handheld application after a number of invalid attempts. The number of invalid attempts is specified in the FailedLoginAttempts setting. The Siebel Handheld application locks for the number of days specified in the LockOutPeriod setting. The default value is FALSE. | TRUE or FALSE |
| Enable Encryption | Single-line setting. If set to TRUE, enables encryption of the database on the handheld device. The default value is FALSE. | TRUE or FALSE |
| EnableBatchExtraction | Single-line setting. This parameter enables or disables the batch synchronization functionality. Disable batch synchronization is the default. | TRUE or FALSE |
| EnableThrottleDebugMode | Single-line setting. This parameter enables or disables Thread Throttling functionality. | TRUE or FALSE |
| ExtractionExpiration | For batch synchronization functionality. It specifies the number of hours that is used to determine when a database extraction has expired. When a database extract expires, batch synchronization will reextract the databases. If no value is provided, then databases will be extracted every time they are requested. | integer >= 0 |

Table 8. Settings for Siebel Handheld Applications

| Setting Name | Definition | Valid Values |
|--------------------------|---|--|
| LoadBalanceLoc | Single-line setting. This setting can be specified only from the application level. The IP address or server name of the load balancer and port number are specified in the following format: [IP Address]: [port number] | Load balancing must be used for batch synchronization, if enabled. |
| MaxBatchsyncTasks | Single-line setting. This parameter enables a percentage of synchronization component tasks to be allocated to performing DBextract or Transaction Processing. | integer >= 0; default is 0.5 |
| MaxThrottleWaitTime | Single-line setting. For Thread Throttling functionality. This is the upper limit on the amount of time (in minutes) any individual throttling request will be allowed to wait. | integer >= 0; default is 5 |
| MaxTotalThreadLoads | Single-line setting. For Thread Throttling functionality. It defines the maximum number of total thread loads (determined by DBExtractThreadLoad and TranProcThreadLoad for each type of thread) that will be allowed to run across the scope defined by ThreadThrottleScope. If a throttling request is made that causes this maximum to be exceeded, the requesting thread is delayed. The default value is zero, which completely disables the throttling mechanism. | integer >= 0; default is 0 |
| NoDatabaseBusComp | Multi-line setting. Defines a business component that must be omitted from the device database and excluded from database extraction. | |
| NoExtractBusComp | Multi-line setting. Defines a business component that must be excluded from database extraction. | |
| SchemaChangeNotification | Multi-line setting. Defines a Business Service that must receive the Handheld Schema Change Notification. | (Siebel internal) |

Table 8. Settings for Siebel Handheld Applications

| Setting Name | Definition | Valid Values |
|---------------------|---|----------------------------|
| ThreadThrottleScope | <p>Single-line setting. For Thread Throttling functionality. Defines the scope of the thread throttling mechanism as using one of two different modes:</p> <ul style="list-style-type: none"> ■ Process Thread, where throttling occurs independently within each process. ■ Application, where throttling occurs across all processes running the same synchronization application on a single server. <p>Only the first character of the ThreadThrottleScope parameter is used to determine the value. All other characters are ignored. Thus <i>aardvark</i> is interpreted as <i>Application</i>. Case is also ignored. The default value is Application.</p> | Application or Process |
| TranProcThreadLoad | <p>Single-line setting. For Thread Throttling functionality. It is the load (or weight) that is assigned to each individual Transaction Processor thread. For each running transaction processing thread, this load value is used to determine the total system load, which is then compared against the MaxTotalThreadLoads value. A new thread will only be started if the total system load does not exceed the MaxTotalThreadLoads value. A value of zero disables throttling of transaction processing threads.</p> | integer >= 0; default is 3 |

Table 8. Settings for Siebel Handheld Applications

| Setting Name | Definition | Valid Values |
|----------------------|--|----------------------------|
| TranProcThreshold | Single-line setting. For Thread Throttling functionality. This is the minimum time (in minutes) that the Transaction Processor will continue processing transactions before performing a handshake with the client. Once this value has been exceeded, a handshake occurs instead of starting to process the next transaction. Doing so prevents the client from timing out the session when thread throttling is enabled. | integer >= 0; default is 5 |
| TranReapplyComponent | Single-line setting. Server component to use for playback of transactions for parent component. | |
| TranReapplyServer | Single-line setting. Siebel Application Server to use for playback of transactions for parent component. | |

To add a new application setting

- 1 Navigate to the Administration-Mobile screen, then the Application Administration view.
- 2 From the Settings list menu, select New Record.
- 3 Click the name field, and select the setting you require.
- 4 Click the Value field, and enter the text of the setting.

Configuring Component Level Settings for Siebel Handheld Applications

You administer server components for your handheld device from the Administration - Mobile screen, then the Server Component view. The component displayed is based on the application selected in the Application Administration view. The Server Component Administration screen is used for overriding application settings at the component level on a specific server. The data entered here is used by the synchronization engine when the user synchronizes with the Siebel Server.

NOTE: You can use this method for Direct Server Synchronization but not Companion Synchronization.

The Server Component screen contains a list of server components associated with the application that was selected in the Application Administration screen. For each server component, the Settings view details the associated settings.

The Name field value needs to be the exact component name, such as SalesceObjMgr_enu, of the synchronization server component for the Handheld application.

Server component settings are used primarily for functionality required for individual server components. For example, thread throttling is dependent on components running on a particular server and is best set at the Siebel Handheld Application Server component level. Examples of settings that might be made at the component level are MaxTotalThreadLoads, TranProcThreadLoad, DBExtractThreadLoad, and so on. Generally, settings made at the application level are shared by all server components.

Administering Barcode Settings for Siebel Handheld Applications

Siebel Handheld applications can scan and parse data from HIBC and UCC/EAN barcodes. Administrators can set up Siebel Handheld application to scan and parse any one-dimensional barcode and render the corresponding data in an applet. For example, if a control in a list applet is enabled for barcode scanning and the barcode scanned is valid, the Handheld application will parse the barcode, create a new record, and fill in the field with parsed data. In form applets, after the user scans an item, the data representing the item automatically populates the barcode-enabled field. In order to do this, Siebel Administrators must define the barcodes and how data will be parsed from these barcodes in the Barcode Administration view.

The following procedures are included in this topic:

- [“Enabling Barcode Scanning in Siebel Handheld Applications” on page 112](#)
- [“Setting New Record Behavior for Barcode Scans” on page 113](#)
- [“Mapping a Button for Socket In-Hand Scan Card” on page 114](#)
- [“Creating New Barcode Definitions for Siebel Handheld Applications” on page 114](#)
- [“Enabling Applications Views for Barcode Scanning” on page 119](#)
- [“Enhancing the Behavior of the Barcode Scan” on page 119](#)
- [“Example of Creating a New Barcode Definition in the Siebel Handheld Application” on page 122](#)

Enabling Barcode Scanning in Siebel Handheld Applications

Before using the barcode scanning feature, you must add the following parameter value to the setup.ini file:

```
EnableBarcode = Y
```

If EnableBarcode is not set on a handheld device which has a barcode scanner attached to it, then:

- When you install the Siebel Handheld application on that handheld device, and
- Synchronize with the Siebel Server and default filters

An error message similar to the following displays as the application opens:

The barcode scanner could not be initialized.

If your Siebel Handheld application is not enabled for barcode scanning, then you must create an Application Event Services object in Siebel Tools. For more information about enabling barcode scanning for your Siebel Handheld application, see [“To enable barcode scanning for a Handheld application” on page 113](#).

To enable barcode scanning for a Handheld application

- 1 Navigate to Application object in Siebel Tools, and select Siebel ePharma CE.
- 2 Expand to the Application Event Services object in Siebel Tools, and add a new entry with the following details:

| Field | Comments |
|--------------|--|
| Name | The description of the event. For example: SHCE Core Barcode Servi ce |
| Event Name | The name of the application event service. For example: HHCE_Start |
| Service Name | The name of the business service on which to invoke the method. For example: SHCE Core Barcode Servi ce |

- 3 Add the following parameter value to the setup.ini file for your Handheld application:

Enabl eBarcode = Y

Setting New Record Behavior for Barcode Scans

You can specify whether to create a new record when a user scans a barcode. Typically, you create a new record for a parseable scan, and update the active record for a non-parseable scan.

To set the new record behavior

- In Siebel Tools, set the applet user property BarcodeNewRecord in Siebel Tools. For more information about this user property, see [BarcodeNewRecord User Property on page 310](#).

Mapping a Button for Socket In-Hand Scan Card

Siebel Handheld applications support embedded barcode scanners provided in Symbol and Intermec devices. In addition, Handheld applications support certain Socket scan cards. See *Siebel System Requirements and Supported Platforms* on Oracle Technology Network for a list of supported handheld devices and socket scan cards. Note that different socket scan cards will require different scan card driver programs to be installed. Check with the card vendors or their Web sites for required driver program installation.

For Intermec and Symbol devices, the barcode scanning can be triggered by the built-in buttons. If you are using Socket scan cards, the following steps are required to be taken on the handheld device for mapping a button to the barcode-triggering utility software, SocketTrigger.exe.

To map a button to SocketTrigger.exe (Socket Scan Card only)

- 1 On your handheld device, open the application that enables you to customize the hardware button functions. For example, on Windows Mobile Version 5.0, choose Start, Settings , Personal, then Buttons.
- 2 From the button list, select a button to trigger In-Hand Scan Card.
- 3 In the button assignment field, select SocketTrigger.
- 4 Confirm your selection.

NOTE: The ^ character signifies a circumflex.

Creating New Barcode Definitions for Siebel Handheld Applications

You can define new one-dimensional barcodes in the Barcode Administration views. To define a new barcode, you need to create a new barcode definition in the Barcodes list, then create corresponding records in the Barcode Item list to define data elements within the barcode.

To create a new barcode definition

- 1 Navigate to the Administration - Mobile screen, then the Barcode Administration view.

- 2 Click New, then fill in the fields.

The fields in the Barcodes list are described in the following table:

| Field | Comments |
|--------------|--|
| Name | Required. Any descriptive name for the barcode. Must be unique for all records defined in the Barcode screen. |
| Sequence | Required. This is the sequence for barcode parsing. The lower of two sequence numbers takes precedence when deciding which barcode in the list to match. For example, if a scanned barcode matches two entries in the barcode list, the one with a lower sequence number will be picked for the match. The lowest sequence number is 1. |
| Barcode Type | Required. Type of barcode standard. Can either be Location or Code. For a Location type barcode, the barcode data element parsing is done solely by the Sequence field in Barcode Item list. For a Code type barcode, the barcode data is first parsed into multiple sections by the Code field in the Barcode Item list, then is further parsed by the Sequence field for each section. For more information about barcode type, see "Example of Location Type Barcode" on page 117 and "Example of Code Type Barcode" on page 118 . |
| Buscomp | Optional. Used to designate into which business component the target data is placed. |
| Min Length | Optional. The minimum length for the barcode data in digits. If it is not set, the default minimum length is 1. |
| Max Length | Optional. The maximum length for the barcode data in digits. If it is not set, the default maximum length is 10000. |
| Prefix | Optional. Defines the value that the parsing algorithm uses to identify barcode definition to use. The allowed format for Prefix is any ASCII character for exact match, '#' for any single digit number match (that is, 0 - 9), '&' for any alphabet character in lower and upper case, or '*' for a wild card match. For example, you can enter ##9&* as the prefix, which means the first two characters can be any number, the third character has to be '9', the fourth character can be any alphabet character, and the fifth character can be any ASCII character. |
| Suffix | Optional. Allows you to associate suffixes with individual fields. The allowed format for Suffix is the same as the one specified for Prefix. In addition, the rules for matching Suffix are the same as the ones for Prefix. |
| Description | Optional. Detailed description for this barcode. |

- 3 In the Barcode Item list, add new records to identify how data will be parsed from barcode.

Depending on the type of barcode standard you just defined (either Location or Code), you then need to create new records in the Barcode Item list to identify how data is parsed.

- 4 In the Barcode Item list, Click New, then update the required fields.

Some of the fields are described in the following table:

| Field | Comments |
|-------------|---|
| Name | Required. Name of the barcode item (for example, Product Name corresponds to the Product Name field). This name does not determine which field in BusComp the parsed data is written to. That field is determined by the Data Desc field. |
| Code | Applicable for Code barcodes only. Used to break a barcode into multiple barcode sections. |
| Sequence | Required. The sequence for barcode data elements parsing. For Location barcodes, this corresponds to the sequence used to parse data elements from the barcodes. For Code barcodes, this corresponds to the sequence used to parse data elements within barcode sections. |
| Min Length | Required. Minimum Length of the character string for this data element. If this is a variable length string, then Minimum Length is not equal to Maximum Length. |
| Max Length | Required. Maximum Length of the character string for this data element. If this is a variable length string, then Minimum Length is not equal to Maximum Length. |
| Data Desc | Optional. Used by algorithm to identify what field to parse this data into. The value in this field must correspond to the exact field name. For example, Product Name corresponds to the Product Name field. |
| Data Format | Optional. Defines format for data element (for example, MMDDYY is for date). |
| Action | Optional. Defines the action for the data element. Both Data Format and Action are written in the following format to the Data Desc field: <parsed data element>, <Data Format>, <Action> See the example in Table 12 . |

Example of Location Type Barcode

The two entries for the Barcode list, HIBC Secondary and IBM, are shown in [Table 9](#).

Table 9. Location Type Barcode List Entries

| Name | Sequence | Barcode Type | Buscomp | Minimum Length | Maximum Length | Prefix | Suffix | Description |
|----------------|----------|--------------|-----------------|----------------|----------------|-----------------|--------|-------------|
| HIBC Secondary | 1 | Location | Medical Product | | 31 | +\$9## ###3 | | |
| IBM | 2 | Location | Tech Product | | 30 | +\$9## ###39 | | |

Entries for Barcode - HIBC Secondary in the Barcode Item list are shown in [Table 10](#).

Table 10. Location Type Barcode Item List Entries

| Name | Code | Sequence | Minimum Length | Maximum Length | Data Description | Data Format | Action |
|---------------|------|----------|----------------|----------------|------------------|-------------|--------|
| Prefix | | 1 | 4 | 4 | Prefix | | |
| Quantity | | 2 | 5 | 5 | Quantity | | |
| Exp date flag | | 3 | 1 | 1 | Exp flag | | |
| Date | | 4 | 6 | 6 | Expiration Date | YYMMDD | |
| Lot | | 5 | 0 | 13 | Lot # | | |
| Link | | 6 | 1 | 1 | Link | | |
| Check | | 7 | 1 | 1 | Check | | |

If the barcode being scanned is +\$90010039509283C001LZ, both barcode entries HIBC Secondary and IBM are good matches. HIBC Secondary has a lower sequence as 1, it is picked as the final match for parsing the barcode. The data is parsed as follows:

```
{(Prefix, "+$9") (Quantity, "00100") (Exp flag, "3") (Expiration Date, "950928,YYMMDD") (Lot #, "3C001") (Link, "L") (Check, "Z")}
```

NOTE: In this example, the use of "Prefix" in the two tables is different. The Prefix column in the Barcode list table is required for matching the record in the Barcode list. The Prefix row in the Barcode Item list table is a user defined data element to be parsed for a particular barcode.

Example of Code Type Barcode

Entries for Code type barcode in the Barcode list are shown in [Table 11](#).

Table 11. Code Type Barcode List Entries

| Name | Sequence | Barcode Type | Buscomp | Minimum Length | Maximum Length | Prefix | Suffix | Description |
|---------|----------|--------------|-----------------|----------------|----------------|--------|--------|-------------|
| UCC/EAN | 1 | Code | Medical Product | | 37 | 01 | | |

Entries for Code type barcode in the Barcode Item list are shown in [Table 12](#).

Table 12. Code Type Barcode Item List Entries

| Name | Code | Sequence | Minimum Length | Maximum Length | Data Description | Data Format | Action |
|---------------------|------|----------|----------------|----------------|------------------|-------------|----------|
| AI 01 | 01 | 1 | 2 | 2 | AI 01 | | |
| Packaging Indicator | 01 | 2 | 1 | 1 | Indicator | | |
| UCC/EAN # | 01 | 3 | 12 | 12 | UCC/EAN | | |
| Link | 01 | 4 | 1 | 1 | Link | | |
| Quantity Section | 30 | 1 | 3 | 5 | Quantity Section | 1000 | Validate |
| AI 240 | 240 | 1 | 3 | 3 | | | |
| Product # | 240 | 2 | 1 | 13 | Product # | | |

The barcode UCC/EAN is parsed to three sections, 01, 30, and 240. If the barcode being scanned is **01103123456789033050240**ABCDEFG, the data is parsed as follows:

Parsed Data = {(AI 01, "01") (Indicator, "1") (UCC/EAN, "031234567890") (Link, "3") (Quantity section, "3050,1000,Validate") (Product #, "ABCDEFG")}

NOTE: The Code portions in the barcode appear in bold text.

Enabling Applications Views for Barcode Scanning

You can enable any existing or new view for barcode scanning.

To enable an application view for barcode scanning

- 1 Navigate to the Administration - Mobile screen, then the Barcode Enabling view.
- 2 Click New, then enter the necessary information.

The fields in the Barcode Enable list are described in the following table:

| Field | Comments |
|---------------|---|
| View | The view in the Handheld application from which the scan occurs. |
| Applet | The applet within the view from which the scan occurs. |
| Buscomp | The business component from which information is drawn. |
| Buscomp Field | The specific field within the business component that must be updated through barcode scanning. |

Enhancing the Behavior of the Barcode Scan

If your business process requires that a specific barcode logic be performed when you perform a barcode scan using a handheld device, then you can enhance the behavior of your barcode scan. Enhancing your barcode scan behavior reduces the time taken to perform routine tasks with the scanner and quickly finds the correct record that needs to be updated.

The default behavior of a barcode scan results in the creation of a new record or the update of an existing record. However, if you want to find and update a record at the same time without having to use multiple key strokes, then you must define a new user property at the applet level.

To enhance the barcode scan behavior

- 1 Create a new applet user property called TriggerFindRecord in Siebel Tools.
For more information about creating TriggerFindRecord, see [“Defining the TriggerFindRecord” on page 120](#).
- 2 Create your barcode definition and add records to it identifying how the data will be parsed from the barcode.
For more information about creating a barcode definition and adding records to it, see [“Creating New Barcode Definitions for Siebel Handheld Applications” on page 114](#).
- 3 Set your application views for barcode scanning.
For more information about enabling an application view for barcode scanning, see [“Enabling Applications Views for Barcode Scanning” on page 119](#).

Defining the TriggerFindRecord

To configure the behavior of your barcode scan, you must first define the user property TriggerFindRecord at the applet level in Siebel Tools. The syntax for TriggerFindRecord is as follows:

TriggerFindRecord = <Search fields> | <Activate field> | <Method> | <Action to apply>

The fields in TriggerFindRecord are described in [Table 13](#). Examples also follow [Table 13](#).

Table 13. TriggerFindRecord Fields

| Field | Definition |
|------------------|---|
| <Search fields> | Key fields used for searching. Search fields must be separated by a comma, for example, as follows: <Key field 1>, <Key field 2>, ... <Key field N> |
| <Activate field> | The name of the (focus) field that is activated in the UI after a record is found or added. |
| <Method> | This field defines what functionality needs to be performed on the barcode scan. Each applet can define only one method in the user property. The following methods (or process modes) are supported: <ul style="list-style-type: none"> ■ FindOnly—searches for matching records in a list based on key fields. If no record is found, no new record is created. If matching records are found, the first matching record is selected, and the specified <Activate Field> has focus. If no focus field is specified, users must click the field they want to update. ■ AddOnly—searches for a record in a list based on key fields. If no record is found, a new record is created. If matching records are found, the first matching record is selected, and the specified <Activate Field> has focus. If no focus field is specified, users must click the field they want to update. This method does not update any fields in matching records. ■ UpdateOnly—searches for a record in a list based on key fields. If no record is found, no new record is created. If matching records are found, the first matching record is selected, and its data is updated with the scanned data. Additionally, if <Action to apply> has been specified, the appropriate intelligent updates occur. If no focus field is specified, users must click the field they want to update. This method does not create new records if the scanned record does not exist in the list. ■ AddAndUpdate—searches for a record in a list based on key fields. If no record is found, a new record is created and the specified <Activate Field> is activated. If matching records are found, the first matching record is selected, and its data is updated with the scanned data. Additionally, if <Action to apply> has been specified, the appropriate intelligent updates occur. If no focus field is specified, users must click the field they want to update. |

Table 13. TriggerFindRecord Fields

| Field | Definition |
|-------------------|---|
| <Action to apply> | <p>Separated by a comma, this field defines the action that will be applied to each field, for example, as follows:</p> <p><Field 1>, <Action> ^ <Field 2>, <Action> . . . <Field N>, <Action></p> <p>The following actions can be applied to any field. Only one action can be specified for each field:</p> <ul style="list-style-type: none"> ■ Add—adds the scanned value to the field value. If the field specified is nonnumeric, an error is returned. ■ Subtract—subtracts the scanned value from the field value. If the field specified is nonnumeric, an error is returned. ■ Replace—sets the scanned value on the field. <p>If no action is specified, the default action is to <i>Replace</i> the value in the field.</p> <p>NOTE: <Action to apply> applies to the UpdateOnly and AddAndUpdate methods only.</p> |

Example 1: FindOnly

In the following example, Product Code and Lot# are specified as the key fields on which the search is based and Qty1 will be the activated field when the record is found:

TriggerFindRecord = Product Code, Lot# | Qty1 | FindOnly

Example 2: AddOnly

The following example specifies Product Code and Lot# as the key fields on which the search is based. If the record is found, then it is selected. If no record is found, then a new record is created with the scanned information, and Qty2 will be activated when the record is found or added:

TriggerFindRecord = Product Code, Lot# | Qty2 | AddOnly

Example 3: UpdateOnly

The following example specifies Product Code and Lot# as the key fields on which the search is based. UpdateOnly is specified as the method so the fields of any found records will be updated with the scanned information. The scanned Qty1 will be added to Qty1, the scanned Qty2 will be added to Qty2, and no field will be activated when the record has been updated:

TriggerFindRecord = Product Code, Lot# | | UpdateOnly | Qty1, "Add" ^ Qty2, "Add"

Example 4: AddAndUpdate

In the following example:

- SKU Code is specified as the key field on which the search is based.
- AddAndUpdate is specified as the method, so the fields of found records will be updated with the scanned information or a new record will be created with the scanned information.
- The scanned Qty1 will be added to Qty1.
- The scanned Qty2 will be subtracted from Qty2.
- The Expiry Date will be set.
- Qty1 will be the activated field when the record is found or added and updated.

```
TriggerFindRecord = SKU Code | Qty1 | AddAndUpdate | Qty1, "Add" ^ Qty2, "Subtract"  
^ Expiry Date, "Replace"
```

Example of Creating a New Barcode Definition in the Siebel Handheld Application

This topic gives one example of creating a new barcode definition in Siebel Service Handheld. Although this example is for the Siebel Service Handheld application, it pertains to all Siebel Handheld applications.

You can define new one-dimensional barcodes in the Barcode Administration views. To define a new barcode, you need to create a new barcode definition in the Barcode list, then create corresponding records in the Barcode Item list to define data elements within the barcode.

To create a new barcode definition, complete the following procedures. You can perform these procedures differently, depending on your business model.

- ["To create a new barcode definition for the Siebel Service Handheld application" on page 122](#)
- ["To enable the SHCE Service FS Activity Parts Movements views for barcode scanning" on page 123](#)

To create a new barcode definition for the Siebel Service Handheld application

- 1 Navigate to the Administration - Mobile screen, then the Barcode Administration view.
- 2 Click New, then fill in the necessary fields.

The following table describes the fields in the Barcodes list.

| Field | Sample Data |
|--------------|--|
| Name | Hard Disk 100 GB |
| Sequence | Set to the greatest number in the current list view plus one |
| Barcode Type | Location |
| Buscomp | FS Activity Parts Movement |

| Field | Sample Data |
|-------------|-------------|
| Min Length | 12 |
| Max Length | 12 |
| Prefix | # |
| Description | Service |

- 3 In the Barcode Item list, add a new record to identify how data will be parsed from the barcode. The following table describes the fields in the Barcode Item list.

| Name | Sequence | Min Length | Max Length | Data Desc |
|--------------|----------|------------|------------|--------------|
| UPC Code | 1 | 1 | 1 | |
| MFG Code | 2 | 5 | 5 | |
| Product Name | 3 | 5 | 5 | Product Name |
| CC | 4 | 1 | 1 | |

NOTE: Reducing the amount of barcode seed data extracted to the Siebel Service Handheld application improves client performance and reduces synchronization database extract size. The following steps provide an example of extracting only the barcode standards that are required for the Siebel Service Handheld application. This is applicable as long as all the barcode records for the Siebel Service Handheld application defined in the Barcode list have the Description attribute set to Service.

- a Navigate to the Administration - Mobile screen, then the Application Administration view.
- b Select Siebel Service for CE.
- c Click the Settings view tab, and make sure the parent record is the application Siebel Service for CE.
- d Locate the DefaultBusObj directive that has the following value:
 - ❑ SHCE Barcode|0||SHCE Barcode||SHCE Barcode Item||SHCE Barcode Enable|.
 Replace that value with the following value:
 - ❑ SHCE Barcode|0||SHCE Barcode|[Description]="Service"|SHCE Barcode Item||SHCE Barcode Enable|.

To enable the SHCE Service FS Activity Parts Movements views for barcode scanning

- 1 Navigate to the Administration - Mobile screen, then the Barcode Enabling view. The Barcode Enable list appears.

- If the following record does not exist, create a new record to enable the Product Name field in the SHCE Service FS Activity Part Movements List Applet for barcode scanning with the following information:

| View | Applet | Buscomp | Buscomp Field |
|---|---|----------------------------|---------------|
| SHCE Service FS Activity Part Movements | SHCE Service FS Activity Part Movements List Applet | FS Activity Parts Movement | Product Name |

- If the following record does not exist, create a new record for the Product Name field in the SHCE Service FS Activity Part Movements Form Applet for barcode scanning with the following information:

| View | Applet | Buscomp | Buscomp Field |
|---|---|----------------------------|---------------|
| SHCE Service FS Activity Part Movements (More Info) | SHCE Service FS Activity Part Movements Form Applet | FS Activity Parts Movement | Product Name |

- Save the changes. Now the Buscomp Field Product Name in both the SHCE Service FS Activity Part Movements view and the SHCE Service FS Activity Part Movements (More Info) view have been enabled for barcode scanning. The Buscomp Field Product Name specified here matches the value in the Data Desc column in the Barcode Item list defined earlier.

NOTE: The example provided above assumes that the product Hard Disk 100GB has been defined and properly set in the appropriate product catalog.

About BatchSync for Siebel Handheld Applications

BatchSync is a utility that enables Siebel Handheld customers to split the Siebel Handheld synchronization process into several steps, as opposed to requiring a complete synchronization (sync) session. When batch processing is enabled, you can select from one of four sync options after logging in:

- Full Sync.** Uploads transactions to the Siebel Server, processes transactions, extracts a configuration repository and database, downloads the repository and database to the handheld device, and logs off.
- Upload Only, Auto Download Later.** Uploads transactions to the Siebel Server and sets the handheld device to wait mode. After a specified period of time, the handheld device begins polling the Siebel Servers at a predefined time interval. If, during polling, the device detects a valid extract such as an un-expired dbfile.txt, the extract automatically downloads to the handheld device.
- Upload Only.** Uploads transactions to the Siebel Server and immediately logs off to enable transaction processing. If you need to create a new extract on the handheld device, the Upload Only session must be followed up by a Download Only session after the batchextract.exe completes on the Siebel Server.

- **Download Only.** Downloads a new database extract, followed by logoff. This option is only available if there is a valid or un-expired extract on the Siebel Server, and if there are no transactions remaining on the handheld device that requires uploading to the Siebel Server.

Process for Administering Batch Sync for Siebel Handheld Applications

The following procedures are included in this topic:

- [“Enabling BatchSync for Siebel Handheld Applications” on page 125](#)
- [“Downloading a Database Extraction” on page 126](#)
- [“Reviewing Batch Synchronization Status” on page 127](#)
- [“Running a Batch Extract” on page 129](#)
- [“Running a Batch Transaction Process” on page 129](#)

Enabling BatchSync for Siebel Handheld Applications

To split the synchronization process into several steps, you must enable the BatchSync utility for the Siebel Handheld application.

To enable Batch Synchronization

- 1 Install the Siebel Handheld application on the handheld device. No changes are required to the setup.ini or registry files.
- 2 Log on to the Siebel application from the Web. Ensure that the following are set up and running: Siebel Server, Handheld Synchronization components, and Batch Sync component.
- 3 Navigate to the Administration - Mobile, then the Application Administration view.
- 4 Select the application you are using, navigate to the Settings Tab, and set the following parameters to enable batch synchronization:
 - **EnableBatchExtraction: TRUE**
If set to TRUE, this parameter enables the batch synchronization functionality. Disable batch synchronization is the default
 - **ExtractionExpiration: 12**
This parameter specifies the number of hours that is used to determine when a database extraction has expired. When a database extraction expires, the batch synchronization re-extracts the databases. If no value is specified, then the databases are extracted each time they are requested.
- 5 Launch the Siebel Handheld application from the handheld device.

- 6 Select File, then the Synchronize to synchronize the Siebel Handheld application with the Siebel Server.

After the synchronization process completes, the new settings are downloaded to the handheld device.

- 7 Select File, then the Synchronize to synchronize the Siebel Handheld application with the Siebel Server again.

- 8 Enter your username and password, and select Start.

- 9 Verify that the four sync options are displayed:

- Full Sync
- Upload Now, Auto Download Later
- Upload Only
- Download Only

The option, Upload Only, is inactive if no new transaction records reside on the device. Download Only will be inactive because no extract exists for the user on the Siebel Server yet.

- 10 Select Full Sync, and select All to set all filters during the synchronization process.

- 11 Launch the Siebel Handheld application from the device, and add some transaction records to the handheld device.

- 12 Start the synchronization, select Upload Only, and select OK.

To check the status of the batch synchronization, see [“Reviewing Batch Synchronization Status” on page 127](#).

Downloading a Database Extraction

You can download a new database extraction from the Siebel Server.

To download a database extract

- 1 Navigate to the Administration – Server Management, then the Jobs view.
- 2 Add a new record and specify the following:
Component/Job: BatchSync
Mode: Asynchronous
- 3 Enter the requested server name.
- 4 In the Parameter applet, enter the following parameters:
 - Run Mode: Extract
 - Application Name: Siebel ePharma CE
 - Application Language code: ENU
- 5 In the Job Details applet, enter the time schedule parameters.
- 6 Select Start in the top parent applet.

- 7 Navigate to the Administration Mobile, then the Batch Sync Administration view.
- 8 When the batch synchronization process is complete, launch the Siebel Handheld application from the handheld device.
- 9 Select File, then the Synchronization, and select Download Only.

Table 14 lists the specific parameters for scheduling Siebel Handheld database extract jobs.

Table 14. Parameters for Scheduling Handheld Database Extract Jobs

| Parameter Name | Definition |
|----------------|--|
| RunMode | Run Mode. The following are valid values: <ul style="list-style-type: none"> ■ Extract—default value; extracts data for users who have uploaded their transactions to the Siebel Server where the component is running ■ ProcessTxn—processes pending transactions ■ ExtractUpgrade—extracts data for users who have uploaded their transactions to the Siebel Server during a major upgrade. Use only if an upgrade occurs between an upload and download of data ■ ProcessTxnUpgrade—processes pending transactions during the upgrade. Use only if an upgrade occurs between an upload and download of data |
| AppName | Application Name. An optional parameter to run batch synchronization for a specific application. Apply this parameter only in deployments where there are multiple Siebel Handheld applications. If no value is specified, or if the value is unclear, batch synchronization will be performed for all valid applications in the Siebel Handheld deployment. |
| AppLangCode | Application Language Code. This is a standard Siebel three-character language code. It is an optional parameter to run batch synchronization for a specific language application; it is only applicable in Siebel Handheld application deployments where there are multiple languages. If no value is specified or if the value is unclear, batch synchronization is performed for all valid languages of an application (if specified) or all applications (if not specified). |

Reviewing Batch Synchronization Status

The Batch Synchronization Status enables administrators to schedule and run batch synchronization as well as check the status of a batch synchronization for a specific application.

To review batch synchronization status

- 1 Navigate to the Administration - Mobile screen, then the BatchSync Administration view.

The Batch Synchronization Status displays a list of Handheld application users supported by batch synchronization. The fields in the Batch Sync Status list are described in the following table:

| Field | Comments |
|-------------------------------|--|
| User | User name of the Handheld application user supported by batch synchronization. |
| Sync Node ID | Internal identifier of a Handheld device or node. |
| Batch Transaction Proc Status | Last known status of the user's uploaded transactions with respect to batch transaction processing. The status values are as follows: <ul style="list-style-type: none"> ■ Completed ■ Failed ■ No Transaction ■ Scheduled ■ Scheduled by Admin ■ Scheduled by Sync ■ Unknown |
| Transaction Received | Timestamp of the last upload of transactions to be processed in batch. |
| Transaction Processed | Timestamp of the last processed transaction. |
| Batch Extract Status | Last known status of the user with respect to batch extract. The status values are as follows: <ul style="list-style-type: none"> ■ Downloaded ■ Extracted ■ Failed ■ Scheduled ■ No Extraction ■ Scheduled by Admin ■ Scheduled by Sync ■ Unknown |
| Extract Completed | Timestamp of the last completed batch extract. |

- 2 Click the Batch Extraction Session tab to display the timestamp and status of batch extract sessions for the specified user.

If you use the UploadOnly sync option, the records have a status of TxnRcvOk.

- 3 Click the Batch Transaction Processing Session tab to display the timestamp and status of batch transaction processing sessions for the specified user.

Once the records are processed, the status changes to TxnProcOk.

NOTE: TxnRcvOk and TxnProcOk status are BatchSync specific. If you are running full sync, you will not see these statuses. The transactions are generally processed when you logoff from an Upload Only Session. The administrator can start a task to process all pending transactions which would also change the status. In the case of a full sync, the transactions are processed as and when they are uploaded.

Running a Batch Extract

When specifying a batch extract, administrators can specify that they want to extract data for users who have uploaded their transactions to the server where the component is running.

To run a batch extract

- 1 Navigate to the Administration - Mobile screen, then the BatchSync Administration view.
- 2 Select the users for whom you want to run a batch extract.
- 3 Click Schedule Extract to change the batch extract status of the selected Handheld application user to Scheduled.

Details of the Handheld application user is extracted when the batch synchronization component is run with the appropriate run-mode. When the batch extract finishes, all users with a status of Batch Extract have the value of Scheduled and are extracted for download.

NOTE: The devices for the selected users are able to perform a download-only synchronization and download their extract.

- 4 Click Unschedule Extract to change the batch extract status of the selected user to Unscheduled.

Running a Batch Transaction Process

Administrators can run a batch transaction process for transactions that are uploaded to the Siebel Server by users.

To run a batch transaction process

- 1 Navigate to the Administration - Mobile screen, then the BatchSync Administration view.
- 2 Select the users for whom you want to run a batch transaction process.

- 3 Click Schedule Transaction-Processing to change the transaction process status of the selected user to Scheduled.

The user's uploaded transactions are processed when the batch synchronization component is run with the appropriate run-mode. When the batch transaction process finishes, all users with a status of Batch Transaction Process have the value of Scheduled and are processed.

- 4 Click Unschedule Transaction-Processing to change the transaction process status of the selected user to Unscheduled.

Load Balancing Support for Synchronization

Siebel Handheld application direct server synchronization can be load-balanced (at the application server level) to achieve better scalability and high availability. This capability removes the limitation of assigning users to a specific application server at installation time.

The application server uses a location on the server to store users' extracted files and state information. There is a unique synchronization directory for each user and device combination. The ApplicationDirectory user parameter enables the administrator to define this location. For information about setting the ApplicationDirectory user parameter, see ["Setting the ApplicationDirectory Parameter" on page 131](#).

By defining a shared location for this parameter, synchronization services for Handheld applications can be load-balanced. When a user requests a synchronization session, their request is directed (by a load-balancer) to an available Handheld application server hosting the appropriate Siebel Handheld Object Manager, which will be able to access the user's synchronization directory. For example, a Siebel Handheld application can be configured to point to a generic URL for synchronization.

The LoadBalanceLo parameter defines the location of the hostname (or IP address) port to be set for the load balancing server. This setting must be used in a load-balanced environment. When this value is specified, batch synchronization connects to the specified load balancer server instead of connecting to the application server directly.

The load balancers supported for synchronization are the same as those for the full Siebel Business application. You can use either the native load balancer or a third-party HTTP load balancers supported by Oracle. For more information about the supported load balancers, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

For information about configuring Siebel Server load balancing, see *Applications Administration Guide*. Load balancing distributes workload across multiple Siebel Servers. For background information on load balancing, see *Siebel Installation Guide* for the operating system you are using and *Siebel Deployment Planning Guide*.

Setting the ApplicationDirectory Parameter

ApplicationDirectory is an application-level parameter that is defined in the Administration - Mobile screen. The default value for this parameter in the standard product is hhsync\app_name, where app_name is the name of the application, such as sales or service, and so on. You can set this parameter using the following procedures.

To set the ApplicationDirectory parameter

- 1 Login to the Siebel Server using a user ID with a Siebel administrator responsibility.
- 2 Navigate to the Administration - Mobile screen, then the Application Administration view.
- 3 Select the appropriate application (for example, Siebel Sales for CE) from the Handheld Applications list.
- 4 In the bottom applet, choose the Settings view tab and set the value of the ApplicationDirectory parameter to be the shared network location using a UNC format, for example, as follows:
\\shareddsk\hhsync\sales.
- 5 Save the change.
- 6 Restart the Handheld Object Manager server component.

NOTE: The location for the ApplicationDirectory must be a networked, high performance, highly available drive. SAN or NAS with a high RAID rating is a good option for this purpose.

Enabling Encryption for Siebel Handheld Applications

This topic relates to the encryption and security of the handheld device. Siebel Handheld applications provide robust encryption of the Siebel password and database on the handheld CE devices. Users are authenticated at application startup time and certain components of the Handheld applications require intermittent access to the user's password.

To enable full encryption of your Siebel Handheld application, complete the following procedures:

- [“Enabling Application Lockout for a Siebel Handheld Application” on page 132](#)
- [“Enabling Authentication for a Siebel Handheld Application” on page 132](#)
- [“Enabling Local Database Encryption for a Siebel Handheld Application” on page 133](#)

Enabling Application Lockout for a Siebel Handheld Application

An application lockout can occur if a user fails to enter their correct password after a certain number of attempts. If authentication fails n times, where n is defined by the administrator by setting the FailedLoginAttempts setting, the current date is written to the security settings file, and a notification message that the application is exiting is displayed. The administrator can set the duration of the lockout by setting a value for the Lockout Period setting from the Administration - Mobile screen, Application Administration, then the Settings view tab.

If a user forgets their password and transactions exist in the device database, then the administrator can reset the user's password to blank using third-party device management software.

After the password is reset, the user can log in to the Siebel Handheld application using the blank password. However, users must still provide their password to synchronize. After the next successful synchronization login, the application password is automatically set to match the synchronization password.

To configure the Enable Application Lockout setting

- 1 Navigate to the Administration - Mobile screen, Application Administration, then the Settings view tab.
- 2 Select the Enable Application Lockout setting.
- 3 Set the value to TRUE.

Enabling Authentication for a Siebel Handheld Application

The authentication of a handheld device is necessary for security reasons. Any user who has access to a handheld device must not be able to use the Handheld application without the appropriate access rights. For this reason, users are required to enter a password at the splash screen, before the application loads. You are prompted for both a username and password when you want to perform synchronization. For more information about logging in to the Handheld application, see [“Logging in to the Siebel Pharma Handheld Application” on page 226](#).

The Allow Remember Password feature allows the user to save their password when they log into the Siebel Handheld application. If the user selects the Save Password check box on the log in splash screen, the password is saved to the device registry. When complete, users will not need to enter the password each time they log in to the Handheld application.

The check box on the splash screen is only enabled when the administrator sets the Allow Remember Password property to true.

To configure the Allow Remember Password setting

- 1 Navigate to the Administration - Mobile screen, Application Administration, then the Settings view tab.

- 2 Select the Allow Remember Password setting.
- 3 Set the value to TRUE.

Enabling Local Database Encryption for a Siebel Handheld Application

Siebel Handheld database encryption is required to provide extra security for users data on the handheld device. If the Enable Encryption setting is enable, the database is always encrypted so that no other application can open and view the data. Once the application is loaded, the database is decrypted as expected. No other application can try to access the database because it is locked by the Handheld application. The database is always encrypted when not in use by the Handheld application.

To enable encryption on a local database

- 1 Navigate to the Administration - Mobile screen, Application Administration, then the Settings view tab.
- 2 Select the Enable Encryption setting.
- 3 Set the value to True.

When this setting is enabled, the database is encrypted when the application exits, and is decrypted when the application is started.

Storing Asynchronous Data

An asynchronous query enables users to search information from the Siebel Server through SFM if the information is not available from the Siebel Handheld application local database. Users can integrate query results from an asynchronous query in the Siebel Handheld application database.

Users can perform an asynchronous query for the business objects (for example, Solution, Product, Asset and Orders) defined for the application. For example, if a mobile field technician at a customer site wants to check the status of an order and the order is not available on the local database, the technician can send an asynchronous query to the Siebel Server through SFM by specifying the account name and order date for the order. Shortly afterwards, the Asynchronous Query icon flashes, which means that the query results are returned from the Siebel Server.

To store asynchronous data

- 1 Tap the icon to open the Asynchronous Query Result Alert window.
The alert window lists all the asynchronous queries performed by the user.
- 2 Highlight the query you want and tap the Details button.
The Asynchronous Query Result List view lists the data that meets the query criteria.

- 3 Select the record that you want and tap the Store button to store the record into the Siebel Handheld application database for further reference.

NOTE: When the record is stored on your Siebel Handheld application, you can review, search, or update the record from the Handheld application screens and views.

Receiving Incoming Messages from the Server

From the incoming message notification window, users can drill down on the hyperlink under the dispatched activity section to navigate directly to the activity detail view. Messages or activities are dispatched through SFM. You can handle incoming alert messages differently, depending on your business model.

You can tap the message alert icon to open the incoming message alert window. The alert window displays all the incoming messages sent from the Siebel Server by SFM. The top row shows the most recent message. You can close the alert window or can tap the hyperlinked activity # or message to review the detail information of the activity or message.

Removing Reassigned Activities

You can remove reassigned activities differently, depending on your business mode. Users can delete an activity from a Siebel Handheld application database through SFM if the activity has been assigned to another user. If an activity has been dispatched to user A on a Handheld application and then the activity is assigned to another mobile user, the activity on user A's Handheld application must be deleted.

For example, Technician A is assigned activity 12-98Y98 via SFM and the activity 12-98Y98 is stored in the Handheld application database. A call center agent, however, then decides to re-assign the activity (12-98Y98) to another technician, Technician B. The call center agent goes to the Dispatch Board, drags and drop the activity, and assigns the activity to Technician B. Because the activity has been reassigned to Technician B, the Siebel Server sends an SFM message to Technician A's Handheld application. This message arrives on Technician A's handheld device, and removes the activity from their Handheld application database. Technician A can see an alert message in the application's incoming message alert window, which indicates that the activity has been assigned to another technician and removed from their Handheld application database.

About Monitoring Synchronization Sessions for Siebel Handheld Applications

You can monitor synchronization sessions and synchronization conflicts for Siebel Handheld applications from a number of screens within the Siebel Mobile Web Client. If you are using Direct Server Synchronization (DSS) or Direct Server Synchronization via Proxy (DSSvP), you do so by way of the Siebel Web Client. Every time a user synchronizes with the Siebel Server, a record is made.

The following procedures relate to monitoring synchronization:

- ["Monitoring Synchronization Sessions for Siebel Handheld Applications" on page 135](#)

- [“Obtaining Error Details from Synchronization Sessions in Siebel Handheld Applications” on page 136](#)
- [“Obtaining Business Component Information from Synchronization Conflicts in Siebel Handheld Applications” on page 136](#)
- [“Obtaining an Audit Trail for Synchronization Sessions in Siebel Handheld Applications” on page 137](#)
- [“Responding to Synchronization Errors in Siebel Handheld Applications” on page 137](#)

Monitoring Synchronization Sessions for Siebel Handheld Applications

The Session Administration view allows you to gather all data related to each synchronization session a user performs. In the Session Administration view, the Sync Status list appears above two child lists—Sync Sessions and Extraction Info.

- **Sync Status** Allows you to monitor each user synchronization session, and includes information on synchronization status, date, type of synchronization, and so on.
- **Sync Sessions** Allows you to gather further information on the status of each synchronization session, including information on any conflicts that occurred during the synchronization, the size of the data extracted, and so on.
- **Extraction Info** Allows you to review a detailed description of all data extracted during the synchronization session, including business component information and the number of records extracted.

To monitor synchronization sessions

- Navigate to the Administration - Mobile screen, then the Session Administration view.

The Sync Status list appears with the two child lists, Sync Sessions and Extraction Info, below it.

To monitor data download

- 1 Navigate to the Administration - Mobile screen, then the Session Administration view.

The Sync Status list appears with the two child lists, Sync Sessions and Extraction Info, below it.

- 2 Scroll down to the Extraction Info list.

There is one extraction record for each unique extraction context. Review the data extraction information recorded. Information recorded for review is described in the following table.

| Field | Comments |
|---------------------|---|
| Business Component | The business component on which the synchronization is based. |
| Context | The context of the extraction. |
| # Records Extracted | The number of records that exist in the context of applied filters. |
| Extract Time | The length of time it took to extract the set of records. |

| Field | Comments |
|---------------------------|---|
| Business Object Filter | The business object filter (if any) applied to the extraction. |
| Business Component Filter | The business component filter (if any) applied to the extraction. |

You should use this information to help answer the following questions:

- Why did I download a particular record to the device?
- Why is an expected server record not downloaded to the device?
- Why is the total number of records showing up in some views so big?
- Why is the total number of records showing up in some views so small?
- Why does the extraction take so long?

The extraction record lists the filters used to control the extraction, the context of the extraction, how many records exist in that context for those filters, and how long it took to extract that set of records. By analyzing these records, and looking at all extraction changes resulting from filter changes, and, or changes to search specifications (defined in Siebel Tools), you can answer all of the above questions.

NOTE: Because storage space on a handheld device is typically limited, and available networking bandwidth for synchronization might not always be optimal, it is critical to carefully configure the filters and keep the size of your local database small. It is recommended that users should have a local database size of around three megabytes (MB), and this should never exceed 10 MB.

Obtaining Error Details from Synchronization Sessions in Siebel Handheld Applications

The Conflict Administration view allows you to obtain information on synchronization errors that occur during individual synchronization sessions.

To obtain synchronization error details

- 1 Navigate to the Administration - Mobile screen, then the Conflict Administration view.
- 2 In the Transactions list, select the appropriate record, and then click the Error Details view tab.
- 3 In the Error Details list, select or query for the desired record.

Obtaining Business Component Information from Synchronization Conflicts in Siebel Handheld Applications

The Transactions view on the Conflict Administration screen allows you to obtain business component information related to synchronization errors that occur during individual synchronization sessions.

To obtain business component error details

- 1 Navigate to the Administration - Mobile screen, then the Conflict Administration view.
The Transactions list appears.
- 2 In the Transactions list, select the appropriate record, and then click the Business Components view tab.

Obtaining an Audit Trail for Synchronization Sessions in Siebel Handheld Applications

The Transactions view on the Conflict Administration screen allows you to obtain audit trail information for an action that was taken against a particular transaction. You must enable Audit Trail separately. For information about enabling Audit Trail, see *Applications Administration Guide*.

To obtain audit trail information for synchronization error details

- 1 Navigate to the Administration - Mobile screen, then the Conflict Administration view.
The Transactions list appears.
- 2 In the Transactions list, select the appropriate record, and then click the Audit Trail view tab.

Responding to Synchronization Errors in Siebel Handheld Applications

When reviewing a synchronization error, you can respond to it in one of three ways—you can ignore it, export it, or reapply it.

To respond to synchronization transaction errors

- 1 Navigate to the Administration - Mobile screen, then the Conflict Administration view.

The Transactions list appears. The transaction buttons are described in the following table:

| Action | Result |
|---------|--|
| Reapply | <p>Reapplies the selected transaction with values listed in the view (or Field Value view). These values override the results provided by the Siebel Server.</p> <p>TranReapplyComponent and TranReapplyServer are necessary when using Companion Synchronization and ignored for Direct Server Synchronization. These are used when selecting transactions to reapply from the Administration - Conflict screen.</p> <p>If TranReapplyComponent and TranReapplyServer are not specified, it is impossible to reapply transactions originating from Companion Synchronization.</p> |
| Export | <p>Exports the content of the transaction in the same format used by the synchronization engine. For help with interpreting the content, create a service request (SR) on <i>My Oracle Support</i>. 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 are listed on <i>My Oracle Support</i>.</p> |
| Ignore | <p>Accepts the current status and does not act upon the error. The record becomes read-only. There is no impact, except that the error will no longer appear in queries for unresolved conflicts.</p> |

- 2 Click either Reapply, Export, or Ignore depending on your needs.

NOTE: This works only when the Object Manager is running on a Windows-based server.

Organization Enabled LOV Support

Siebel Handheld applications by default support a fixed set of LOV values for given LOV types across all organizations. If you want your Handheld application to support multiple organization enabled LOVs, then you must enable organization LOV support in the Siebel Handheld Client.

- ["Enabling Organization LOV Support in the Siebel Handheld Client" on page 139](#)
- ["Example of Configuring, Enabling, and Using Multiple Organization LOVs in Handheld Applications" on page 139](#)

NOTE: Before enabling organization LOV support in the Siebel Handheld Client, make sure that you have satisfactorily associated your organizations with LOV types and LOV values in the Siebel Web Client. For more information about LOVs, see *Applications Administration Guide*. For more information about setting up organization enabled lists of values, see *Configuring Siebel Business Applications* and also the example that follows.

Enabling Organization LOV Support in the Siebel Handheld Client

There are two main stages involved in enabling organization LOV support in the Siebel Handheld Client.

- First you must create a new business object in Siebel Tools (for example, SHCE LOV Internal) and attach *List Of Values (Internal) buscomp* to it. For more information about creating business objects in Siebel Tools, see *Using Siebel Tools*.
- Then you must force extraction of the *List Of Values (Internal) buscomp* with a specific filter expression. This involves the following:
 - Creating a new DefaultBusObj directive in the Siebel Web Client and assigning a value to it. For more information about this task, see [“To create a new default directive” on page 139](#).
 - Adding a business object filter setting to your Siebel Web Client, which will run on DefaultBusObj. For more information about this task, see [“To add a business object filter setting” on page 102](#).

To create a new default directive

- 1 Navigate to the Administration - Mobile screen, then the Application Administration view.
- 2 In the Handheld Applications list, select the appropriate application, and then click the Settings tab.
- 3 Click New, and then enter the required information.

| Field | Comments |
|-------|---|
| Name | The name of the directive. For example, DefaultBusObj. |
| Value | The value to set in the directive. This value should correspond to the business object from which you want to extract List Of Values (Internal) buscomp. For example, SHCE LOV Internal 0 List Of Values (Internal) [Intersection Table Org Id]. |

Example of Configuring, Enabling, and Using Multiple Organization LOVs in Handheld Applications

This topic gives an example of configuring, enabling, and using multiple organization LOVs in the Siebel Service Handheld application. Although this example is for the Siebel Service Handheld application, it pertains to all Siebel Handheld applications. You can perform these procedures differently, depending on your business model.

- [“To configure organization LOVs in the Web client for Siebel Service Handheld” on page 140](#)
- [“To enable organization LOV support for Siebel Service Handheld” on page 140](#)

- "To use organization enabled LOVs in Siebel Service Handheld" on page 141

To configure organization LOVs in the Web client for Siebel Service Handheld

- 1 Create the new LOV type CONTACT_TYPE LOV in Siebel Tools. Enter new records for its LOV values and configure a picklist to display the LOV type.

For information about creating new LOV types and LOV values using Siebel Tools, see *Configuring Siebel Business Applications*.

- 2 Set up CONTACT_TYPE LOV to be organized enabled. Do this by associating the following LOV values to CONTACT_TYPE LOV for the following organizations.

| LOV Type | LOV Value |
|---|---------------------|
| For Xerco Pharmaceutical Organization: | |
| CONTACT_TYPE LOV | Physician |
| CONTACT_TYPE LOV | Nurse Practitioner |
| CONTACT_TYPE LOV | Pharmacy Directory |
| For Medical Products Organization: | |
| CONTACT_TYPE LOV | Cardiologist |
| CONTACT_TYPE LOV | Purchasing Director |
| CONTACT_TYPE LOV | Nurse Practitioner |

For information about associating organizations to LOV values and LOV types, including guidelines for setting up organization enabled lists of values, see *Configuring Siebel Business Applications*.

To enable organization LOV support for Siebel Service Handheld

- 1 Create a new business object (SHCE LOV Internal) in Siebel Tools and attach *List Of Values (Internal) buscomp* to it. For more information about enabling organization LOV support, see *Using Siebel Tools*.
- 2 Create a new default directive for Siebel Service Handheld as follows:
 - a Navigate to the Administration - Mobile screen, then the Application Administration view.
 - b In the Handheld Applications list, select Siebel Service for CE, and then click the Settings tab.

- c Click New, and then enter the required information.

| Field | Sample Data |
|---|---|
| Name | DefaultBusObj |
| Value | SHCE LOV Internal 0 List Of Values (Internal) [Intersection Table Org Id] |
| <p>NOTE: This value should correspond to the business object from which you want to extract List Of Values (Internal) buscomp.</p> | |

- 3 Add a business object filter setting for Siebel Service Handheld, which will run on DefaultBusObj, as follows:
 - a Navigate to the Administration - Mobile screen, then the Application Administration view.
 - b In the Handheld Applications list, select Siebel Service for CE, and then click the Settings tab.
 - c Click the Business Object Filters view tab.
 - d Click New, and then enter the required information.

| Field | Comments |
|-----------------|--|
| Owner | Siebel Service for CE |
| Default | Leave unchecked |
| Business Object | Account |
| Name | OrgLOVFilter1 |
| Query | [Intersection Table Org Id] is NOT NULL OR [OrgSpecializedLOV.Internal_Organization] is NOT NULL |

To use organization enabled LOVs in Siebel Service Handheld

- 1 Open your Siebel Service Handheld application in your handheld device, then synchronize with the Web client. Make a note of your active position. For example, are you associated with the Xerco Pharmaceutical Organization, the Medical Products Organization, or some other organization.

For information about how to synchronize your handheld device with your Siebel Handheld Application Server, see the online help for the Siebel Handheld application.

NOTE: If you do not know what your active position is, contact your system administrator.

- 2 Navigate to My Contacts on your handheld device and add a new contact record.

When you specify the contact type, the LOVs associated with your current organization (the organization to which your active position is associated) are displayed in the Type field.

For information about how to add records, see the online help for the Siebel Handheld application.

For example, if your active position is associated with Xerco Pharma Organization, then the Type field includes the following LOV options: Physician, Nurse Practitioner, and Pharmacy Director. But if your active position is associated with Medical Products Organization, then the Type field includes the following LOV options: Cardiologist, Purchasing Director, and Nurse Practitioner.

NOTE: A user can be associated with multiple organizations if they hold multiple positions, but only the organization of the active position determines which LOVs are displayed.

- 3 Navigate to My Contacts on your handheld device and open an existing contact record by tapping it. Notice that when you view the contact type, the LOVs associated with the record's owner organization are displayed.

For example, if your active position is associated with Xerco Pharma Organization, but the owner of the record you are viewing is associated with Medical Product Organization, then the LOVs displayed for the picklist are those associated with Medical Product Organization (Cardiologist, Purchasing Director, and Nurse Practitioner).

- 4 Synchronize the handheld device again to copy any changes made using your handheld device with the Siebel Web Client database.

For more information about synchronization, see [“Synchronizing Data on Your Handheld Device” on page 214](#).

Process of Administering Siebel Pharma Handheld Applications

The following topics cover only those administration procedures that apply to Siebel Pharma Handheld applications. For detailed information about managing samples, see *Siebel Life Sciences Guide*.

The following procedures relate to the process of administering your Siebel Pharma Handheld application:

- [“Managing Sample Records in Siebel Pharma Handheld Applications” on page 143](#)
- [“Administering Disclaimer Text in Siebel Pharma Handheld Applications” on page 144](#)
- [“Configuring Distributor Names in Siebel Pharma Handheld Applications” on page 144](#)
- [“Performing a Signature Audit in Siebel Pharma Handheld Applications” on page 145](#)
- [“Verifying Signature Capture in Siebel Pharma Handheld Applications” on page 146](#)
- [“Configuring User Properties for Closing Inventory Periods in Siebel Pharma Handheld Applications” on page 148](#)

- [“Configuring User Properties for Signature Capture Verification in Siebel Pharma Handheld Applications” on page 149](#)
- [“Validation Logic of the Sign and Submit Buttons in Siebel Pharma Handheld Applications” on page 150](#)
- [“Validating the DEA Number in Siebel Pharma Handheld Applications” on page 157](#)
- [“Preparing Receipts for Calls in Siebel Pharma Handheld Applications” on page 157](#)

Managing Sample Records in Siebel Pharma Handheld Applications

Administrators can manage sample records using either Siebel Pharma Handheld or Siebel Life Sciences, but they must not use both. In order to keep sample records synchronized, users must be prevented from submitting a samples transaction on the when the same record is still editable on their handheld device.

When a user submits a record, the status changes to Submitted and it becomes read-only, meaning no additional operations are allowed on the record. By updating a user property and writing two small Visual Basic scripts, you can specify that synchronized records are also read-only.

- First, the administrator changes the value of the user property Update Status To Synchronized to Y.

Making this change invokes the UpdateStatusToSynchronized method on the Sample Txn business component.

- Then, the administrator writes a small Visual Basic script for this method to change the status on the Sample Txn to Synchronized.

- Finally, the administrator writes another Visual Basic script to specify that the record becomes read-only when its status changes to Synchronized.

This prevents users from accidentally submitting the sample transactions on the server while the same record is still editable on their handheld device.

Sample transactions are not logged upon submitting a sample drop. They only appear on the handheld device after synchronization.

NOTE: The Disburse on Signature Capture system parameter is set to TRUE by default. This means that when the user clicks the Save button on the Signature Capture view, the application decrements the inventory count for samples dropped only. For promotional items dropped, the transaction records are generated when submitting the call.

Administering Disclaimer Text in Siebel Pharma Handheld Applications

Your legal department might require that a change be made to the disclaimer text that appears below the signature area on the Siebel Signature Capture display of Siebel Pharma Handheld applications. Only one disclaimer can be active at a time.

NOTE: Signature capture configurability is only available on Handheld applications, not on Siebel Web Client applications.

To edit the disclaimer text

1 Navigate to the Administration - Samples screen > Disclaimer Administration view.

2 In the Disclaimers list, add a record.

3 Enter the new text in the Disclaimer field.

The Status, Start Date, End Date, Created By, and Created fields are read-only.

4 Click Submit.

The submitted disclaimer becomes active, and the status and date fields are updated. When the sales representatives next synchronize, the disclaimer on their handheld device is updated.

Configuring Distributor Names in Siebel Pharma Handheld Applications

The name of the distribution company appears by default on the Signature Capture display. You might need to change this name if, for example, there is a name change as a result of a merger between two companies.

In addition, a company can use a contract sales force to distribute some of its products. You can use the Distributor by Employee feature to customize the display to show the name of the contract distributor for the contract sales representatives and to show the name of the primary pharmaceutical company for its sales representatives.

To change the distributor name in the Siebel Signature Capture display

1 Navigate to the Administration - Application screen > System Preferences view.

2 Find the Company Name system preference and change the value to the new company name.

When the sales representatives next synchronize their PDAs, the new company name is downloaded to their Siebel Pharma Handheld application.

3 Navigate to the Administration - User screen > Employees view.

4 Select the employee from the Employees list.

- 5 In the More Info view, enter the account of the distributor in the External Account field. (The external account must already have been set up as an account.)

If the External Account field is blank for an employee, the company name set in the System Preference Administration view, set in [Step 2](#), is used.

- 6 Repeat [Step 4](#) and [Step 5](#) for each employee of the distributor.

Performing a Signature Audit in Siebel Pharma Handheld Applications

The Siebel administrator is typically assigned responsibility for checking the validity of signatures recorded electronically. The administrator uses the Verify button in the Signature Audit Administration view. For more information about the logic behind the Verify button, see [“Validation Logic of the Sign and Submit Buttons in Siebel Pharma Handheld Applications” on page 150](#).

If you are the samples administrator, you must check signature audit administration regularly, and perform signature capture verification to make sure that the signatures collected from each contact are consistent.

Users might see an error message if signature audit administration tasks are performed on a desktop computer running multiple versions of the Siebel Pharma application. Make sure that only the latest version of Siebel Pharma is running, and uninstall all previous versions.

To verify signatures

- 1 Navigate to the Administration - Samples screen > Signature Audit Administration view.
- 2 In the Sample Signatures list, create a query.
- 3 Visually compare the signatures for consistency.

Contact names are retained even if the Contact record is deleted. If a Contact record is deleted from the Administration - User screen > Persons view, the contact name is removed from all applets with reference to that contact's name in the S_CONTACT table. Therefore, if the Contact ID is deleted from the S_ACT_SIGN table, the last name and first name fields appear empty in the Sample Signature form in the Signature Audit Administration view.

However, the Contact Full name is not affected by deleting a contact record because it is stored in a text field in the same table. In the Signatures applet, the control above the signature displays the full name of the contact at the time of the call (that is, no reference to the S_CONTACT table). Because the contact's name is not removed from this applet, a permanent and complete electronic record of calls captured with signatures recorded electronically is maintained.

Verifying Signature Capture in Siebel Pharma Handheld Applications

Signature capture verification compares information captured at the time of signature (the signature event string) with the current call information (the current call string) to assess if the call information has been modified. Reasons that the call information might be modified include:

- Recreation of a sample receipt
- Sample adjustments
- Fraud

Signature capture verification is launched from the Verify button on the Signature Audit Administration view. For an example of signature audit administration, see [“Performing a Signature Audit in Siebel Pharma Handheld Applications” on page 145](#).

Signature Key

When the signature is first created, a signature key is calculated from the signature and stored. When the signature is verified, this key is recalculated from the current signature and compared to the stored key. If the two keys do not match, the signature key checksum fails. A failed checksum indicates that the current signature has been altered since the time of capture. If the checksum fails, the signature is not displayed in the Signature Audit Administration view.

Outcomes of Checksums

The signature capture verification status (which can be Passed, Adjusted, or Failed) is determined by the outcome of checksums. Checksums is a user property which compares text saved at the time of the signature capture with fields in the current call record and the signature key checksum. See [Table 15 on page 146](#) for examples of checksum combinations.

NOTE: You might need to expose the Verification Status field using Siebel Tools. This is a configuration change that might need to be performed before implementing Siebel Pharma Handheld. For more information about exposing the Verification Status field, see the release notes for Siebel Pharma Handheld.

Each user property checksum returns results for a different group of fields involved in the signature capture. The checksum examples and verification status settings are shown in [Table 15](#).

Table 15. Checksum Examples

| If... | Then the verification status is set to... |
|--|---|
| All checksums pass | Passed |
| Any of the checksums fail | Failed |
| One or two checksums return Adjusted and the others pass | Adjusted |

User Properties for Signature Capture Verification

In the default installation of Siebel Pharma Handheld, there are three user properties that set fields for the checksums (Checksum Field: 1, Checksum Field: 2, and Checksum Field: 3). These user properties and their default configurations are described in the following topics.

Using Siebel Tools, you can create additional checksums. For more information about checksums, see [“Creating New Checksum Field User Properties” on page 150](#).

Checksum Field: 1—Signature Header Text

This checksum compares the signature header text recorded at time of signature with the similar string in the current call record.

The three possible outcomes (Passed, Adjusted, and Failed) for this checksum depend upon the signature header text strings and the (signature) Status field, as shown in [Table 16](#).

Table 16. Checksum Status for Signature Header Text

| The Checksum Status Is... | If the Signature Event String... |
|---------------------------|---|
| Passed | Equals the current call string. |
| Adjusted | Is different from the current call string and the (signature) Status is Inactive. |
| Failed | Is different from the current call string and the (signature) Status is Active. |

Checksum Field: 2—Signature Body Text

This checksum compares the signature body text recorded at the time of signature with the similar string in the current call information.

The outcome (Passed and Adjusted) for this checksum depends on whether the signature body text strings match, as shown in [Table 17](#).

Table 17. Checksum Status for Signature Body Text

| The Checksum Status Is... | If the Signature Event String... |
|---------------------------|--|
| Passed | Equals the current call string. |
| Adjusted | Is different from the current call string. |

Checksum Field: 3—Sales Representative’s Row Number

This checksum compares the identity of the call creator at time of signature with the identity of the current creator of the call. The employee’s row number is used to identify the call creator.

An employee’s row number can be obtained by selecting the employee in the Employees view and then selecting Help > About Record. The Employee’s Row # appears in the About Record dialog box.

The outcome (Passed and Failed) for this checksum depends on whether the sales representative name text strings match, as shown in [Table 18](#).

Table 18. Checksum Status for Sales Representative's Row

| The Checksum Status Is... | If the Signature Event String... |
|---------------------------|--|
| Passed | Equals the current call string. |
| Failed | Is different from the current call string. |

Configuring User Properties for Closing Inventory Periods in Siebel Pharma Handheld Applications

The recommended process for closing an inventory period is:

- 1 Synchronize the handheld device with Siebel Server immediately before closing the period.
- 2 Close the inventory period.
- 3 Synchronize the handheld device with Siebel Server a second time, immediately after closing the inventory period.

Users must synchronize the Siebel Handheld application with the Siebel Server before closing the inventory period because there can be calls on the Handheld application that have not yet been synchronized with the Siebel Server. Only the current active period is downloaded to the handheld device. Therefore, after the current inventory period is closed, they must synchronize a second time in order to bring down the next active period.

To enforce this procedure, the inventory period closure logic has been added. When it is enabled, this logic goes into effect during the inventory period reconciliation. It ensures that the user synchronizes their Handheld application with the Siebel Server before closing the inventory period, and it requires the user to submit all outstanding calls (that is, when an electronic signature has been captured, but the call has not been submitted).

The following user properties have been created on the Pharma Inventory Period business component to support the inventory period closure logic:

- **Inventory Closure Validation.** The default value is N (No), and the Inventory Period Closure logic is not triggered. To activate the Inventory Period Closure logic, set this parameter to Y (Yes). Setting the parameter to Y forces the user to synchronize the Handheld application with Siebel Server prior to closing the inventory period.
- **Acceptable Period to Complete Synchronization.** The default value is 30 (minutes). When Inventory Closure Validation is set to Y, the Acceptable Period to Complete Synchronization user property determines a grace period during which the user must synchronize the Handheld application before closing the inventory period. If the user does not synchronize within the time period, they will not be permitted to close the inventory period.

If Inventory Closure Validation is enabled and the Acceptable Period to Complete Synchronization user property is not created, the default value of 30 minutes is used.

After a user successfully closes the inventory period, a message prompts the user to synchronize the handheld device with the Siebel Server. However, note that this step is not enforced. If the user fails to synchronize a second time, the handheld device will not receive the current period, and this can potentially cause synchronization errors the next time the user synchronizes.

NOTE: The same logic is enforced during inventory reconciliation.

Configuring User Properties for Signature Capture Verification in Siebel Pharma Handheld Applications

Use Siebel Tools to edit the following user properties:

- Turn checksums on or off
- Add additional fields to checksums
- Remove fields from checksums
- Create new checksums of the same form as Checksum Field: 3

Editing Checksum Field: 1—Signature Header Text

This checksum compares the Signature Header Text field in the Pharma Call Signature business component with the concatenated text string of the Header Text Fields (Header Text Field: 1 to Header Text Field: 7) in the Pharma Professional Call business component.

In Siebel Tools, you can edit these user properties by adding, deleting, or editing the Header Text Fields in the Pharma Professional Call business component. The maximum allowed number of Header Text Fields is 20 (Header Text Field: 20).

Editing Checksum Field: 2—Signature Body Text

This checksum compares the signature body text in the Pharma Call Signature business component with the concatenated text string of the Signature Body Text Fields (Signature Body Text Field: 1 to Signature Body Text Field: 3) in the Pharma Call Sample Dropped business component.

In Siebel Tools, you can edit these user properties by adding, deleting, or editing the Signature Body Text Fields in the Pharma Call Sample Dropped Call business component. The maximum allowed number of Signature Body Text Fields is 10 (Signature Body Text Field: 10).

Editing Checksum Field: 3—Sales Representative's Row Number

This checksum compares the "Created by" field of the Pharma Call Signature business component with the "Created by" field of the Pharma Professional Call business component.

In Siebel Tools, you can edit these user properties by replacing "Created by" with any field found in both business components.

Creating New Checksum Field User Properties

In Siebel Tools, you can create user properties for up to 25 checksums that compare fields in the Pharma Call Signature business component with fields in the Pharma Professional Call business component.

To create a new checksum field user property

- In Siebel Tools, create a new user property called Checksum Field: n , where n is less than or equal to 25. For more information on creating checksum field properties, see *Using Siebel Tools*.

Use Checksum Field: 3 as a template to create new user properties with the same format.

Table 19. Checksum Field User Properties

| User Property Name | Value |
|---------------------|---|
| Checksum Field: 3 | "Created by", "Created by" |
| Checksum Field: n | "field in Pharma Call Signature", "field in Pharma Professional Call" |

Validation Logic of the Sign and Submit Buttons in Siebel Pharma Handheld Applications

You can edit the validation logic for the Signature and Submit buttons for professional and account calls.

Default Validation Logic Flowcharts

The default validation logic for the Signature and Submit buttons is shown in the following figures: [Figure 4 on page 151](#), [Figure 5 on page 152](#), and [Figure 6 on page 153](#). Not all implementations of Siebel Pharma Handheld require the extensive validation as shown in these figures. Before most validation steps, the procedure consults a user property to determine if the field must be used during the validation. For example, if it is not necessary to validate the License Number field, the user property Validate License Number is set to false. The Signature button validation then skips the validation question: Is the License Number field populated?

User Properties for altering validation logic are described in ["About Changing Validation Logic by Editing User Properties" on page 153](#).

Figure 4 shows the default validation logic for the Signature button.

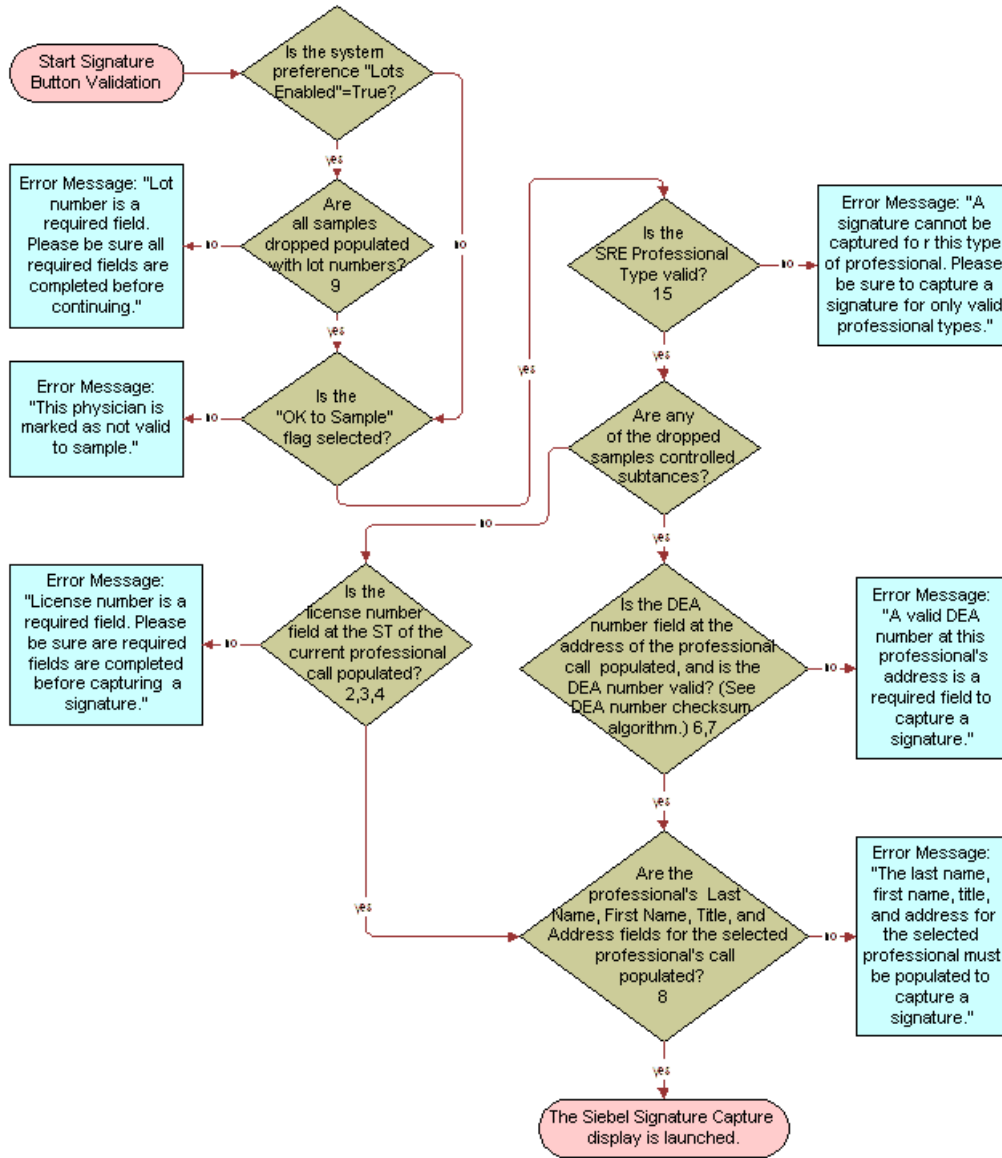


Figure 4. Default Validation Logic Flowchart for Signature Button

NOTE: The numbers in Figure 4 correspond to the numbers of the user properties described in Table 20 on page 154.

Figure 5 shows the default validation logic for the Professional Call Submit button.

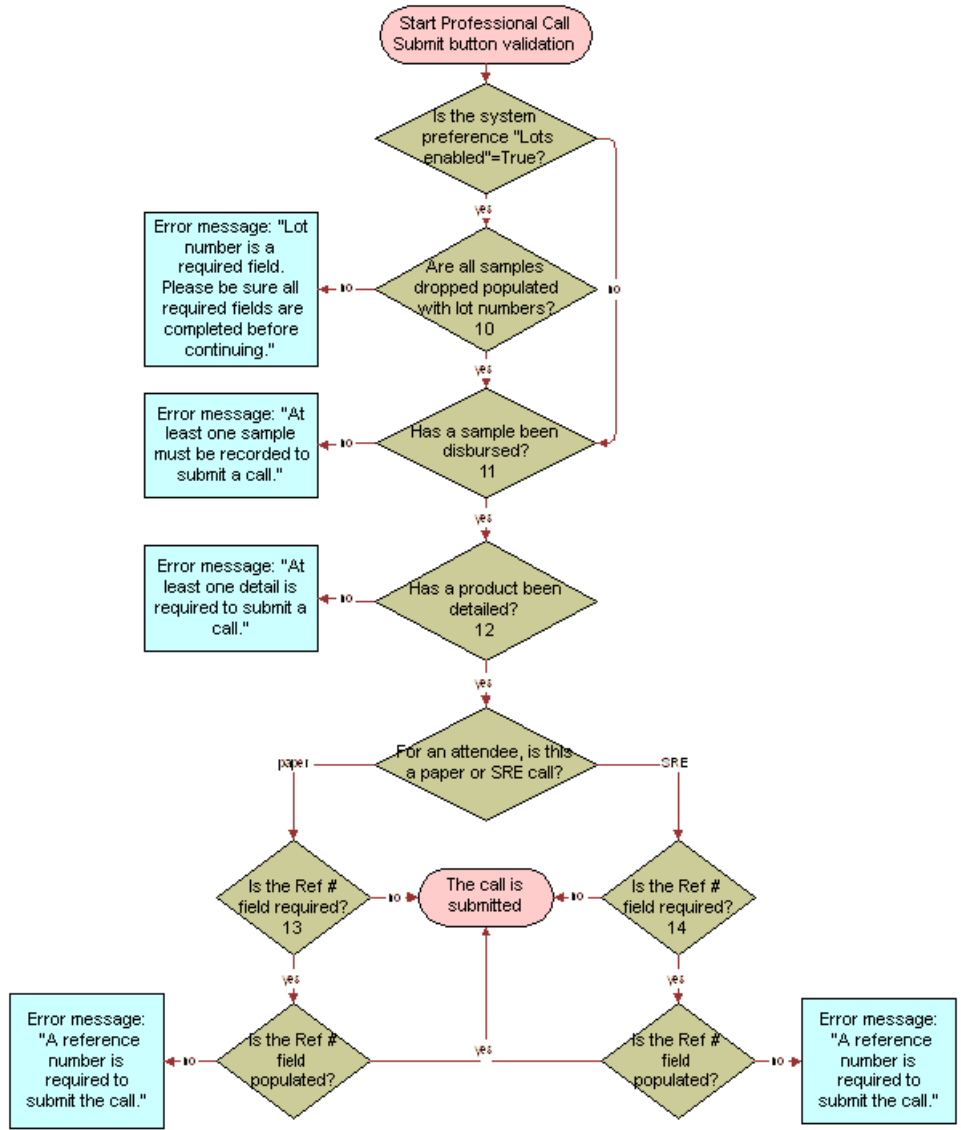


Figure 5. Default Validation Logic Flowchart for Professional Call Submit Button

NOTE: The numbers in Figure 5 correspond to the numbers of the user properties described in Table 20 on page 154.

Figure 6 shows the default validation logic for the Account Call Submit button.

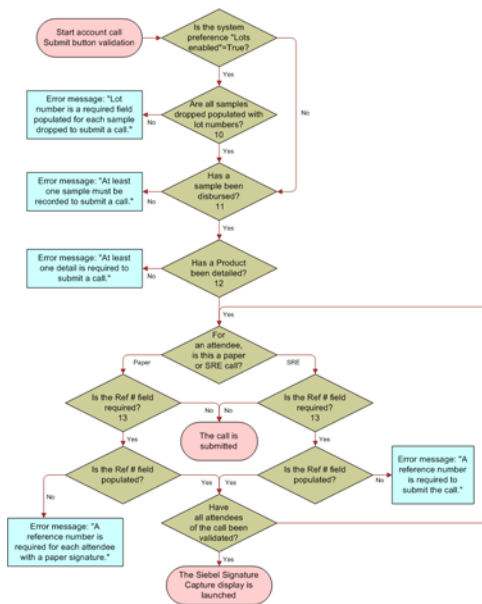


Figure 6. Default Validation Logic Flowchart for Account Call Submit Button

NOTE: The numbers in Figure 6 correspond to the numbers of the user properties described in Table 20 on page 154.

About Changing Validation Logic by Editing User Properties

Validation logic uses user properties to determine whether a field must be used during validation.

Table 20 explains the user properties that you can use to alter the validation logic of the Sign, Professional Call Submit, and Account Call Submit buttons.

Table 20. User Properties for Setting Sign and Submit Button Logic

| Flowchart Reference | User Property | Action if Y... | Action if N... | Default Value |
|---------------------|---|--|---|---------------|
| 1 | <p>Validate OK to Sample</p> <p>Use to enforce confirmation that the OK to Sample flag is selected before launching the Signature Capture display.</p> | Enforce the OK to Sample validation. | Skip the OK to Sample validation. | Y |
| 2 | <p>Validate License Number</p> <p>Use to enforce validation that the License # is populated before launching the Signature Capture display.</p> | Enforce the License Number validation. | Skip the License Number validation. | Y |
| 3 | <p>Validate License Number Status</p> <p>Use to enforce validation that the License # status = Active before launching the Signature Capture display.</p> | Enforce the License Number Status validation. | Skip the License Number Status validation. | N |
| 4 | <p>Validate License Number Expiration</p> <p>Use to enforce validation that the Expiration date on the License # is later than Today before launching the Signature Capture display.</p> | Enforce the License Number Expiration validation. | Skip the License Number Expiration validation. | N |
| Not applicable | <p>Validate Address DEA Number</p> <p>Use to enforce that the DEA number is valid (DEA # checksum algorithm) at the time the value is entered into the DEA # field on the Address child applet (on the Handheld application).</p> <p>This user property applies to both the laptop and the Handheld applications.</p> <p>NOTE: DEA stands for Drug Enforcement Administration.</p> | Enforce the Address DEA Number validation (field is populated <i>and</i> number is valid). | Skip the Address DEA Number validation (field is populated <i>and</i> number is valid). | Y |

Table 20. User Properties for Setting Sign and Submit Button Logic

| Flowchart Reference | User Property | Action if Y... | Action if N... | Default Value |
|---------------------|--|---|---|---------------|
| 6 | <p>Validate Signature DEA Number</p> <p>Use to enforce that the DEA number is populated and that the value is valid (DEA # checksum algorithm) before launching the Signature Capture display.</p> | Enforce the DEA Number validation in the Signature button (field is populated <i>and</i> number is valid). | Skip the DEA Number validation in the Signature button (DEA number is valid for each checksum routing <i>and</i> the field is populated). | Y |
| 7 | <p>Validate DEA Number Expiration</p> <p>Use to enforce validation that the Expiration date on the DEA is later than Today before launching the Signature Capture display.</p> | Enforce the DEA Number Expiration validation in the Signature button (expiration date is later than today). | Skip the DEA Number Expiration validation in the Signature button (expiration date is later than today). | N |
| 8 | <p>Validate Professional Profile</p> <p>Use to validate if the professional's Last Name, First Name, Title, and Address fields are populated before launching the Signature Capture display.</p> | Enforce the Professional Profile validation. | Skip the Professional Profile validation. | Y |
| 9 | <p>Validate Sign Lot Number</p> <p>Use to validate if the lot number is populated on launching the Signature Capture display.</p> | Enforce the Signature button Lot Number validation. | Skip the Signature button Lot Number validation. | N |
| 10 | <p>Validate Submit Lot Number</p> <p>Use to validate if the lot number is populated on submitting a call.</p> | Enforce the Submit button Lot Number validation. | Skip the Submit button Lot Number validation. | Y |
| 11 | <p>Sample Disbursed Required</p> <p>Use to require that at least one sample be disbursed before submitting the call.</p> | Enforce that at least one sample be disbursed. | Do not require at least one sample be disbursed. | N |
| 12 | <p>Must Detail Products</p> <p>Use to require that at least one product be detailed before submitting the call.</p> | Enforce that at least one product be detailed. | Do not require at least one product detailed. | N |

Table 20. User Properties for Setting Sign and Submit Button Logic

| Flowchart Reference | User Property | Action if Y... | Action if N... | Default Value |
|---------------------|--|---|---|---------------|
| 13 | Paper Reference Number Required Use to require a sample reference number for calls with paper signatures. | Enforce the Paper Reference Number required validation. | Skip the Paper Reference Number required validation. | Y |
| 14 | SRE Reference Number Required" Use to require a sample reference number for signature recorded electronically (SRE) calls. | Enforce the SRE Reference Number required validation. | Skip the SRE Reference Number required validation. (If the Ref # field is blank, it will be populated with the Row ID of the call.) | N |
| 15 | SRE Professional Type Use to require that the professional type be valid for signature recorded electronically (SRE) calls. | The professional type (for example, Physician) must match the value of this user property. If the value is null, all professional types are valid. | | null |
| Not applicable | Submit Confirmation Note: If this user property does not exist (for example, in the standard Siebel Pharma application), behavior is equivalent to Submit Confirmation = Y. | Confirmation message appears when the Submit button is tapped. | No confirmation message appears when Submit button is tapped. | Y |
| Not applicable | Submit On Sign OK Note: If this user property does not exist (for example, in the standard Siebel Pharma application), behavior is equivalent to Submit On Sign OK = N. | Tapping the OK button on the Siebel Signature Capture display for new professional calls invokes the submit method. The submit method is not invoked for account calls nor for re-created professional calls. | Tapping the OK button on the Siebel Signature Capture display closes the display. To submit the call, tap the Submit button. | N |

Validating the DEA Number in Siebel Pharma Handheld Applications

In the U.S., the Drug Enforcement Administration (DEA) number is a federally authorized number for physicians with a practice at a specific address.

In Siebel Pharma Handheld, the DEA number is stored at the address level for each physician. The Siebel Pharma Handheld application can confirm that a valid DEA number is populated at the selected address for a professional call. Depending on how your Handheld application is configured, the valid DEA number check can be performed at the time of entering the DEA number, upon launching the Siebel Signature Capture display, or both.

If both of the following are true:

- The product dropped is a controlled substance.
- The user property Validate Address DEA Number or Validate Signature DEA Number is true.

then Siebel Pharma Handheld checks the validity of the DEA number by executing the algorithm described in [Table 21](#).

Table 21. The DEA Number Checksum Algorithm

| Step | Algorithm | Example |
|------|---|---|
| 1 | The first two characters must be uppercase letters. | <i>BB1388568</i> |
| 2 | Sum the first, third, and fifth digits. | $1 + 8 + 5 = 14$ |
| 3 | Sum the second, fourth, and sixth digits. | $3 + 8 + 6 = 17$ |
| 4 | Double the number obtained in step | $2 \times 17 = 34$ |
| 5 | Sum the numbers from steps 2 and 4. | $14 + 34 = 48$ |
| 6 | If the last digit of the DEA number is the same as the last digit of the number obtained in step 5, the DEA number is considered valid. | The last digit of <i>BB1388568</i> and <i>48</i> is the same, "8". This checksum indicates that the DEA number is valid. |

Preparing Receipts for Calls in Siebel Pharma Handheld Applications

As the Siebel administrator, you might be asked to print receipts for sample drops to a physician. From the Sample Event Administration view, you generate a list of sample drops and print receipts using the Sample Receipt report.

To print receipts for samples dropped

- 1 Navigate to the Administration - Samples screen > Sample Event Administration view.

- 2 In the Sample Events list, create a query.
- 3 Navigate to Reports
- 4 From the Select a Report drop-down list, select Sample Receipts and click Run.
- 5 To print the receipt, click the Print button in the Siebel Report Viewer window.

5

Data Filtering for Siebel Handheld Applications

The purpose of this chapter is to outline how to develop data filters for your Siebel Handheld application. It includes the syntax for constructing business components and business object filters, as well as some useful information on managing your business object filters. The following topics are covered in this chapter:

- [“Developing Data Filters for Siebel Handheld Applications” on page 159](#)
- [“Constructing Business Component and Business Object Filters for Siebel Handheld Applications” on page 160](#)
- [“Business Object Filters for Siebel Handheld Applications” on page 161](#)
- [“Recommendations for Managing Business Object Filters for Siebel Handheld Applications” on page 161](#)

Developing Data Filters for Siebel Handheld Applications

The following general steps are recommended for developing synchronization filters for Siebel Handheld applications.

- Identify the screens and views of your application. Then identify the business components that are required to support each screen and view, noting the areas where each business component occurs. For example, identify the form, list, or pick applets that are based on each business component.
- Within Siebel Tools, establish the base parent-child relationships and any reverse relationships at the component level.
- Review the business components that occur in picklists and consider creating business component filters for those that will cause large numbers of records to synchronize to the device. These filters are created and maintained in the Mobile Administration screens. For each business component, determine whether it is possible to apply a search specification to all instances of the business component within the Siebel Handheld application. Develop a set of suitable business component filters in the Mobile Administration screens.

Common examples of business components that are suitable for global business component filters include:

- Employees (for large organizations)
- Activities (for example, filter for current Activities, Activities of only certain kinds)
- Accounts (for example, by geography)
- Products (for example, active, orderable products)

- Orders (for example, open, closed in last three months)
- Create appropriate business object synchronization filters for each business object and make sure that each business object includes a default filter. Default filters make sure that users avoid accidental download of an excessive number of records.

NOTE: In a multilingual environment, default filters are created for the primary language only. If your environment is multilingual, then you must configure additional filters for each additional language you want to run.

Constructing Business Component and Business Object Filters for Siebel Handheld Applications

Business component filters are used to limit the extraction of a business component for every context in which it is extracted.

Example of Business Component Filter Syntax:

Business component filters use query by example (QBE) syntax:

```
[Status] = "Not Started" OR [Status] = "In Progress" OR ([Status] IS NULL) OR  
[Status] = "On Hold" OR ([Status] = "Done" AND [Done] >= Today() - 7)
```

For more information about QBE syntax, see *Siebel Developer's Reference*.

Business object filters are used to supply filters that a user can select at synchronization time. These are the filters that are displayed in the SetFilters dialog on the device.

Example of Business Object Filter Syntax:

Business object filters use predefined query (PDQ) syntax.

PDQ syntax rules are as follows:

- The business component name must be a valid business component in the business object on which the PDQ is based
- Search and Sort can be specified for all active buscomps in the business object on which the view is based, appended to each other in any order using space as delimiter.

The normal search specification and sort specification syntax applies except that quotes in the search specification have to be escaped using double quotes because the search specification itself is enclosed in double quotes.

The syntax is as follows:

```
'Account'.Search = "[State] = LookupValue ( ""STATE_ABBREV"", ""CA"" )" 'Account'.Sort =  
"Name, Location"
```

Descriptions of the terms are shown in [Table 22](#).

Table 22. Business Object Filter Syntax

| Syntax Term | Description |
|--|---|
| Account | Business component name. |
| Search | Key word. |
| "[State] = LookupValue (""STATE_ABBREV"", ""CA"")" | Search specifications using PDQ syntax. |
| State | Field name. |
| LookupValue | Function for find and LOV value. |
| STATE_ABBREV | LOV type. |
| CA | LOV value. |
| 'Account'.Sort = "Name, Location" | Sort specification for sorting the returned list. |

For more information about PDQ syntax, see *Siebel Developer's Reference*.

Business Object Filters for Siebel Handheld Applications

All business object filters for Siebel Handheld applications are available for selection by a user at the time of synchronization. Additionally, there is no naming convention for business object filters.

A default filter for a business object is indicated by checking the Default flag in the Administration - Mobile screen, Application Administration, then the Business Object Filters view of the Siebel Web Client. Be sure to define a default filter for each business object in the Handheld application. If no filters are applied during the synchronization process, an unacceptably high volume of data will be downloaded to the handheld device.

NOTE: In Siebel 7.7, a required field is always extracted in to the handheld device even if it is not exposed in the UI or explicitly defined in Mobile Administration.

CAUTION: You must select only one filter as the default for each business object. If two filters are marked as the default for the same business object, the actual default can be determined only by proceeding to synchronize and select the default filters from the selection UI.

Recommendations for Managing Business Object Filters for Siebel Handheld Applications

The following topics provide recommendations for creating and managing filters.

- "Companion Synchronization Deployments" on page 162

- [“Direct Server Synchronization Deployments” on page 162](#)
- [“Examples of Default Business Object Declarations” on page 162](#)
- [“Designating Visibility” on page 164](#)
- [“Overriding the Popup Visibility” on page 165](#)
- [“Multiple Visibility Methods” on page 165](#)

Companion Synchronization Deployments

Administrators typically provide their users with an initial set of business object filters for the Handheld application and instruct them how to manage the synchronization filters themselves later. They are maintained in the Administration - Mobile screen, Application Administration, then the Business Object Filters view. This view is hidden from mobile users so they cannot create their own business object filters.

You manage these filters at the server level, then replicate any updates you make down to the companion computers via Siebel Remote. The Handheld user can then access the updated filters after performing a Companion Synchronization operation. For more information about filters, see [“Setting Business Object Filters for Siebel Handheld Application Users” on page 102](#).

Additionally, there are several sample business object filters for Handheld applications in the sample database, as well as in the Siebel 7.8 seed data.

Direct Server Synchronization Deployments

For Direct Server Synchronization deployments, administrators can create user-specific business object and business component filters in the Administration - Mobile screen, then the User Administration view. For more information about creating business objects and business component filters, see [“Setting Business Component Filters for Siebel Handheld Application Users” on page 101](#) and [“Setting Business Object Filters for Siebel Handheld Application Users” on page 102](#).

Both application-level and user-level filters can be used in this deployment. They are managed in the same set of views. Changes to any filters will be effective on your next synchronization session, without having to restart the synchronization OM component for the Handheld application.

These business object filters are made available to all users and, therefore, must be defined with the needs of all users in mind.

Examples of Default Business Object Declarations

You can set default business object declarations in the Administration - Mobile screen, Application Administration, then the Settings view, as described in the following examples.

Example 1

“State Model” default business object setting:

```
State Model - Engine|0||State Model |[BusComp Name]="Expense" |State Model -  
Transi ti on||State Model - State||State Model - Posi ti on|
```

The first three segments of the declaration are followed by a series of repeating BusCompName search specification pairs:

- 1 Business object name
- 2 Visibility Type
- 3 (Siebel internal)
- 4 Root business component name
- 5 Search specification QBE for root BusComp records
- 6 Child business component name
- 7 Search specification QBE for child BusComp records (this must normally be empty, causing all children to be extracted)
- 8 Grandchild business component name
- 9 Search specification QBE for grandchild BusComp records (this must normally be empty, causing all grandchildren to be extracted), and so on.

The total number of fields must be at least 5 and must be an odd number.

Example 2

```
Acti on|0||Acti on| |Acti on Empl oyee|
```

This example default business object declaration results in an extraction of the Action and Action Employee business components where they occur with visibility 0 using the business object filter indicated. The business object filters are generally used only for extracting business components that do not correspond to any views or applets. Business object declarations identify specific business components within the business object to be extracted. The business object filters normally give a query based on the primary business component in the business object and force extraction of any business components mentioned in the query when there is no corresponding applet within the application that can cause the desired extraction to occur.

Example 3

In the following example, the declaration is used to extract all the Price List Line Items under the Order Entry (Sales) business object.

```
Defaul tBusObj s13= Order Entry (Sales)|0||Pri ce Li st I tem|
```

Note that this declaration lacks an associated QBE clause. When this is the case, all of the records available to the user for the given visibility are extracted.

Exercise caution when modifying a default business object setting for pick applets, and do so only with the support of a skilled Siebel configurator.

Use default business object declarations to force the extraction of data for a given business component in the context of a given business object and Visibility type. For more information about visibility designations, see [“Designating Visibility” on page 164](#). The default business object extractions are performed in the first phase of data extraction.

You will most often use these declarations to extract additional data that is not extracted as part of the user interface extraction. This can be useful for business components that are referenced programmatically by other business components, but are not exposed in the user interface. The default business object declarations are used to enforce the extraction of a subset of data to the Handheld application that might not have been otherwise extracted or, most commonly, to filter the data downloaded for dynamic picklists or pick applets.

All pick data is not downloaded to the device due to the memory constraints on the device. Only pick data for fields that are editable is downloaded. Therefore, if you are doing a query on a pick field, you might not be able to display the picklist. When viewing a record in an applet, if a pick field is read-only, its pick button will be disabled. If users select a read-only field with a pick control, they receive the following error: No data available in picklist because field is read-only.

Designating Visibility

It is very important that you use the correct visibility designation in the default business object declarations. The visibility entered must be the visibility that governs the data displayed for the business component within the context of the business object specified in the declaration. In the previous Example 1, visibility code 3 is designated. This corresponds to All Visibility, which is the visibility that applies to the Account business component when it is used in a pick applet.

You determine the correct visibility to use by examining View and business component properties within Siebel Tools. When creating a default business object declaration to restrict the volume of data extracted for a pick applet, begin by examining the Popup Visibility Type in Business Component Properties. Cross-reference the visibility type to the list below, and enter the appropriate visibility code in the declaration. If the Popup Visibility Type field on the business component is null, the default value is All. [Table 23](#) summarizes popup visibility types and the corresponding visibility codes.

Table 23. Visibility Codes

| Popup Visibility Type | Visibility Codes |
|-----------------------|------------------|
| Sales Representative | 0 |
| Manager | 1 |
| Personal | 2 |
| All | 3 |
| None | 4 |
| Organization | 5 |
| Contact | 6 |

NOTE: Do not use the Group, Catalog, or Sub Organization Popup Visibility Types. These are not supported in the CE Handheld application.

Overriding the Popup Visibility

If the Popup Visibility Type setting is not sufficiently restrictive (for example, All Visibility), this can lead to a large number of records being extracted when the picklist is processed during synchronization. Making the Popup Visibility Type more restrictive reduces the number of records extracted for picklists. The `OverridePopupVisibility` user property for an applet overrides the Popup Visibility Type on a business component. Use this property when the desired visibility differs from the business component's Popup Visibility Type, and you do not want to change the Popup Visibility Type. If several picklists use the same business component, you can also use the `OverridePopupVisibility` user property to vary the visibility of the picklists in the different applets.

There are two considerations you must keep in mind if multiple applications are sharing the same repository. First, changing the Popup Visibility Type has a global effect for all instances where the business component is used. If multiple applications share the same repository and use the same business components, changing the Popup Visibility Type for a common business component might have an undesirable effect for another application. Second, if multiple applications share the same applet, the picklist in each applet shares the same visibility as defined by `OverridePopupVisibility` property. This is because the `OverridePopupVisibility` property is set on the containing applet.

In Siebel Tools, navigate to Applet, then the Applet User Properties, and define the `OverridePopupVisibility` user property on the applet containing the control that opens the pick applet. For more information about this property, see [OverridePopupVisibility User Property on page 317](#).

If `OverridePopupVisibility` is used, then the picklist and pick applet search specifications are ignored during the synchronization when the records for the picklist are extracted.

Multiple Visibility Methods

When a record is extracted by means of different visibilities, the record is tagged with each visibility. For example, if an individual contact record is extracted under Org visibility, and later the same contact record is extracted under Salesrep visibility, the visibility field on the record contains a bitmask indicating that the two visibilities apply for this record. Handheld synchronization extracts data in the following order:

- DefaultBusObjs
- Views
- PickList

You can check the visibilities for a record in the dbfile.txt file. [Table 24](#) lists the values for the visibility bitmasks.

Table 24. Visibility Bitmask Values

| Visibility Bitmask | Values |
|-------------------------|--------|
| VISIBILITY_SALESREP | 1 |
| VISIBILITY_MANAGER | 2 |
| VISIBILITY_PERSONAL | 4 |
| VISIBILITY_ALL | 8 |
| VISIBILITY_NOVISIBILITY | 16 |
| VISIBILITY_ORG | 32 |
| VISIBILITY_CONTACT | 64 |

For example, the visibility for a record extracted for both SalesRep and Org would have a visibility set to 33.

NOTE: Not all visibilities are supported on the handheld device.

Specifically, Group, Catalog and Suborg visibilities are not supported. These are mapped to different visibilities as follows:

- Group -> All
- Catalog -> Org
- SubOrg -> Org

6

Deploying Siebel Handheld Applications

This chapter provides important information about how to deploy a Siebel Handheld application, and includes information about server topology, hardware, network configuration, server installation, Siebel Handheld application installation, and performance and scalability. The following topics are covered in this chapter:

- [“Overview of Deployment for the Siebel Handheld Application Client” on page 167](#)
- [“Installing the Siebel Server for Direct Server Synchronization” on page 169](#)
- [“Configuring the Siebel Server for Direct Server Synchronization” on page 171](#)
- [“Performance and Scalability of Synchronization for Handheld Applications” on page 173](#)
- [“Providing Synchronization Security with Secure Sockets Layer” on page 174](#)
- [“Process of Installing the Siebel Handheld Application” on page 175](#)
- [“Installing Print Templates in the Siebel Handheld Install Directory” on page 181](#)
- [“Setting Up Siebel Handheld Application Installation on a CompactFlash Card” on page 182](#)
- [“Changing the SyncURL on Handheld Devices” on page 182](#)
- [“Installing the Siebel Handheld Application from External Media” on page 183](#)
- [“Distributing Siebel Handheld Application Patches” on page 183](#)
- [“Setting up PatchAgent on Windows 2003” on page 192](#)

Overview of Deployment for the Siebel Handheld Application Client

This topic provides an overview of important factors to consider when planning a Siebel Handheld application deployment using Direct Server Synchronization (DSS). One aspect of successful planning and execution of a Handheld application deployment using DSS is to understand hardware and network requirements. While these factors are generally applicable to all deployments, there are unique aspects to each customer configuration and implementation strategy. Therefore, it is recommended that customers conduct detailed configuration, sizing, and production readiness reviews. For help with detailed configuration, sizing, and production readiness reviews, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services. This is especially important for complex deployments, such as those supporting large numbers of users over wide geographic areas.

You can also improve the performance and scalability of DSS deployments by working with the Siebel application server component parameters. For more information about the Siebel application server, see [“Optimizing Siebel Server Process Management for Direct Server Synchronization” on page 173](#).

NOTE: The recommendations in this chapter are general ones and are intended to raise your awareness of the key factors in Siebel Handheld application synchronization, performance, and scalability. These factors must be confirmed for your actual environment, and the usage patterns in your enterprise must be taken into consideration.

Siebel Server Topology Overview

The diagram in Figure 7 shows the servers and components that are required to deploy Direct Server Synchronization for a Handheld application. The components are the Siebel Handheld application database, Siebel Server, Siebel Web engine, and Siebel Handheld application. The synchronization components are the synchronization engine for the Siebel Handheld application which resides on Siebel Server, and the Siebel Handheld Synchronization Client which resides on the Siebel Handheld application.

CAUTION: Companion Synchronization uses Siebel Remote to complete the synchronization process to the Siebel Server from the laptop. The remote mobile user's routing model must be set to Handheld User. Otherwise, synchronization fails. For more information about Siebel Remote, see *Siebel Remote and Replication Manager Administration Guide*.

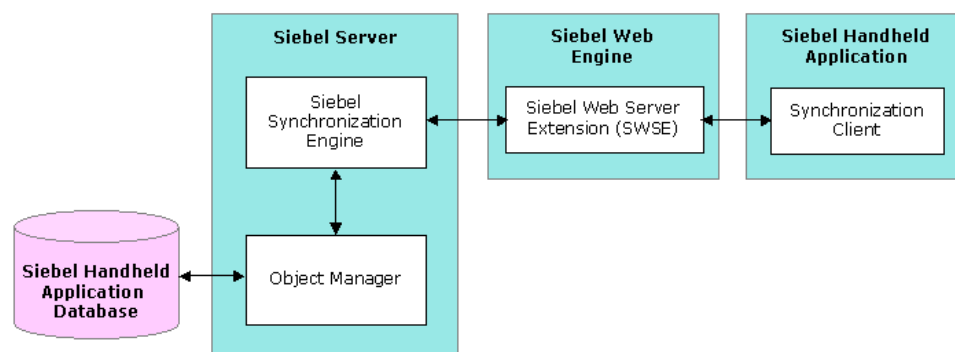


Figure 7. Servers and Components

Recommended Hardware and Network Configuration

There are many factors to take into account when designing and implementing your server and network hardware and architecture, and sizing your Siebel Handheld application database. The critical factor in sizing the implementation for a Siebel Handheld application is the expected peak synchronization load. Specific Handheld application variables to consider in calculating this load are: the number of concurrent users, number of transactions, and size of the data file. These variables must be considered within the context of the hardware specifications, including memory and processor speed.

The following resources are available as you plan the implementation of your Siebel Handheld application:

- Refer to Microsoft and manufacturer documentation for comprehensive information on server hardware and registry parameters.

- For specific recommendations on the following topics, refer to Bulletin 476964.1 on *My Oracle Support*. This document was previously published as Siebel Technical Note 405.
 - Server architecture
 - Server hardware and registry parameters
 - Database server sizing
 - Database tuning
 - Dial-Up and network
 - Wireless
 - VPN (Virtual Private Networks)
- The Handheld Review Package assists customers with planning their hardware and network implementations for Handheld applications. For help with Handheld Review Package, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Installing the Siebel Server for Direct Server Synchronization

The components that are required for Siebel Handheld applications are installed when you install the Siebel Server.

This topic includes the following topics:

- ["Installing the Components for Direct Server Synchronization Using the Siebel Server Installer" on page 169](#)
- ["Siebel Server Component Parameters for Direct Server Synchronization" on page 170](#)

For information about installing and configuring Siebel Server, see *Siebel Installation Guide* for the operating system you are using.

Installing the Components for Direct Server Synchronization Using the Siebel Server Installer

Install the Siebel Server, following the instructions in the Siebel Server installation guide for the operating system you are using. When you reach the Setup Type screen in the Installer wizard, you are presented with three installation options:

- **Typical.** This setup option installs all Siebel Server components except those displayed.
- **Compact.** This setup option installs only those components necessary to run the Siebel Server, but no additional components or help.
- **Custom.** This setup option lets you customize your installation by choosing the specific components you want to install.

To install the components for Direct Server Synchronization with your Siebel Server installation

- 1 During the Siebel Server installation, when the Setup Type screen appears, choose the Custom setting, and then click Next.

NOTE: At this screen, you can also choose a different directory for your Siebel Server installation.

- 2 From the following components, choose Handheld Synchronization:

- Object Manager Component
- Handheld Synchronization
- Siebel Data Quality Connector
- Remote Search Support
- Siebel Java Integrator

NOTE: You must enable synchronization on your handheld device to enable Direct Server Synchronization for your installation.

- 3 Click Next.

Continue with the installation in the *Siebel Installation Guide* for the operating system you are using.

The install wizard automatically installs and configures the components for Direct Server Synchronization.

Siebel Server Component Parameters for Direct Server Synchronization

See [Table 25](#) for a list of required parameters. For information on adding server components for Direct Server Synchronization, see *Siebel System Administration Guide*.

Table 25. Siebel Server Component Parameters for Direct Server Synchronization

| Parameter | Default Value | Description |
|------------------|------------------|---|
| ApplicationName | Siebel Pharma CE | The Siebel Handheld application. This value must match the name of the Handheld application as it is specified in Siebel Tools. |
| ApplicationTitle | Siebel Pharma CE | The name of the Siebel Handheld application as shown on the title bar on the browser. This parameter is only used if the Siebel application developed for the handheld device is run on the Siebel Mobile Web Client. |

Table 25. Siebel Server Component Parameters for Direct Server Synchronization

| Parameter | Default Value | Description |
|-----------------------|------------------|---|
| ApplicationSplashText | Siebel Pharma CE | The Siebel Handheld application name as shown on the splash screen. This parameter is only used if the Siebel Tools application developed for the handheld device is run on the Siebel Mobile Web Client. |
| ComponentName | | Required for Direct Server Synchronization, this parameter is entered in the registry on the handheld device. |

Configuring the Siebel Server for Direct Server Synchronization

The Siebel Server installer copies the necessary DLLs to create an Object Manager for Direct Server Synchronization and creates and configures a Server Component and Component Group. This topic describes which options you must select to correctly install the software, along with creating or configuring additional Object Managers.

This topic includes the following topics:

- [“Configuring Logging Levels for Direct Server Synchronization” on page 171](#)
- [“Configuring Logging Levels for Companion Synchronization” on page 172](#)
- [“Optimizing Siebel Server Process Management for Direct Server Synchronization” on page 173](#)

Configuring Logging Levels for Direct Server Synchronization

To set the logging level, you need to perform the following two tasks:

- [“To configure Siebel Server logging levels” on page 172](#)
- [“To configure Siebel Server logging levels using the Server Configuration screen” on page 172](#)

You set the logging level to a value between 1 and 5 as shown in [Table 26](#).

Table 26. Logging Levels for Direct Server Synchronization and Companion Synchronization

| Logging Level | Results |
|---------------|-----------------|
| 1 | Error listing |
| 2 | Warning listing |

Table 26. Logging Levels for Direct Server Synchronization and Companion Synchronization

| Logging Level | Results |
|---------------|--|
| 3 | Information listing, including SQL tracing |
| 4 | Information listing |
| 5 | Debug |

To configure Siebel Server logging levels

- 1 Log into Server Manager from a (m: \siebel\bin\w32ud) prompt.
- 2 Run the following:

change evtlog\HandheldSyncLog=[LogLevel] for comp [OMName], where [LogLevel] = an integer between 1 - 5 (5 enables all logging) and [OMName] = the OM name (for example, <ApplicationName>CEObjMgr_enu)

Use the following procedure to configure Siebel Server logging levels using the Server Configuration screen.

To configure Siebel Server logging levels using the Server Configuration screen

- 1 Navigate to Administration - Server Configuration screen, Servers view, then the Components view tab.
- 2 In the Component list, query for the appropriate component, such as Handheld Sales CE (ENU).
- 3 In the Events list, search for the event type Handheld Sync Logging.
- 4 Set the value to a number between 1-5 and then save the change.

See [Table 26](#) for logging level results.

Configuring Logging Levels for Companion Synchronization

To enable logging for Companion Synchronization, you must add a parameter to the [pda] section of the CFG file.

To configure logging levels for Companion Synchronization

- 1 Navigate to \client\bin\enu (or other language directory, such as \client\bin\fra).
- 2 Add the parameter HandheldSyncLogLevel and a valid value (1-5) to: phce.cfg. For example, HandheldSyncLogLevel = 5.

See [Table 26](#) for logging level results.

NOTE: To enable Object Manager logging, add ObjMgrSqlLogLevel to the PDA section of your Siebel Handheld application's CFG file. Valid values range from 1-5 as described in [Table 26](#) on page 171.

Optimizing Siebel Server Process Management for Direct Server Synchronization

Several parameters are outlined below that help optimize Siebel Server processing for Direct Server Synchronization.

To set parameters to optimize Siebel Server process management

- 1 Log into the Siebel application as a user with Administrative responsibility.
- 2 Navigate to the Administration - Server Configuration screen, Servers view, Components, then the Parameters view tab.
- 3 In the Component Parameters sub-view, query for SalesCEObjManager.
- 4 Use the following table to update the parameters.

| Parameter | Description |
|----------------|---|
| Max Tasks | Determines the maximum number of threads that can run concurrently for a given component. Set the value of this parameter to a number equal to that of the expected peak number of concurrent synchronizations for each Siebel Server. |
| Min MT Servers | Determines how many multithreaded server processes are started by default for the Siebel Object Manager. |
| Max MT Servers | Defines the maximum number of processes supported by the instance of the Siebel Object Manager. The value for this parameter must be the same as Min MT servers and must be set so that threads for each process (Max Tasks/Max MT Servers) is optimized. |

Performance and Scalability of Synchronization for Handheld Applications

There are a number of factors that affect the performance and scalability of synchronization in Siebel Handheld applications. It is very important that administrators and those involved in the design and deployment of Siebel Handheld applications understand the impact of usage parameters on synchronization performance and scalability. For more information, see Bulletin 476964.1 (Siebel Technical Note 405) for application-specific variables affecting synchronization and data from synchronization scalability tests.

Using Multiple Siebel Handheld Synchronization Servers

As you scale your deployment, you might need to use multiple Siebel Handheld Synchronization Servers to accommodate all of your users. Your users must be manually assigned to a particular Siebel Handheld Synchronization Server. Resonate products cannot be used for load balancing. For more information about scaling deployments, refer to Bulletin 476964.1 (Siebel Technical Note 405) and to your hardware manufacturer's documentation.

Providing Synchronization Security with Secure Sockets Layer

The synchronization data stream can be secured using Secure Sockets Layer (SSL). To employ SSL security:

- Make sure that a certificate has been created and installed on all Web Servers that service synchronization in Handheld applications.

Use the following steps to install the root certificate on each handheld device.

NOTE: PPC 2002 and PPC 2003 do not have a hosts file similar to the Windows Operation System (for example, C:\WINNT\system32\drivers\etc\hosts. file), where you can set up mappings for the IP address and host name.

To install the root certificate on the handheld device running PPC 2003 or newer

- 1 Navigate to the certificate authority server from your desktop computer.
- 2 Select the first option (retrieve the CA certificate or certificate revocation list), and click Next.
- 3 Click on Download CA certificate.
- 4 Click Save, and save as certca.cer on your desktop.
- 5 Use ActiveSync to copy the downloaded certificate, certca.cer, from the desktop to the My Documents folder on the handheld device.

NOTE: Make sure the certca.cer file is copied into the My Documents folder.

- 6 Go to My Documents and tap the certificate icon.

The certificate must be installed. You can validate it by going to Settings, System, then the Certificate.

- 7 Click the Root tab.

The new certificate is displayed on the list.

- 8 Open a browser, and run the following command, substituting the appropriate server name, application name, UserName and Password.

```
https://<SERVERNAME>/<APPLICATION_NAME>/  
start.swe?SWEExtSource=siebel pda&SWEExtCmd=ExecuteLogi n&UserName=<USER  
LOGI N>&Password=<PASSWORD>
```

- 9 When a dialog box appears, click Yes.

To complete root certificate installation on PPC 2002 device

- 1 From a Web browser, navigate to the Microsoft Web site and download and launch the following executable file:
AddRootCert.exe
- 2 Double-click to open the executable file.
A list of certificates appears.
- 3 Select certca.cer.
Information for the certca.cer Certificate appears.
- 4 Click Install Certificate.
- 5 Following successful installation, click OK.

Process of Installing the Siebel Handheld Application

Before you install the Siebel Handheld application, perform the following tasks:

- ["Editing the Siebel Handheld Application Installation Configuration File" on page 175](#)
- ["Enabling Back Up of Siebel Handheld Application Database" on page 180](#)
- ["Deploying Patches to Handheld Applications with Direct Server Synchronization" on page 184](#)
- ["Deploying Handheld Application Patches with Companion Synchronization" on page 186](#)
- ["Creating Handheld Application Upgrade Kits for Companion Synchronization PDA" on page 188](#)
- ["Deploying Attachments to Handheld Applications with Direct Server Synchronization" on page 188](#)
- ["Deploying Handheld Application Attachments with Companion Synchronization" on page 190](#)
- ["Downloading Patches to the Handheld Device" on page 192](#)
- ["Setting up PatchAgent on Windows 2003" on page 192](#)

NOTE: If you are setting up your Siebel Handheld application to allow end user translation functionality, see ["Installing and Setting Up End User Translation Functionality" on page 81](#).

Editing the Siebel Handheld Application Installation Configuration File

This task is a step in ["Process of Installing the Siebel Handheld Application" on page 175](#).

The Siebel Handheld application setup.ini file configures the Siebel Handheld Client installer. Before you deploy a custom-configured Siebel Handheld application, you must update the parameters in the setup.ini file so that the correct values are written to the device registry before distributing the Siebel Handheld Client installer to users.

The setup.ini file resides in the Siebel Handheld application directory. To modify this file, create a Handheld application installation directory on the network, or create and duplicate a custom DVD-ROM for distribution to users. This installation directory must include the same files and directory structure as the original DVD-ROM. Then, modify the setup.ini file in the custom installation directory.

NOTE: If you wish to set up parameter values other than default values, add the parameter to the setup.ini file and add the proper values.

Table 27 lists the parameters that you might need to modify in the setup.ini file, their default value, and a description of the parameter.

Table 27. Siebel Handheld Application setup.ini Parameters

| Parameter | Default Value | Description |
|-------------------------------|---|--|
| Version | 8.0 | Version of the desktop Siebel application against which the Siebel Handheld application version synchronizes. |
| ApplicationName | Siebel Pharma CE | The Siebel Handheld application. This value must match the name of the Handheld application as it is specified in Siebel Tools. |
| ApplicationSplashText | Pharma CE Handheld | The Siebel Handheld application name as shown on the splash screen. |
| EnableRemote | N | Used for Integrated Sync. The default value is N. Valid values are Y = [Enable Remote Sync] check box is selected, N = [Enable Remote Sync] check box is unselected in the synchronization window of the Handheld application. |
| MultiCurrency | True | Required for the Handheld application to handle multiple currencies. |
| SyncURL | http://Web Server Name or IP address/Virtual Directory Name | Required for all synchronization methods. This value is entered in the registry for the Siebel Handheld application. NOTE: PPC 2002 and PPC 2003 do not have a hosts file similar to the Windows Operation System (for example, C:\WINNT\system32\drivers\etc\hosts file), where you can set up mappings for the IP address and host name. For more information about this parameter, see "Changing the SyncURL on Handheld Devices" on page 182. |
| [EndUserLocalization] Section | | |

Table 27. Siebel Handheld Application setup.ini Parameters

| Parameter | Default Value | Description |
|-------------------------------|---------------|--|
| User DefinedLang | Y | |
| LogUserDefinedLangErrors | N | |
| [Printing] Section | (various) | For more information about the parameters used to configure printing, see Appendix F, "Print Configuration Settings." |
| [Backup] Section | (various) | Enables Siebel Handheld application database backups. For more information about settings for the parameters in this section of the setup.ini file, see "Enabling Siebel Handheld Application Database Backup" on page 180. |
| [ApplicationLocation] Section | (various) | Specifies where the Siebel Handheld application and database are installed. For more information about the parameters in this section of the setup.ini file, see "Setting Up Siebel Handheld Application Installation on a CompactFlash Card" on page 182. |
| InternetOptionReceiveTimeout | 600000 | Length of time to receive data or a response (except data extraction) from the server in the case of Direct Server Synchronization or the Laptop in the case of Companion Synchronization, before the Siebel Handheld application times out. The default value, 600000 milliseconds, is 10 minutes. |
| InternetOptionSendTimeout | 600000 | Length of time to send data to the server in the case of Direct Server Synchronization or to the Laptop in the case of Companion Synchronization, before the Siebel Handheld application times out. The default value, 600000 milliseconds, is 10 minutes. |

Table 27. Siebel Handheld Application setup.ini Parameters

| Parameter | Default Value | Description |
|--------------------------------------|---------------|--|
| InternetOptionDataExtraction Timeout | 900000 | Length of time to extract data from the server in the case of Direct Server Synchronization or from the Laptop in the case of Companion Synchronization (to the handheld device), before the Siebel Handheld application times out. The default value, 900000 milliseconds, is 15 minutes. |
| RemoteTimeout | 1800000 | Length of time for the handheld device Syncmanager process to wait for a remote synchronization to complete, before timing out. The default value, 1800000 milliseconds, is 30 minutes. NOTE: This parameter only applies to synchronizations using Companion Synchronization and when remote synchronization is enabled. |
| [Logging] Section | | |
| TXNErrorLevel | 1 | This level controls the extent of transaction error information received from the server. The default value is 1. Valid values are 0 = no log information, 1 = report txn errors in the Siebel Handheld Client log, txnerror.hh. A "Transaction Errors Reported" message appears during synchronization when txn errors occur. To view errors, double click the message. |

Table 27. Siebel Handheld Application setup.ini Parameters

| Parameter | Default Value | Description |
|-------------------|---------------|---|
| DefaultLogLevel | 2 | <p>The default value is 2. You can set the value to any of the following:</p> <ul style="list-style-type: none"> -1 = NO LOGGING 0 = LOG_SYSTEM 1 = LOG_ERROR 2 = LOG_WARNING 3 = LOG_INFO, INCLUDING SQL TRACING 4 = LOG_DEBUG <p>The log messages with level equal to or lower than the set value will be logged. For example, if the default log level is set to 2, the messages belonging to System, Error, and Warning log levels will be logged. If the DefaultLogLevel is -1, no logging will be provided.</p> |
| MaxLogFileSize | 2000 | <p>This is the maximum log file size in KB. If the program finds the size of the latest log file is greater than this value, a new log file must be used for logging.</p> |
| MaxSavedLogFiles | 3 | <p>Determines the number of log files to save on the device. It does not include the one currently used by the program. If the value is set to 2, then the valid log file names are siebel_log.txt, Siebel_log1.txt, and siebel_log2.txt. Siebel_log.txt is the current log file. Siebel_log1 is an older one and Siebel_log2 is the oldest.</p> |
| PressureThreshold | 29 | <p>Prompts user to restart the Handheld application after 30 visits, thereby freeing application memory.</p> |

Enabling Back Up of Siebel Handheld Application Database

This task is a step in [“Process of Installing the Siebel Handheld Application”](#) on page 175.

The Siebel Handheld application framework provides a way to automatically back up a snapshot of a user’s Siebel Handheld application database and current settings, and to restore those settings if the handheld device fails. The backup functionality automatically copies all the necessary files to a single compressed file. Each time a user backs up, the backup process creates a new copy of the file and deletes older backup files. The backup process must close the Handheld application in order to copy the RDBMS and handheld configuration files. After the files are copied, it automatically restarts the Handheld application.

The backup feature must be enabled in the Handheld application. If the backup feature is enabled, a backup is created during every synchronization. This way, the user retains an up-to-date snapshot of the Siebel Handheld application database, and older backup versions are removed.

When the database backup feature is enabled, a Siebel Handheld application database backup occurs automatically at the end of each synchronization. This makes the overall time to synchronize somewhat longer. The size of the Siebel Handheld application database and the speed of the CompactFlash card also affect the time it takes to complete synchronization.

NOTE: Backup files must not be used as a replacement for synchronization.

Setting Up Siebel Handheld Application Database Backups

To back up the Siebel Handheld application database, you must do the following:

- 1 Provide users with an external memory card.
- 2 Install the Siebel Handheld application with the backup functionality enabled. See, [“Enabling Siebel Handheld Application Database Backup”](#) on page 180.
- 3 Instruct users to back up their Siebel Handheld application databases on a regular basis.

Backing Up to External Media

It is recommended that you back up to external media—for example, a CompactFlash card or the embedded backup storage that is included in many devices.

Enabling Siebel Handheld Application Database Backup

Before users can back up their Siebel Handheld application databases, you must first enable the Backup menu item on the Siebel Handheld application. This requires editing the setup.ini file to enable backup.

NOTE: Backup files must be kept on external media or built-in storage. Trying to set the location to the main memory, such as \My Documents, will not work.

The following is an example of the default settings for the backup parameters:

```
[Backup]
BackupLocation = \Storage Card\Siebel Backup
BackupEnabled = N
```

- BackupEnabled = [Y/N] The default is N (No). Change this setting to Y (Yes) to enable the Backup menu item and to trigger a backup to occur automatically after each synchronization.

If the product is installed with Backup = N, the backup menu item is disabled. You need to change the parameter, then reinstall the product in order to enable the menu item.

- BackupLocation = [Backup location] The default is an external storage card. Specify the location where you want the data backed up.

For more information about the setup.ini file, see [Chapter 8, "Installing the Siebel Handheld Application."](#) For more information about backing up Siebel Handheld application data, see ["Backing Up and Restoring Data with the Siebel Handheld Client"](#) on page 212

Restoring from a Backup

For information about restoring a Siebel Handheld application database from a backup, see ["Backing Up and Restoring Data with the Siebel Handheld Client"](#) on page 212.

Installing Print Templates in the Siebel Handheld Install Directory

If your Siebel Handheld application allows printing, you must include all print templates in the template subdirectory of the Handheld application Install directory. The print templates are language-specific, so copy the templates to the template subdirectory within the language directory of the language you are deploying (for example, . . . \ENU\Templ ate). If the template includes double byte characters, such as Japanese string, you must save the template files as Unicode (UTF16) files. For supported language codes, see Siebel System Requirements and Supported Platforms on Oracle Technology Network.

For more information about printing and print templates, see the following:

- ["Views with Associated Print Templates or Reports"](#) on page 34
- ["Configuring Printing from the Siebel Handheld Application"](#) on page 83
- ["Printing with the Siebel Handheld Application"](#) on page 210
- Appendix E, "Print Tagging Language"
- Appendix F, "Print Configuration Settings"

Setting Up Siebel Handheld Application Installation on a CompactFlash Card

The default, when installing Siebel Handheld applications, is to install both the Handheld application and the Siebel Handheld application database into random access memory (RAM) on the device. In most instances, there is sufficient available memory, and users are satisfied with the Handheld application performance. There are situations, however, where you might need to optimize storage on the device.

If you need to maximize main memory availability, you can install the Handheld application binary files on an external location such as a CompactFlash card. Installing the data on the CompactFlash card is not recommended because performance likely degrades.

You can specify, in the Handheld application's setup.ini file, where the Handheld application files and Siebel Handheld application database files are to be installed. By default, this is set to the handheld device. In the ApplicationLocation section of the setup.ini file, there are two parameters. The following shows the default settings for these parameters:

```
[ApplicationLocation]
ApplicationBinariesLocation = Device
ApplicationDataLocation = Device
```

ApplicationBinariesLocation = Device means that the default location for the binaries is the handheld device. The other value is CF (CompactFlash).

ApplicationDataLocation = Device means that the default location for data is the handheld device. It is recommended that you always install the data files on the handheld device.

After you have edited the setup.ini file, follow the instructions for installing the Siebel Handheld application.

Changing the SyncURL on Handheld Devices

For Direct Server Synchronization, SyncURL is used to locate the Web Server and the virtual directory so synchronization for the Siebel Handheld Client can communicate with synchronization for the Siebel Handheld Application Server. For example:

```
SyncURL = http://web server name or IP address/virtual directory name
```

You can reset the SyncURL by re-installing the Handheld application with the updated setup.ini file.

For Companion Synchronization set up, the SyncURL parameter identifies the configuration file that will be accessed on the Siebel Handheld Client for synchronization.

For example, the default configuration file for Siebel Pharma Handheld is named `phce.cfg`. For companion synchronization set up, the `SyncURL` value must be `http://xxxx/phce` where `xxxx` is any text string.

NOTE: PPC 2002 and PPC 2003 do not have a hosts file similar to the Windows Operation System (for example, `C:\WINNT\system32\drivers\etc\hosts.file`), where you can set up mappings for the IP address and host name.

Installing the Siebel Handheld Application from External Media

Users who synchronize in Direct Server Synchronization mode might not have a PC, and so cannot create a partnership and synchronize using Microsoft ActiveSync. You can install the Handheld application for each user through a standard partnership, or you can configure a stand-alone installer.

During the installation, the InstallShield Wizard screen prompts you by asking:

Would you like to deploy a CAB file to enable rapid installation for additional devices?

You can configure a stand-alone installer by selecting Yes.

You are prompted to select a language for the Handheld application and to specify a location for the CAB file. The installer creates the CAB file in the location you specify.

The CAB file is a self-extracting file that includes all the files needed to install the Siebel Handheld application on the handheld device. When you single-click the file, the CAB file installs the Handheld application.

You can create an image of the Handheld application installer that can be loaded onto external media such as a CompactFlash card. When you create a stand-alone installer platform, a Cabinet file (CAB) is created in a location that you specify. The CAB file name must be `siebel.pocketpc_arm.CAB`. You can copy the CAB file to a CompactFlash card, insert the card into the handheld device, and single-click on the file to install the Siebel Handheld application on the device.

NOTE: When installing the Handheld application on the handheld device, the user is asked if they want to install the Handheld application in the default directory. Regardless of the answer, the Handheld application is always installed in the directory specified by the `setup.ini` file. Therefore, the only way to change the directory the Handheld application is installed in is to modify the parameters in the `ApplicationLocation` section of the `setup.ini` file.

Distributing Siebel Handheld Application Patches

PatchAgent is the patch delivery mechanism for your Siebel Handheld applications. PatchAgent is a tool that automates the deployment of patches to handheld devices as part of the synchronization process. It allows administrators to distribute patches to users with minimal involvement by the users.

During synchronization the PatchAgent executable is called, and if it detects a Handheld application patch, it automatically installs the patch on the handheld device. In order for PatchAgent to detect and deploy the patch, the patch files must be created and placed in the directory where PatchAgent expects to find the files.

The following procedures are included in this topic:

- [“Deploying Patches to Handheld Applications with Direct Server Synchronization” on page 184](#)
- [“Deploying Handheld Application Patches with Companion Synchronization” on page 186](#)
- [“Creating Handheld Application Upgrade Kits for Companion Synchronization PDA” on page 188](#)
- [“Deploying Attachments to Handheld Applications with Direct Server Synchronization” on page 188](#)
- [“Deploying Handheld Application Attachments with Companion Synchronization” on page 190](#)
- [“Downloading Patches to the Handheld Device” on page 192](#)
- [“Using PatchAgent to Distribute New Print Templates to the Handheld Device” on page 193](#)

Deploying Patches to Handheld Applications with Direct Server Synchronization

This task is a step in [“Process of Installing the Siebel Handheld Application” on page 175](#).

In order to set up patch deployment for Direct Server Synchronization users, you need to patch the Siebel Server and patch one Siebel Handheld Client which serves as the “model” from which patch files are pulled. Then the patch files are then packaged and placed in the location on the Siebel Server where PatchAgent expects to find the files. The next time the user synchronizes with Siebel Server, the patch is installed onto their handheld device.

For more information about how the patch process works with synchronization, see [“Downloading Patches to the Handheld Device” on page 192](#).

To set up patch deployment

- 1 On Siebel Server, verify that the following directories exist. If they do not, create the directories.
 - C: \patch\AppBi nDi r
 - C: \patch\Wi nDi r

NOTE: You can specify a drive other than C:. However, the directory names must be specified as shown above.
- 2 Apply the patch to Siebel Server.
- 3 Apply the patch to one Siebel Handheld Client.
- 4 Examine the Handheld application directory (My Devi ce\Program Fi les\Sibel Handhel d) to determine which files to copy over to the Siebel Server directory (C: \patch\AppBi nDi r).

- 5 Examine the Handheld application directory (My Device\Windows) to determine which files to copy over to the Siebel Server directory (C:\patch\Windows).

To do [Step 4](#) and [Step 5](#) accurately, it is recommended that you use one of the following methods:

- First, take a snapshot of all files in the Siebel Handheld application directory before you install the new build. Then install the new Handheld application build, making a note of all files that are installed in the process.

By comparing the list of files in the Handheld application directory before installation of the new build with the list of files in the Handheld application directory after installation of the new build, you can determine which files are new and which files remain the same.

Copy all new files over to the Siebel Server directory.

- Take a snapshot of the siebel.inf file before you install the new build. The siebel.inf file contains a complete list of all files that are installed on the handheld device, and is created just before the installation of the Handheld application (before clicking the finish button to finish installation of your Siebel Handheld application, the %temp%\siebel hh\Files\siebel.inf file is created).

In the case of [Step 4](#), by comparing the list of files in the [Files.AppBinDir] section of siebel.inf with the list of files that are installed with the new build in the \Program Files\Siebel Handheld folder, you can determine which files are new and which files remain the same.

In the case of [Step 5](#), by comparing the list of files in the [Files.Windows] section of siebel.inf with the list of files that are installed with the new build in the \Windows folder, you can determine which files are new and which files remain the same.

Copy all new files over to the Siebel Server directory.

NOTE: In all cases, compare the file size and timestamp on the older build with the newer build to determine which files need to be patched.

- 6 From the DOS prompt, change directory to *Siebel Root\SWEApp\BIN*.
- 7 Execute the following command to stage the patch files and the patchlist.txt in the correct location on the Siebel Server:

```
C:\Siebel Root\SWEApp\BIN>createpatchlist /s c:\patch /t c:\Siebel Root\sweapp\cepatch
```

where

- /s – source directory. The directory where the patch files were moved.
- /t – target directory. The directory where PatchAgent checks for the patch files.

NOTE: You can specify a drive other than C:. However, the location of the source and target directories must be specified exactly as shown.

After the patch files have been moved and the patch list created, the following message appears: Patch List has been successfully created in *Siebel Root\SWEApp\cepatch* files transferred successfully.

The patch files are now ready to be deployed to Direct Server Synchronization users.

Deploying Handheld Application Patches with Companion Synchronization

This task is a step in [“Process of Installing the Siebel Handheld Application”](#) on page 175.

In order to set up patch deployment for Companion Synchronization users, you need to create a “model” of a patched Siebel Mobile Web Client and a patched Siebel Handheld Client. Then you pull the designated patch files and place them in the correct directory on the user’s Mobile Web Client. You can use Siebel Anywhere to create upgrade kits to automate this process of distributing patch files to your user’s Mobile Web Client. The next time the user uses Companion Synchronization to synchronize their handheld device with the Siebel Mobile Web Client, the patch is installed on the device.

For more information about how the patch process works with synchronization, see [“Downloading Patches to the Handheld Device”](#) on page 192.

To set up patch deployment

1 On Siebel Server, verify that the following directories exist. If they do not, create the directories.

- C: \patch\AppBi nDi r
- C: \patch\Wi nDi r

2 Apply the patch to Siebel Server.

3 Apply the patch to Siebel Mobile Web Client.

This Mobile Web Client serves as the model from which the patch files can be pulled.

4 Apply the patch to one Siebel Handheld Client.

This Siebel Handheld Client serves as the model from which the patch files can be pulled.

5 Examine the Handheld application directory (My Devi ce\Program Fi les\Sibel Handhel d) to determine which files to copy over to the Siebel Server directory (C: \patch\AppBi nDi r).

6 Examine the Handheld application directory (My Devi ce\Wi ndows) to determine which files to copy over to the Siebel Server directory (C: \patch\Wi nDi r).

To do [Step 5](#) and [Step 6](#) accurately, it is recommended that you use one of the following methods:

- First, take a snapshot of all files in the Siebel Handheld application directory before you install the new build. Then install the new Handheld application build, making a note of all files that are installed in the process.

By comparing the list of files in the Handheld application directory before installation of the new build with the list of files in the Handheld application directory after installation of the new build, you can determine which files are new and which files remain the same.

Copy all new files over to the Siebel Server directory.

- Take a snapshot of the siebel.inf file before you install the new build. The siebel.inf file contains a complete list of all files that are installed on the handheld device, and is created just before the installation of the Handheld application (before clicking the finish button to finish installation of your Siebel Handheld application, the %temp%\siebel hh\files\siebel.inf file is created).

In the case of [Step 5](#), by comparing the list of files in the [Files.AppBinary] section of siebel.inf with the list of files that are installed with the new build in the \Program Files\Siebel Handheld folder, you can determine which files are new and which files remain the same.

In the case of [Step 6](#), by comparing the list of files in the [Files.Windows] section of siebel.inf with the list of files that are installed with the new build in the \Windows folder, you can determine which files are new and which files remain the same.

Copy all new files over to the Siebel Server directory.

NOTE: In all cases, compare the file size and timestamp on the older build with the newer build to determine which files need to be patched.

- 7 From the DOS prompt, change directory to C:\Siebel Root\SWEApp\BIN.
- 8 Execute the following command to move the patch files to the correct location and generate a patchlist.txt file:

```
C:\Siebel Root\SWEApp\BIN>createpatchlist /s c:\patch /t c:\Siebel Root\sweapp\cepatch_xxx
```

where

- /s – source directory. The directory where the patch files were moved.
- /t – target directory. The directory where PatchAgent checks for the patch files.
- xxx is the language [enu, esn, and so on].

NOTE: You can specify a drive other than C:. However, the location of the source and target directories must be specified exactly as shown.

After the patch files have been moved and the patch list created, the following message appears: Patch List has been successfully created in Siebel Root\SWEApp\cepatch_xxx files transferred successfully.

- 9 Create the following directory on the user's Mobile Web Client: Siebel Root\client\cepatch_xxx (where xxx is the language [enu, esn, and so on]), and push the files from the \Siebel Root\SWEApp\cepatch_xxx directory on Siebel Server to the Siebel Root\client\cepatch_xxx on the Mobile Web Client.

NOTE: Siebel Anywhere upgrade kits can be used to carry out this step.

Creating Handheld Application Upgrade Kits for Companion Synchronization PDA

This task is a step in [“Process of Installing the Siebel Handheld Application”](#) on page 175.

NOTE: This step is optional. Additionally, you must have a license for Siebel Anywhere to distribute patches to users.

You can use Siebel Anywhere upgrade kits to remotely distribute self-extracting zip files to your users' PCs. You can use self-extracting zip files to create new directories and immediately extract files into the directories. Create one upgrade kit for each of the following:

- Create the Siebel Mobile Web Client upgrade kit (Siebel Client Revisions).

Navigate to the Siebel image location for the current software version such as `D: / Siebel_Install_Image/7.8.1.0`, and then navigate to the patch directory. Configure the zip file to self-extract in the `Siebel Root\client\BIN` directory on the user's Mobile Web Client.

- Create the Siebel Handheld application upgrade kit (Siebel Client Revisions).

Create a self-extracting zip file that includes the files in the `Siebel Root\SWEApp\cepatch` directory on the server. Configure the zip file to self-extract in the `\client\cepatch` directory on the user's Mobile Web Client.

- Create any other upgrade kits as needed (for example, the Siebel Customer Repository File).

See *Siebel Anywhere Administration Guide* for detailed information on creating upgrade kits.

NOTE: If there are read-only files on the Mobile Web Client, the user might need to change the file attributes to read-write before receiving and installing the upgrade kits. For help with a list of files whose attributes need to be changed, create a service request (SR) on My Oracle Support. You can use Siebel Anywhere to distribute a batch script that makes this change.

Deploying Attachments to Handheld Applications with Direct Server Synchronization

This task is a step in [“Process of Installing the Siebel Handheld Application”](#) on page 175.

You deploy attachments to the Siebel Handheld application with Direct Server Synchronization in much the same way you deploy patches. PatchAgent automates the deployment of attachments to handheld devices as part of the synchronization process. For more information about how the patch process works with synchronization, see [“Downloading Patches to the Handheld Device”](#) on page 192.

You can specify the storage location for attachment files, and set a limit on the storage capacity to be made available for attachments on a handheld device. For more information about storage location and capacity, see [“Configuring the Attachment Storage Location and Storage Capacity”](#) on page 190.

Setting Up Your Attachment Files

In order to set up attachments for Direct Server Synchronization users and subsequently download them to a handheld device, you must first set up the attachment files on the Siebel Server that requires a patch.

To set up attachment files on the Siebel Server that requires a patch

1 Set up your patch deployment as already described in steps 1-6 of “Deploying Patches to Handheld Applications with Direct Server Synchronization” on page 184.

2 Execute the following command to stage the patch files and the patchlist.txt in the correct location on the Siebel Server:

```
C: \Siebel Root\SWEApp\BIN>createpatchlist /s C: \AttachmentDir\UserResponsibilityX
/t C: \Siebel Root\sweapp\cepatch_xxx\UserResponsibilityX
```

where

- /s – source directory. The directory where the patch files are located. Put all attachment files for a certain responsibility (for example, *UserResponsibilityX*) here.
- /t – target directory. The directory where PatchAgent checks for the patch files.
- xxx is the language [enu, esn, and so on]

Repeat for other responsibilities, as required:

```
C: \Siebel Root\SWEApp\BIN>createpatchlist /s C: \AttachmentDir\UserResponsibilityY
/t C: \Siebel Root\sweapp\cepatch_xxx\UserResponsibilityY
```

After the patch files have been moved and the patch list created, the following message appears:
Patch List has been successfully created in
Siebel Root\SWEApp\cepatch_xxx\UserResponsibilityX files transferred successfully.

NOTE: The name of the user responsibility directory corresponds to the actual user responsibility (for example, *Sales Rep*).

3 Copy the patchlist.txt and all the compressed patch files to the appropriate subdirectory, named according to the responsibility, under the CEPatch directory on the Siebel Web Server:

```
\SWEApp\Public\enu\CEPatch_xxx\Route Sales - HH CE\patchlist.txt
```

The CEPatch directory on the Web Server will look similar to the following:

```
\SWEApp\Public\enu\CEPatch_xxx\patchlist.txt
    \file1
    \file2
    \...
    \UserResponsibilityX|
    \UserResponsibilityX\patchlist.txt
    \UserResponsibilityX\AttachmentFile1
    \UserResponsibilityX\AttachmentFile2
    \UserResponsibilityX\...
    \UserResponsibilityY|
    \UserResponsibilityY\patchlist.txt
    \UserResponsibilityY\AttachmentFileA
```

```
\UserResponsibility\AttachmentFileB  
\UserResponsibility\  
\...
```

The patch files are now ready to be downloaded to Direct Server Synchronization users. For more information about downloading patch files, see [“Downloading Patches to the Handheld Device” on page 192](#).

Configuring the Attachment Storage Location and Storage Capacity

After the initial installation of your Siebel Handheld application, you can change the storage location for your attachment files, and set a limit on the storage capacity to be made available for attachments on your handheld device.

To change the storage location for attachment files, add the following line to the RegKeyList section of patchlist.txt:

```
HKLM, %reg_path%, "AttachmentDir", 0x00000000, DIRECTORY
```

where

DIRECTORY is the new attachment folder you want to use.

To set a size limit on the storage capacity, add the following line to the RegKeyList section of patchlist.txt:

```
HKLM, %reg_path%, "MaxAttachmentStorage", 0x00000000, SIZE
```

where

SIZE is a positive integer in bytes of storage (for example, 2MB=2,097,152 bytes).

Deploying Handheld Application Attachments with Companion Synchronization

This task is a step in [“Process of Installing the Siebel Handheld Application” on page 175](#).

You deploy attachments to the Siebel Handheld Application with Companion Synchronization in much the same way you deploy patches. PatchAgent automates the deployment of attachments to handheld devices as part of the synchronization process. For more information about how the patch process works with synchronization, see [“Downloading Patches to the Handheld Device” on page 192](#).

You can specify the storage location for attachment files, and set a limit on the storage capacity to be made available for attachments on a handheld device. For more information about storage location and capacity, see [“Configuring the Attachment Storage Location and Storage Capacity” on page 190](#).

Setting Up Your Attachment Files

In order to set up attachments for Companion Synchronization users and subsequently download them to a handheld device, you must first set up the attachment files on the server that requires a patch.

To set up attachment files on the server that requires a patch

- 1 Set up your patch deployment as already described in steps 1-7 of “Deploying Handheld Application Patches with Companion Synchronization” on page 186.
- 2 Execute the following command to stage the patch files and the patchlist.txt in the correct location on the server:

```
C: \Siebel Root\SWEApp\BIN>createpatchlist /s c: \AttachmentDir\UserResponsibilityX
/t c: \Siebel Root\sweapp\cepatch_XXX\UserResponsibilityY
```

where

- /s – source directory. The directory where the attachment files were located. Put all attachment files for a certain responsibility (for example, *UserResponsibilityX*) here.
- /t – target directory. The directory where PatchAgent checks for the patch files.
- xxx is the language [enu, esn, and so on].

Repeat for other responsibilities, as required:

```
C: \Siebel Root\SWEApp\BIN>createpatchlist /s c: \AttachmentDir\UserResponsibilityY
/t c: \Siebel Root\sweapp\cepatch_XXX\UserResponsibilityX
```

After the patch files have been moved and the patch list created, the following message appears: Patch List has been successfully created in *Siebel Root\SWEApp\cepatch\UserResponsibilityX* files transferred successfully.

NOTE: The name of the user responsibility directory corresponds to the actual user responsibility (for example, *Pharma Sales Rep*).

- 3 Copy the patchlist.txt and all compressed patch files to the appropriate subdirectory, named according to the responsibility, under the CEPatch directory on the laptop:

```
\Client\CEPatch_XXX\Route Sales - HH CE\patchlist.txt
```

If using Companion Synchronization, use Siebel Anywhere to push the patch files to the laptop. The CEPatch directory on the Web Server will look similar to the following:

```
\SWEApp\Public\enu\CEPatch_XXX\patchlist.txt
    \file1
    \file2
    \...
    \UserResponsibilityX|
    \UserResponsibilityX\patchlist.txt
    \UserResponsibilityX\AttachmentFile1
    \UserResponsibilityX\AttachmentFile2
    \UserResponsibilityX\...
    \UserResponsibilityY|
    \UserResponsibilityY\patchlist.txt
```

```
\UserResponsibility\AttachmentFileA  
\UserResponsibility\AttachmentFileB  
\UserResponsibility\...  
\...
```

The patch files are now ready to be downloaded to Companion Synchronization users. For more information about patch download, see [“Downloading Patches to the Handheld Device” on page 192](#).

Downloading Patches to the Handheld Device

This task is a step in [“Process of Installing the Siebel Handheld Application” on page 175](#).

Patches are downloaded as part of the synchronization process. When the patch is staged, the patch is downloaded the next time the user synchronizes the handheld device with Siebel Server. The users synchronize as they normally do using Direct Server Synchronization or Companion Synchronization. First, any transactions are uploaded from the handheld device. Then:

- If a patch needs to be applied to the handheld device, the synchronization process terminates, and the patch process is started, which downloads the patch.
- If no patch needs to be applied, the synchronization process continues until the process is complete.

After a successful patch installation, synchronization for the Siebel Handheld application launches automatically and continues the extraction.

If the patch files are not successfully downloaded, the user is prompted to restart the patch process. The user must locate and launch patchagent.exe manually. Then, once the patch is successfully installed, the user is prompted to complete the synchronization.

Setting up PatchAgent on Windows 2003

This task is a step in [“Process of Installing the Siebel Handheld Application” on page 175](#). When Siebel Web Server Extension (SWSE) software is installed on Windows 2003, you must complete the steps outlined in [“To set up PatchAgent on Windows 2003” on page 192](#) to allow PatchAgent to run without error on Siebel Handheld applications.

To set up PatchAgent on Windows 2003

- 1 Make sure that the Internet Guest Account (MACHINE_NAME\USR_MACHINE_NAME) exists in the system. This is automatically created after you install Windows 2003 Server.
- 2 Right-click on the CEPATCH folder and select Properties.
 - a Select the Security tab and add the following account:
Internet Guest Account MACHINE_NAME\USR_MACHINE_NAME
 - b Set the permission to Yes for all rights, except for Full Control.

- 3 Open Internet Information Services (IIS) console and navigate to the Properties tab on the IIS level.

Select the MIME types tab (go to the MIME types section) and add the following two properties to create a new MIME type:

| Property | Comment |
|--------------------------|--|
| .* | The extension field (. * is a period followed by an asterisk). |
| application/octet-stream | The MIME type field. |

Only add these properties if using IIS 5.0 or 5.5. If using IIS 6.0, then add the following properties instead:

| Property | Comment |
|-------------|--|
| .* | The extension field (. * is a period followed by an asterisk). |
| application | The MIME type field. |

This step is required to enable PatchAgent to access all files at the file level.

- 4 Using IIS console, right-click the cepatch_enu virtual folder and select the Properties tab.
 - a Go to the Security tab, and click the Edit... button.
The authentication and access control dialog box displays.
 - b Enter the details for the user account (for example, IUSR_MACHINE_NAME).
 - c Enter the password for the account.
 - d Select the Enable Anonymous Access check box.
- 5 Restart IIS.

NOTE: On a standard Windows 2003 installation, steps 1 and 4 might already be in place, but steps 2 and 3 must be setup by you.

Using PatchAgent to Distribute New Print Templates to the Handheld Device

New print templates can be distributed as a patch to your Siebel Handheld Application users. Patches are downloaded to handheld devices as part of the synchronization process.

Complete the steps in the following procedure to distribute new print templates to a handheld device.

To distribute new print templates using PatchAgent to a handheld device

Bundle your print templates into a patch, using dsspatchagent.exe. When handheld application users then synchronize using Direct Server Synchronization or Companion Synchronization:

- The handheld device checks for the presence of patches in <Siebel Root>\sweapp\cepatch.
- The handheld device uploads and processes transactions as described in [“Downloading Patches to the Handheld Device”](#) on page 192.

For more information about deploying patches to Handheld Applications, see:

- [“Deploying Patches to Handheld Applications with Direct Server Synchronization”](#) on page 184

7

Synchronization Conflict Handling and Recovery with Siebel Handheld Applications

The purpose of this chapter is to outline how to handle synchronization conflicts and recovery for Siebel Handheld applications. The following topics are covered in this chapter:

- [“Preventing Synchronization Transaction Conflicts with Handheld Applications” on page 195](#)
- [“Recovering Error Data in Siebel Handheld Applications” on page 199](#)

Preventing Synchronization Transaction Conflicts with Handheld Applications

When multiple users synchronize with a remote database, transaction conflicts can occur which can result in data loss. The Extended Pick and Insert Failure functionality minimizes synchronization conflicts. If transaction conflicts are unavoidable, then the Synchronization Service for the Siebel Handheld application captures the transaction information so that the data can be recovered. The combination of these features eliminates the potential for data loss.

There are two features that minimize synchronization transaction conflicts—Extended Pick processing and Insert Failure processing:

- [“Extended Pick Processing in Siebel Handheld Applications” on page 195](#)
- [“Insert Failure Processing in Siebel Handheld Applications” on page 198](#)

Extended Pick Processing in Siebel Handheld Applications

The Handheld application is used to capture and record transactions entered by users. During synchronization, these transactions are processed on the server. A situation can arise in which a transaction fails when it is applied to the Siebel Server (through Direct Server Synchronization) or Siebel Mobile Web Client (through Companion Synchronization)—both of which are referred to as server. For example, a Handheld application user creates a professional call activity for one of their contacts. However, since they last synchronized with the server, the contact involved in this transaction was deleted from the server database. As a result, when the user tries to synchronize their handheld device, the server does not find the contact, and the call activity for the contact cannot be inserted into the server database. To generalize this example, any transaction that involves data selected through picklists or pick applets fails on the server if that pick data is not present at the time of synchronization.

With many users synchronizing with Siebel Server and modifying data at different times, this scenario is not uncommon.

Synchronization Conflict Handling and Recovery with Siebel Handheld Applications

■ Preventing Synchronization Transaction Conflicts with Handheld Applications

The following describes how a pick transaction is recorded on a handheld device and processed on the server during synchronization. Assume that the user wants to create a new call activity for the contact. In order to do this, the user inserts a new record in the Professional Call screen and picks the desired contact. When the contact is selected on the handheld device, the Row Id of the contact (Contact Id) is recorded. However, on the handheld device, the application does not record any of the pick map fields associated with the contact; only the Contact Id is recorded. When the user synchronizes their handheld device, that Contact Id is used to locate the contact record in the server database. The Contact Id is then used to retrieve all of the fields used in the pick map for the Contact Id (Last Name, First Name, Middle Initial, Address, City, State, Zip Code, and so on).

In order to minimize pick processing failures, the Handheld application must record the pick row ID and all of the pick map data that is required for a successful transaction. This data must be recorded at the time of the pick, so that it can be passed to the server for processing. This enhanced functionality is called Extended Pick processing. Now, when the user synchronizes, the transaction is processed on the server as follows:

- The synchronization process first looks for the pick record on the server database. For example, to add an activity it looks for the contact and, if it finds the contact, it retrieves the necessary data and adds the activity. This is the default behavior.
- If the server does not find the pick map record in the database and extended pick processing is enabled, it retrieves the pick record ID and pick map data from the Handheld application transaction and applies the data directly to the server.

Enabling Extended Pick Processing in Siebel Handheld Applications

You can enable Extended Pick processing globally or at the business component level when configuring the Siebel Handheld application. When the *DefaultHandheldSyncPickMode* parameter is enabled, every pick transaction is subject to pick processing.

To enable extended pick processing globally

- 1 Navigate to the Administration - Mobile screen, then the Application Administration view.
- 2 Click the Settings tab, then Query on the following value:

DefaultHandheldSyncPickMode

If there is no such record, a new record with that value must be created. See [Table 28](#) for the values to enter.

Table 28. Application Settings

| Setting Name | Value | Description |
|-----------------------------|-------------|---|
| DefaultHandheldSyncPickMode | NoFail | Turns Extended Pick processing ON. |
| | FailOnError | Turns Extended Pick processing OFF This is the default behavior if the parameter is not specified in the Settings view. |

To enable extended pick processing at the business component level

- When enabling extended pick processing at the business component level, the user property is defined on specific business components. Use the HandheldSyncPickMode user property to set the extended pick processing mode for the fields you require. For more information about this user property, see [HandheldSyncPickMode User Property on page 314](#).

Table 29 shows the combined effect of the settings in the Application - Mobile, then the Settings View and on a business component.

- The DefaultHandheldSyncPickMode setting is omitted from application settings.
- If there is no setting on a particular business component, the setting defined in the Settings view applies.
- If there is no setting defined in the Settings view, or on the business component, the default behavior is for Extended Pick to be turned OFF.

In Table 29, read down the Tools Configuration column (which shows what HandheldSyncPickMode is equal to) and across to the Mobile Administration View Setting column (which shows what DefaultHandheldSyncPickMode is equal to) to see the combined result. For example, a NoFail configuration in Tools combined with a FailOnError setting in the Mobile Administration View results in the combined effect NoFail.

Table 29. Combined Effect of Extended Pick Settings for Business Components

| Tools Configuration (User Property) | Mobile Administration View Settings | | |
|-------------------------------------|-------------------------------------|-------------|-------------|
| | NoFail | FailOnError | Omitted |
| NoFail | NoFail | NoFail | NoFail |
| FailOnError | FailOnError | FailOnError | FailOnError |
| Omitted | NoFail | FailOnError | FailOnError |

Recommended Extended Pick Processing Settings

It is recommended that you use the Extended Pick processing settings in the following way:

- **Pharma Professional Call - CE.** Last Name|NoFail, Address|NoFail, Contact Id|NoFail, Contact Address Id|NoFail
- **Pharma Call Products Detailed - CE.** Name|NoFail
- **Pharma Call Sample Dropped - CE.** Name|NoFail, Lot Name|NoFail
- **Pharma Account Call - CE.** Account Name|NoFail, Address|NoFail, Account Id|NoFail, Account Address Id|NoFail
- **Pharma Promotional Items Dropped - CE (Name).** Name|NoFail

The Pharma Handheld application is shipped with these recommended settings.

Insert Failure Processing in Siebel Handheld Applications

When the server executes an insert transaction and the insert fails, there are two ways the insert failure can be handled:

- A record is logged that indicates the insert failed, and synchronization continues. This is the default insert behavior. The Conflict Administration screen can be used to recover the data.
- The insert failure is logged, including details of the failure, and synchronization is interrupted. This is the FailOnError behavior. FailOnError behavior guarantees that, if an insert fails, data loss does not occur since synchronization is stopped until the circumstances causing the failure are resolved.

NOTE: If FailOnError processing takes place, synchronization is interrupted. The synchronization process continues to fail until the problem is resolved. Users must contact their system administrators to recover from the failure. The recovery effort can involve modifying transactions queued for processing, modifying data on the server, or reentering transactions through the Siebel Handheld application.

Handling Insert Failures

You can enable insert failure processing globally or at the business component level when you configure the Siebel Handheld application. The following setting applies globally throughout the application. When FailOnError is specified, every failed insert interrupts synchronization.

DefaultHandheldInsertMode

You can specify the value for this setting through the Settings view in the Administration-Mobile screen.

Table 30 shows the settings for this parameter.

Table 30. Insert Failure Processing Parameter Setting

| Parameter | Value | Description |
|---------------------------|-------------|--|
| DefaultHandheldInsertMode | NoFail | Insert failures are logged and synchronization continues. This is the default behavior. |
| | FailOnError | Insert failures are logged and synchronization terminates. The system administrator must intercede to recover from this failure. |

You can also use the HandheldInsertFailMode user property to enable insert failure processing at the business component level. For more information about this user property, see [HandheldInsertFailMode User Property on page 314](#).

Table 31 shows the combined effect of the setting in the Settings view and the setting on a business component.

- The business component setting always takes precedence over the settings defined in the Administration-Mobile, then the Settings view.
- If there is no setting on a particular business component, the Applications Settings value applies.
- If there is no setting in either the Settings view or on the business component, the default behavior is the same as NoFail. Insert failures are logged and synchronization continues.

In [Table 31](#), read down the Tools Configuration column (which shows what HandheldInsertFailMode is equal to) and across to the Application Settings Value (which shows what the DefaultHandheldInsertMode is equal to) to see the combined result. For example, a NoFail BC setting combined with a FailOnError application setting results in the combined effect NoFail.

Table 31. Combined Effect of Insert Failure Settings

| Tools Configuration (User Property) | Application Settings Value | | |
|-------------------------------------|----------------------------|-------------|-------------|
| | NoFail | FailOnError | Omitted |
| NoFail | NoFail | NoFail | NoFail |
| FailOnError | FailOnError | FailOnError | FailOnError |
| No Setting on Business Component | NoFail | FailOnError | NoFail |

Recommended Insert Failure Processing Settings

It is recommended that you use the insert failure processing settings in the following way:

- Set Application Settings to FailOnError during development and testing to catch synchronization errors.

Recovering Error Data in Siebel Handheld Applications

Selected synchronization and conflict errors are captured to the database, and can be accessed through the Conflict Administration screens. This mechanism is always enabled. For more information about recovering error data, see [“Obtaining Error Details from Synchronization Sessions in Siebel Handheld Applications”](#) on page 136.

8

Installing the Siebel Handheld Application

This chapter guides you through the process of installing and upgrading a Siebel Handheld application. The following topics are covered in this chapter:

- [“Process of Installing, Reinstalling, or Upgrading a Siebel Handheld Application” on page 201](#)
- [“Uninstalling the Siebel Handheld Application” on page 201](#)
- [“Installing the Siebel Handheld Application” on page 202](#)

Process of Installing, Reinstalling, or Upgrading a Siebel Handheld Application

The Siebel Handheld application cannot be upgraded directly on the handheld device. You must uninstall the old Siebel Handheld application and then install the new one. To reinstall or upgrade your existing Siebel Handheld application and migrate its data and settings to the current release, or to install a new Siebel Handheld application, use the following process:

- 1 If you have a Siebel Handheld application installed, before you reinstall or upgrade the application, back up its data by synchronizing your application with the Siebel Handheld application database. See [“Synchronizing Data on Your Handheld Device” on page 214](#).
- 2 If you have a Siebel Handheld application installed, uninstall the Siebel Handheld application. See [“Uninstalling the Siebel Handheld Application” on page 201](#).
- 3 Install the current version of your Siebel Handheld application. See [“Installing the Siebel Handheld Application” on page 202](#).
- 4 Synchronize your Siebel Handheld application data. See [“Synchronizing Data on Your Handheld Device” on page 214](#).

Uninstalling the Siebel Handheld Application

Uninstalling the Siebel Handheld application deletes the database files as well as the application files.

This task is a step in [“Process of Installing, Reinstalling, or Upgrading a Siebel Handheld Application” on page 201](#).

CAUTION: Before uninstalling your Siebel Handheld application, you must synchronize the Siebel Handheld application database to save any changes to your data since the last synchronization. If you do not synchronize before uninstalling, these changes will be lost.

To uninstall the Siebel Handheld application

- 1 Synchronize your Siebel Handheld application database. See [“Synchronizing Data on Your Handheld Device” on page 214.](#)
- 2 Select Start, then Settings.
- 3 Select the System tab, and tap Remove Programs.
- 4 From Remove Programs, select the Siebel Handheld application, and tap Remove.

A dialog box appears indicating that a file cannot be deleted. This occurs because a file might still reside in memory.

- 5 Ignore the message, and tap OK to delete all the Siebel Handheld application and database files.

Installing the Siebel Handheld Application

There are two ways to install software onto any Windows-powered handheld device. The first is to create a partnership between a desktop or laptop PC and the handheld device using Microsoft ActiveSync. The second is to install the Siebel Handheld application from some type of external media.

The steps in this procedure apply to both companion synchronization and Direct Server Synchronization. For more information about Direct Server Synchronization and setting filters, see [Chapter 5, “Data Filtering for Siebel Handheld Applications.”](#)

NOTE: Often, a Direct Server Synchronization user does not have access to a PC for creating a partnership with her or his handheld device. In such cases, an administrator must create a temporary partnership with each user's device and install the Siebel Handheld application. You can install it in Direct Server Synchronization mode with the Siebel Installer as described in this topic.

This task is a step in [“Process of Installing, Reinstalling, or Upgrading a Siebel Handheld Application” on page 201.](#)

To install the Siebel Handheld application

- 1 Using Microsoft ActiveSync, establish a partnership between the handheld device and the PC.
- 2 On the PC network or DVD-ROM, locate the folder where the Siebel Handheld Application folder is installed, then double-click setup.exe.

The installation process begins.

- 3 Follow the directions that appear on screen to complete the installation.

- 4 After installing the Siebel Handheld application, and *before* using it, perform an initial synchronization. If you are using Companion Synchronization to synchronize your data, synchronize the Siebel Handheld application database with the Siebel Remote database. See [“Synchronizing Data on Your Handheld Device” on page 214](#).

NOTE: Before using the Siebel Handheld application, you must synchronize the handheld device to download the Siebel Handheld application configuration and populate the Siebel Handheld application database. Therefore, an administrator can install the Siebel Handheld application for a user, but the synchronization must occur while a user is logged in to the Siebel Handheld application with a valid username and password.

9

Working with Siebel Handheld Applications

The purpose of this chapter is to introduce the Siebel Handheld application interface by describing the components of the interface and how to navigate it, and introducing some useful tasks, such as, entering data, exporting data, synchronizing data, and printing. The following topics are covered in this chapter:

- [“Components of the Siebel Handheld Application Interface” on page 205](#)
- [“Navigating the Siebel Handheld Application Interface” on page 207](#)
- [“Navigating a List of Records in the Siebel Handheld Application” on page 208](#)
- [“Entering Data with the Siebel Handheld Application” on page 209](#)
- [“Finding and Querying Data in the Siebel Handheld Application” on page 210](#)
- [“Printing with the Siebel Handheld Application” on page 210](#)
- [“Exporting Data with the Siebel Handheld Application” on page 211](#)
- [“Setting User Preferences in the Siebel Handheld Application” on page 212](#)
- [“Backing Up and Restoring Data with the Siebel Handheld Client” on page 212](#)
- [“Synchronizing Data on Your Handheld Device” on page 214](#)
- [“About Synchronization Options” on page 218](#)
- [“Troubleshooting Synchronization with the Siebel Handheld Application Client” on page 219](#)
- [“Guidelines for Improving Siebel Handheld Application Performance” on page 220](#)

Components of the Siebel Handheld Application Interface

The components of the Siebel Handheld application interface include the Screens drop-down list, Show drop-down list, application-level menu, Queries drop-down list, toolbar, and status bar.

Figure 8 shows the components of the Siebel Handheld application.

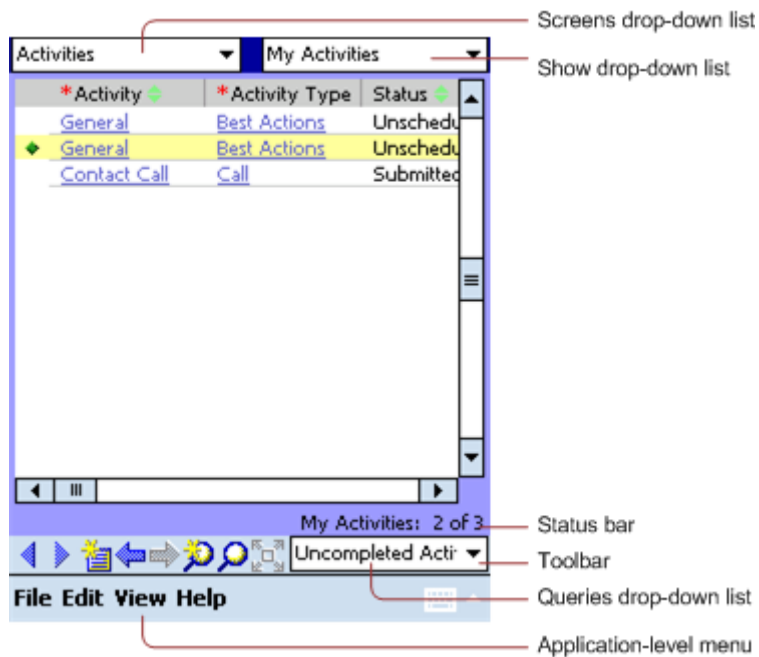


Figure 8. Siebel Handheld Application Screen Components

Screens Drop-Down List

The Screens drop-down list is the first level of navigation in the Siebel Handheld application. When you launch the Siebel Handheld application for the first time, this drop-down list is labeled Screens unless you have configured a start up view to appear. When you tap the Screens drop-down list, a drop-down list of all available screens appears. Tap the desired screen in the list to navigate to it.

For information about configuring a start up view, see [“Setting User Preferences in the Siebel Handheld Application”](#) on page 212.

Show Drop-Down List

The Show drop-down list is the second level of navigation in the Siebel Handheld application. This is where you choose views for the selected screen. When you launch the Siebel Handheld application for the first time, this drop-down list is labeled Views.

Application-Level Menu

The application-level menu consists of the File, Edit, View, and Help menus. Tap a menu to select a menu option. The options that are available in each menu vary depending on the task you are performing.

Queries Drop-Down List

Up to two predefined queries, or business object filters are displayed in the Queries drop-down list. The list displays the query chosen when you synchronized (or the default filter if you did not change the Set Filters selection) and the All Records query, if it is defined. For more information about the queries drop-down list, see [Chapter 5, "Data Filtering for Siebel Handheld Applications."](#)

Toolbar

The toolbar on the Siebel Handheld application has buttons for frequently used functionality like record navigation and querying. You can configure your toolbar and specify which buttons are displayed. For more information about the toolbar on your Siebel Handheld application, see the Siebel Handheld application online help.

Status Bar

The status bar appears at the bottom of the application window. It provides information about the current view.

Navigating the Siebel Handheld Application Interface

Navigating the Siebel Handheld application is achieved through the Screens drop-down list, the Show drop-down list, drilling down, drilling across, toolbar buttons, and toggling.

Screens Drop-Down List

The Screens drop-down list provides the first level of navigation. Here you select the screen you want to work with. For more information about the Screens drop-down list, see ["Components of the Siebel Handheld Application Interface" on page 205.](#)

Show Drop-Down List

The Show drop-down list is the second level of navigation. After you have selected a screen from the Screens drop-down list, select a view from the Show drop-down list. Views in the Show drop-down list are specific to each screen.

For example, in the Activities screen you can pick the My Activities view from the Show drop-down list.

Navigating between Applets

Two or more applets can be displayed at any one time in a Siebel Handheld application. Navigating between applets depends on whether your application is configured to display in traditional two-applet mode or tabbed applet mode.

For information about how to navigate between applets in either mode, see the online help for the Siebel Handheld application.

Expandable Fields

Due to the limited screen size on the handheld device, it is not always easy to view the data in a field without scrolling or resizing the field width. Therefore, some fields that can contain large amounts of text are expandable—that is, the data can be viewed in a text box. An ellipsis (...) appears in the field to indicate that the field is expandable. When you tap the ellipsis, a text box opens showing the data in that field. When you step off the field, the text box closes.

In form applets, an ellipsis is always visible in those fields that are expandable. However, in list applets, due to space constraints, the ellipsis is not always visible in expandable fields. When you tap on an expandable field in a list applet, the ellipsis appears. Once the ellipsis appears, you tap the ellipsis to open the text box.

URLs in Applet Fields

Tapping on a field that contains a URL value launches the URL in PocketIE.

Navigating a List of Records in the Siebel Handheld Application

Within the view you select from the Show drop-down list, you can view detailed information on a record. When records are displayed in a list, you can drill down or drill across the record to get additional details.

Drilling down occurs when you tap a hyperlink in a record and are taken to another view within the current screen. For example, if you are in the Accounts screen, and tap the Account Name hyperlink, you navigate to the details for that account. You do not leave the Accounts screen; you just move deeper into it.

NOTE: Hyperlinks are indicated by blue underlined text as long as they are set up this way in User Preferences. For more information about user preferences, see [“Setting User Preferences in the Siebel Handheld Application” on page 212](#).

Drilling across a record occurs when you tap a hyperlink in a record and are taken to another view in a different screen. For example, tapping a hyperlink in the Accounts column of a contact record in the Contacts screen takes you to the record for that account in the Accounts screen.

Dynamic drilldown occurs when you tap a hyperlink in a record and are taken to another screen or just another view. Where you navigate to depends on the content in the drilldown field. For example, tapping a hyperlink in the Activity field in the Activities screen can take you to different places. If you tap an Account Call activity, you navigate to the Account Call screen. If you tap a Visit activity, you navigate to the Activities More Info view.

History Arrows




To navigate back to views you have recently displayed, tap the Back button. After you have used the Back button, you can use the Forward button to navigate back to your original screen and view.

NOTE: The History arrows allow you to navigate only between views. These arrows do not allow you to recover data that might have been entered or deleted in another view.

Record Navigation Buttons

Use the Record Navigation buttons to move back and forth through a set of records. The navigation buttons are described in [Table 32](#).

Table 32. Record Navigation Buttons

| Button | Description |
|---|---|
|  | Navigates back to the previous record in the list or form. |
|  | Navigates forward to the next record in the list or form. |
|  | Navigates back a view based on the navigation history. You can define which views are part of the history. If a view is set to not be recorded as part of the history, the back button will not go back to that view. |

Entering Data with the Siebel Handheld Application

There are several different methods for entering data in the Siebel Handheld application. Tap the up arrow next to the Input Panel button to select a method. See the user guide for your device for information on using the different input methods.

Working with Columns in the Siebel Handheld Application Client

Data is displayed in lists and forms in your Siebel Handheld application. You can organize and work within it in a number of ways to better suit your needs.

There are several ways in which to organize columns in a list. You can:

- Sort data by up to three columns
- Resize columns
- Change the order in which the columns appear
- Show and hide columns
- Lock columns for horizontal scrolling

You can sort records in a list by tapping in the column header of the column in which you want to sort the records. If the green column sort indicator points up, the column is sorted in ascending order. If it points down, the column is sorted in descending order.

If you need to sort by more than one column, you can sort up to three columns at one time. For information about how to perform an advanced sort, see the online help for the Siebel Handheld application.

Resize columns by tapping and dragging the divider between the column headings until the column is the desired size.

You can also change the order of columns. For information about how to change the order of columns, see the online help for the Siebel Handheld application.

Show and hide columns using the Columns Displayed dialog box. In the Available columns list, select the items you want to show and tap the arrow to move the selected items to the Selected columns list. To hide columns, select the items and tap the arrow to move the selected items to the Available columns list.

Lock or unlock columns by tapping and holding the column header.

Finding and Querying Data in the Siebel Handheld Application

You can search for data using the Find or Query functionality in your Siebel Handheld application.

- [“About Find in Siebel Handheld Applications” on page 210](#)
- [“About Queries in Siebel Handheld Applications” on page 210](#)

About Find in Siebel Handheld Applications

You can use the Find feature to do a simple search for a record. For more information about how to use the Find feature, see the online help for the Siebel Handheld application.

About Queries in Siebel Handheld Applications

Use Query to locate a number of records containing a specific set of criteria. There are a number of ways in which to perform query tasks in your Siebel Handheld application.

For information about how to perform query tasks, see the online help for the Siebel Handheld application.

Printing with the Siebel Handheld Application

You can print from any view in the Siebel Handheld application that has been configured to support printing. When you print from a handheld device, a document that has been configured for the view is printed. You can print directly to a printer, or, if you are not connected to a printer at the time, you can queue your print jobs and wait until you have access to a printer to print your documents.

NOTE: When you print from a view, you are not necessarily printing what is visible on the screen at the time. Every field on the screen might not be relevant for the document. The document might contain only a portion of what is in the view, and it might also contain data that is not viewable at the time. A separate applet, which often cannot be viewed in the user interface, is configured specifically for printing.

The print templates used for generating and formatting print output are installed during setup. For a list of supported printers, see Siebel System Requirements and Supported Platforms on Oracle Technology Network.

For more information about printing and print templates, see the following:

- “Views with Associated Print Templates or Reports” on page 34
- “Configuring Printing from the Siebel Handheld Application” on page 83
- “Installing Print Templates in the Siebel Handheld Install Directory” on page 181
- Appendix E, “Print Tagging Language”
- Appendix F, “Print Configuration Settings”

For information about how to perform print tasks, see the online help for the Siebel Handheld application.

Exporting Data with the Siebel Handheld Application

You can export data from any view in your application in HTML format or as tab delimited text and save it to a file. This file can be viewed in applications such as Microsoft Pocket Word, Microsoft Pocket Excel, and Internet Explorer.

For information about how to export data, see the online help for the Siebel Handheld application.

In the Export dialog box, you can specify the following:

- **Format.** Specify either HTML or Tab delimited text file.
- **Export.** Select Only active applet to export only the data in the active applet. Select All applets in view to export the data in all visible applets.
- **Output file name.** By default, the file is saved in the temp directory with the filename “output.” You can tap Browse to specify a different location or filename.
- **Paste Output to Clipboard.** Select the check box to save the data to the clipboard.
- **Open file.** Select the check box to automatically open the file. When you tap OK in the Export dialog box, the data is exported to the specified file, and the file is automatically opened.

NOTE: You must specify a valid output format for the application. See Table 33 for the correct formats to specify for the supported applications.

Table 33. Export Data Options

| To export data to | Do the following |
|-------------------|---|
| Microsoft Word | Set the Format option to Tab delimited text file. |

Table 33. Export Data Options

| To export data to | Do the following |
|-----------------------------|--|
| Microsoft Excel | Select the <i>And paste output to clipboard</i> check box. Set the Output format to either HTML or Tab delimited text file. Paste the data from the clipboard into an Excel spreadsheet. |
| Microsoft Internet Explorer | Set the Format option to HTML. |

Setting User Preferences in the Siebel Handheld Application

You can set preferences that are related to the appearance and functionality of your Siebel Handheld application. You can do the following:

- Set a default startup screen.
- Set the maximum number of history threads to save.
- Set the maximum percentage of the display area that the parent applet can take up.
- Change font size.

NOTE: After Siebel 7.5.x, the default font size of record text in applets was increased by 1 point to facilitate user interaction with the handheld device.

- Set the grid lines.
- Set preferences for the calendar, for example, default view, start time, time slot interval, and default appointment duration.

For information about how to set the user preferences, see the online help for the Siebel Handheld application.

User preferences are stored within the following files (located on the handheld device in My Device, Program Files, Siebel Handheld directory), which you can distribute to all Siebel Handheld users to preserve a consistent look and feel for your Siebel Handheld application installation:

- Siebel.spf stores all user preferences (list appearance, behavior, and calendar preferences).
- Siebel.ssf stores all layout specifications (column visibility, and column width).

Backing Up and Restoring Data with the Siebel Handheld Client

It is recommended that you back up your data to an external device—for example, a CompactFlash card or a Secure ID card. If the handheld device is damaged, the backup might not be retrievable. If you store your backups in RAM, depending on the size of your application and data extract, you can decrease the amount of memory available for the application, and your application performance can suffer.

Automatic Backup after Synchronization

The automatic backup feature must be enabled in the application by your System Administrator. If the backup feature is enabled, a backup is automatically created at the end of every synchronization. This way, you have an up-to-date snapshot of the database.

CAUTION: You must never synchronize your data to the handheld device and then restore the database using an older copy of the database. If you do, the next time you synchronize, the application tries to rewrite changed transactions. This can cause data integrity problems. Therefore, when the database backup feature is enabled, a database backup occurs automatically at the end of each synchronization.

Keep in mind that enabling automatic backup makes the overall time to synchronize somewhat longer. The size of the database and the speed of the CompactFlash card also affect the time it takes to complete synchronization.

For more information about automatic backup, see [“Enabling Back Up of Siebel Handheld Application Database” on page 180](#).

Backing Up Data

You can back up the data in your Siebel Handheld application. The Siebel Handheld application configuration files and data are copied to the backup location specified in the setup.ini file. Typically, the backup location is a CF or SD card. For example:

```
BackupLocation = \CF Card\Siebel Backup
```

OR,

```
BackupLocation = \Storage Card\Siebel Backup
```

For more information about database backup, see [“Enabling Siebel Handheld Application Database Backup” on page 180](#). For more information about how to back up data, see the online help for the Siebel Handheld application.

Restoring Data from a Backup

In order to restore the application to its previous state, the user must contact the system administrator. Any work performed on the handheld device since the last backup will be lost.

To restore the database

- 1 Remove the external flash card from the damaged device and place it in a new handheld device where the application has been installed.
- 2 Navigate to the \Program Files\Siebel Handheld folder.
- 3 Start the restore by tapping the BackupUtility file twice.

A dialog box appears, asking you to confirm that you want to restore the device from a previous backup.

4 Tap Yes.

A status dialog box appears, indicating that the restore is in progress. The backup utility restores the files to the state when the backup was performed. A dialog box appears telling you if the restore was successful. If successful, it asks you to confirm that you want to restart the Siebel Handheld application.

5 Tap Yes to start using the application.

Synchronizing Data on Your Handheld Device

The tasks covered in this topic are part of the [“Process of Installing, Reinstalling, or Upgrading a Siebel Handheld Application”](#) on page 201.

You must synchronize the data on your handheld device:

- After you install or upgrade the Siebel Handheld application.
- Before uninstalling the Siebel Handheld application.
- On a regular basis to keep your data synchronized with Siebel Server and Siebel Mobile Web Client.

The topics that follow describe how to synchronize using Companion Synchronization (CS), Direct Server Synchronization (DSS) and Direct Server Synchronization via Proxy (DSSvP), and provide some troubleshooting tips:

- [“Using Synchronization with the Siebel Handheld Application Client”](#) on page 214
- [“Using Companion Synchronization with the Siebel Handheld Client”](#) on page 215
- [“Using Direct Server Synchronization with the Siebel Handheld Application Client”](#) on page 217

Using Synchronization with the Siebel Handheld Application Client

Synchronization in the Siebel Handheld application differs depending on whether users synchronize through Companion Synchronization, Direct Server Synchronization or Direct Server Synchronization via Proxy. The following topics show you how to launch and use synchronization in your Siebel Handheld application for each method.

NOTE: A given user can only synchronize using one method. For example, it is not possible to install a Siebel Handheld application that a particular user will synchronize both to the Siebel Mobile Web Client database through Companion Synchronization and also to the Siebel Server database directly through Direct Server Synchronization.

Using Companion Synchronization with the Siebel Handheld Client

If you are using Companion Synchronization, you must follow a three-step process to synchronize your Handheld application, remote, and server databases. Immediately before and after every Siebel Remote synchronization, you must perform a Companion Synchronization. This process can be time consuming for the user and difficult for the administrator to enforce.

Integrated Synchronization eliminates the need for you to initiate three separate synchronization sessions for each Siebel Remote synchronization.

Before using Integrated Synchronization, you must specify or add the parameter RemoteCfgName in the [pda] section of phce.cfg for the CFG file that will be used to invoke Siebel Remote synchronization. For example, RemoteCfgName=epharma.cfg. If you do not specify this parameter it will default to siebel.cfg. This parameter is used by Siebel Handheld application synchronization to invoke Siebel Remote. It applies only when synchronizing using the Integrated Synchronization option.

Integrated Synchronization automatically initiates each of the following synchronization steps:

- 1 Handheld Upload and Processing - Handheld application transactions are uploaded to the companion PC and processed.
- 2 Siebel Remote Synchronization - The Siebel Mobile Client is synchronized with the Siebel Server. Transactions on the Mobile Client database are uploaded to the Siebel Server, and changes to the Siebel Server are downloaded to the Mobile Client database.

CAUTION: If there is a Handheld application patch staged on the server, synchronization terminates and the patch is installed. After the patch installation is completed, you are prompted to synchronize again (using either Integrated Synchronization or Companion Synchronization) to complete the data extraction and download (that is, move to Step 3). It is critical that you synchronize a second time after the patch installation is completed. If you fail to do so, you will not be able to start the Siebel Handheld application.

- 3 Handheld Data Extract and Download - Data is extracted from the Mobile Client database to the handheld device database.

You can use Companion Synchronization to synchronize your Handheld application with Siebel Mobile Client as many times as you wish between Integrated Synchronization sessions. However, it is recommended that you conduct Integrated Synchronization sessions frequently so that the Siebel Remote synchronization process does not take long periods of time to complete and update conflicts are minimized.

To perform Companion Synchronization from the PC

- 1 From the companion PC, establish a Microsoft ActiveSync connection to the handheld device.
- 2 From the companion PC, choose Start, Programs, then Siebel Handheld Sync.

The Siebel Handheld Synchronization dialog box appears.

- a Enter your name and password.

- b** Select the Perform Remote Sync check box to perform an Integrated Synchronization.

If you want to perform a Companion Synchronization only, then unselect the Perform Remote Sync check box.

Selected Bypass Set Filters skips the step of selecting the filters and immediately starts the synchronization process. If you select Bypass Set Filters, then the filters that you selected the last time you synchronized are used. If this is the first time you are synchronize, then the default filters are used.

- 3** Tap Start.

NOTE: Tap Default Filters to restore the filter selection to the default setting.

- 4** Tap OK and tap Start to begin synchronization.

- a** The Siebel Handheld Sync dialog box appears. If "Perform Remote Sync" was selected, this is followed by the Siebel Remote dialog box. When synchronization has successfully completed, the Finished dialog box appears.

- b** If a Handheld application patch needs to be installed, the synchronization process is interrupted and the patch is installed. After the patch is installed, you must restart Siebel Handheld application synchronization to complete the data extract process. If you fail to do so, you cannot start your application.

- c** The Finished dialog box appears indicating that the synchronization has successfully completed.

- 5** Tap OK to exit Siebel Handheld Sync.

NOTE: If synchronization errors occur, an error dialog box appears.

You can also perform a companion synchronization by running hhsync.exe from the command line. This executable runs in silent mode without user input, It allows customers to launch Handheld Sync without entering a user name and password. This can be done by including the hhsync.exe in a batch file.

To perform Companion Synchronization from the command line

- 1** From the companion PC, establish a Microsoft ActiveSync connection to the handheld device.

- 2** Choose Start, Programs, Accessories, then Command Prompt

- 3** Enter the following command:

```
Run hhsync.exe /u sporter /p sporter /errlog d:\txn.log /remote Y /remote_mode F
```

where:

/u = <User name> (Required)

/p = <Password>

/errlog = <Transaction error file path/name>

/remote = <Perform remote sync. Accepted values: Y or N>

/remote_mode = <Remote sync mode. Accepted values: F or U. F: Full Sync and U: Upload Only.

Companion Synchronization can also be performed in the method described for Direct Server Synchronization via Proxy in [“Using Direct Server Synchronization with the Siebel Handheld Application Client”](#) on page 217.

To perform Companion Synchronization from the handheld device

- 1 Start the synchronization process from the application-level menu.
For information about how to start synchronization from the application-level menu, see the online help for the Siebel Handheld application.
- 2 In the Sync window, select the Enable Remote Sync check box. It is disabled by default.
For information on changing the parameter value for EnableRemote in the setup.ini file, see [“Siebel Handheld Application setup.ini Parameters”](#) on page 176.
- 3 See the steps for synchronizing using DSSvP in the procedure [“To synchronize with the Siebel Handheld Application Server”](#) on page 217.

Using Direct Server Synchronization with the Siebel Handheld Application Client

There are two methods for synchronizing between the handheld device and the Siebel Handheld Application Server:

- Direct Server Synchronization (DSS) provides synchronization between the Siebel Handheld application and the Siebel Handheld Application Server.
- Direct Server Synchronization via Proxy (DSSvP) provides synchronization between the Siebel Handheld application and the Siebel Handheld Application Server, using the desktop or companion PC to provide the network connection between the handheld device and the server.

To synchronize with the Siebel Handheld Application Server

- 1 If you are synchronizing using DSS, create a direct network connection through a modem or a direct LAN connection.
If you are synchronizing using DSSvP, establish an ActiveSync connection between the handheld device and the PC.
- 2 Start the synchronization process from the application-level menu.
For information about how to start synchronization from the application-level menu, see the online help for the Siebel Handheld application.
- 3 Enter your user name and password.
- 4 Tap Start.
The Siebel Handheld Sync dialog box appears.

- 5 Tap OK to exit Siebel Handheld Sync dialog, and the application launches automatically. If synchronization errors occur, an error dialog box appears.

NOTE: DSS works if the version of the Siebel Handheld application on your handheld device is slightly different to the version installed on your server. However, it is recommended that versions of the Siebel Handheld application on your device be the same or as close as possible to the Siebel Handheld application version on the server.

About Synchronization Options

From the Siebel Handheld Sync dialog, the following items are contained in the Options menu:

- **Filters:** If this is the first time you synchronize, then the default filters are used. Enables you to choose the screens that you want to filter. Choose the filter for each screen that you select. You can choose only one filter for each screen.
- **Change User:** This option enables a new user to use the handheld device when the original user no longer needs the device. The Siebel Handheld application does not need to be re-installed when a new user wants to access the application from the handheld device.

NOTE: If there are pending transactions for the current user, you have to synchronize them first before you can change to a different user name.

- **Save Password:** If you want to save your password so that you do not need to enter it each time you access the Siebel Handheld Sync dialog, select the Save Password check box.

When you select Save Password, the password is encrypted and saved on your hard drive. Once the password is saved, when you start the application you do not see the Siebel Handheld Sync login dialog. The encrypted password must be deleted in order for you to be able to access the login dialog box. To remove the encrypted password, delete "Password=<encrypted text>" under the [Parameters] section of the pimsync.cfg file.

Save password check box is enabled only if the Allow Save Password application setting is set to TRUE on the server. That is done using the Administration - Mobile screen on the server. Once the password is saved on the device, you are not prompted for the password when the Siebel Handheld application is launched. Similarly, when you invoke the Siebel Handheld Sync login dialog, the password and user name is pre-populated if password is already saved on the device.

Remote Sync option is only enabled for Companion Synchronization users. Remote Sync allows for integrated synchronization which defines an end to end sync session. Data is uploaded from the device to the desktop and then to the server in the same session. The download of data also happens from the server to the desktop and then to the device.

The Remote Sync option only specifies the mode of remote synchronization that you want to perform. You can select to upload transactions or do a full remote session, which also involves downloading the latest data from the server.

You can select from the following options:

- **Upload Only:** Upload transactions from the companion PC to the server.
- **Upload and Download:** By default Upload and Download is selected. This defines what happens when remote synchronization is invoked.

Troubleshooting Synchronization with the Siebel Handheld Application Client

This topic lists potential error messages that are associated with synchronization. In addition, this topic helps to understand transaction errors.

General Synchronization Errors

There are a number of different errors that can occur when synchronization does not complete successfully, for example:

- Faulty wire connection
- Server crash
- Insufficient memory
- Login error
- Where Microsoft (MS) Embedded Visual Basic (EVB) is installed for PPC 2003 on a handheld device, MS EVB replaces any existing MS DLLs in the \Windows directory, resulting in the following error upon (second and subsequent) handheld synchronizations:

The version of resource DLL MSDAEREN.DLL is not valid.

To fix this problem, copy msdadc.dll, msdaer.dll, and msdaeren.dll from \Program Files\Siebel Handheld directory to the \Windows directory.

When these errors occur, the user must exit Siebel Handheld application synchronization and launch the application again.

NOTE: When you restart Siebel Handheld application synchronization, you might see an error message indicating that another instance of syncmanager.exe is running. You are asked whether you want to terminate the first instance or not, and you must choose Yes.

Transaction Conflicts

Transaction conflicts are minimized when customers adhere to the standard configuration practices outlined in this document. However, even when proper configuration practices are followed, conflicts can occur for a variety of reasons. For example, if a field value is updated on both the Siebel Handheld Application Client and the Siebel Server between synchronization events, a transaction update conflict occurs. In this instance, the user is presented with an error dialog after synchronization has completed.

A transaction consists of one or more smaller transactions called mini-transactions. For example, if you update several fields in a record, the update to the record counts as a single transaction, and each update to a field is a mini-transaction. Individual fields are skipped when update conflicts occur. Changes to any field that has been updated on the server since the last synchronization (perhaps by another Siebel Handheld application user) are discarded (but logged and returned as errors to the user). All non-conflicting fields normally are updated. Except for field-level conflicts, which are skipped when they occur, a transaction is applied in its entirety, and at this point will either succeed or fail in its entirety.

Guidelines for Improving Siebel Handheld Application Performance

If you find that the performance of the Siebel Handheld application is very slow or begins to degrade, the following can help to improve the application performance:

- Verify that you are using a Siebel supported device for your application. See Siebel System Requirements and Supported Platforms on Oracle Technology Network.
- Maximize the amount of memory that is available to run the Siebel Handheld application. When you run the Siebel Handheld application, it is recommended that you do not run any other applications.
- Increase the amount of memory that is allocated for running applications.
- Install the binaries to a CompactFlash card. For more information about this task, see [“Setting Up Siebel Handheld Application Installation on a CompactFlash Card” on page 182.](#)

To increase application memory

- 1 From the Start menu, select Settings.
- 2 Select the System tab and tap Memory.
- 3 Move the slider bar so that more memory is allocated to Program Memory, then tap OK.

The following topics contain guidelines to help reduce the Repository Meta Language (RML) file size, database file (dbfile) size, and synchronization time.

Guidelines for Reducing the Synchronization Time

The process of reducing the sync time and improving performance for the Siebel Handheld application depends on your ability to:

- Reduce the amount of data that is downloaded
- Reduce the period of time it takes to extract the data
- Reduce the size of the .RML file

Follow these guidelines:

- Keep the Siebel Handheld application database file under 3 megabytes (MB).
- Keep the RML file under 1.5 MB.
- Limit the amount of views in the application to 30.

Limit the size of the dbfile.txt file to less than 3 megabytes (MB). The RDBMS on the handheld device will be approximately three times the size of dbfile.txt. If the data files are so large that they cannot be imported into the database with the available memory, users will not be able to successfully synchronize their data. Design data filters to limit the amount of data that is synchronized with the handheld device.

If too many records are downloaded, this increases the length of time it takes to synchronize the handheld device and slows down the Siebel Handheld application performance.

For more information about checking to see how much data is downloaded to your handheld device, see [“Monitoring Synchronization Sessions for Siebel Handheld Applications” on page 135](#).

NOTE: You can also check the Siebel Server log file to see how many records are downloaded for each business component. However, you must turn on logging to capture this information in the log file. For more information about enabling logging for Direct Server Synchronization, see [“Configuring Logging Levels for Direct Server Synchronization” on page 171](#). For more information about enabling logging for Companion Synchronization, see [“Configuring Logging Levels for Companion Synchronization” on page 172](#).

Guidelines for Designing Data Filters

Oracle suggests the following general steps for developing synchronization filters for the Siebel Handheld application:

- Ensure you have configured the Business Object filters, in addition to the Business Component filters. Failure to optimize or set Business Object filters will lead to more data being downloaded, which negatively affects the overall synchronization time. Therefore, set the Business Object filters for your Siebel Handheld application deployment.
- Identify the screens and views of your Siebel Handheld application. Also, identify the business components that are required to support each screen and view, noting the areas where each business component occurs. For example, identify the form, list, or pick applets that are based on each business component.
- Set expectations as what data and how many records will be shown to a user for each view (Low, Avg, High). For Child records, make sure you specify a number based on the parent record.
- List the expected record counts for the picklist and pick applets.
- Prioritize the value of the data that is downloaded.
- Within Siebel Tools, establish the base parent-child relationships and any reverse relationships at the component level.
- Create the appropriate business object synchronization filters for each business object or screen and make sure that each screen includes a default filter. Default filters help to ensure that users avoid accidentally downloading an excessive number of records.
- Review the business components that occur in picklists and consider creating default synchronization filter declarations for those that will cause large numbers of records to synchronize to the handheld device. These declarations are maintained in the PDAQBE file.
- Pay attention to the business components containing enterprise visible data and create suitable filters for:
 - Employees (for large organizations)
 - Positions (for large organizations)
 - Products (for example, active, orderable products)
 - Activities (for example, filter for current Activities, Activities of only certain kinds)
- Ensure that the filter queries are based on the indexed fields, and that they are optimized to download only the required data to the handheld device.

- Translate all parent business component filters to the default business component filters to boost performance significantly.

NOTE: Configuration file directives are used only in specific situations and do not consider them as the primary method for filtering data. Modify these filters and directives cautiously and only with the support of skilled Siebel configurators.

For more information about how to design data filters to limit the amount of data that is synchronized to the handheld device, see [Chapter 5, “Data Filtering for Siebel Handheld Applications”](#).

Guidelines for Reducing Synchronization Time

To reduce the synchronization time, filter the amount of data downloaded to the business objects first. Specifically, set the data filters in this order:

- 1 Business Object filters.** This setting restricts data to the necessary parent records.
- 2 Business Component filters.** This setting extracts the relevant child records from only those parent records that made it through the Business Object filters.

Guidelines for Reducing the Time of Data Extraction

Your Siebel Handheld application generates different types of log and output files. Review the generated log file to identify:

- Which business object, business component, views, applets, and picklists are being extracted and in what order?
- How many records are being extracted?
- How long it takes for each record to be extracted?

Based on the information in the log file you need to identify trouble spots, which might be caused by bad filter query, no indexes, or too much data. For more information about how to address some of these issues, see [“Siebel Handheld Application Performance” on page 284](#).

You can also eliminate pick applets and set run-time property=False for read only fields to avoid the unnecessary extraction for data.

Review the data and evaluate if the same data is being extracted twice under two business components with different names. If this situation is the case, change the configuration to rectify this. Additional configuration information can be found Bulletin 476964.1 on *My Oracle Support*. This document was previously published as Siebel Technical Note 405.

Guidelines for Reducing the Size of the RML File

The Siebel Handheld application performs best if you use the following guidelines:

- Keep the number of views in your application to 30 or fewer.
- For optimal usability, limit each view to one or two applets.

- To minimize horizontal scrolling, limit the number of columns displayed in a list applet to no more than ten.
- There is no limit on the number of fields in form applets of single-applet views. However, to minimize scrolling, keep the number of fields to 20 or fewer.
- Use a form applet for the parent for each parent-child view.
- In a multi-applet view, limit the number of fields in form applets to five or fewer for Pocket PC and 10 or fewer for Handheld PC. Add additional fields only if the field width is short, for example, a check box field. Reduce the number of fields if the fields are multiline. for example, a Comments box that contains three lines of text.
- Limit the number of views within a screen to 12 or fewer so that the View drop-down list is not too long and unwieldy.
- Limit the number of screens to six or fewer so that the Screen drop-down list is not too long and unwieldy.

NOTE: You can control the RML file size by making changes to the CFG file used during Companion Sync. For more information, see [“Using Companion Synchronization with the Siebel Handheld Client” on page 215](#).

The areas that affect the RML file size are:

- **DefaultViewxxx.** The greater the number of views, the larger the RML file size, which results in an increase in sync time.
- **DefaultBusObjs.** An increase in the number of business objects and business component results in a larger RML size and dbfile size.
- **DefaultFields.** An increase in the amount of fields also results in larger RML size and dbfile size.

Guidelines for Improving the Configuration of the Siebel Handheld Application

Use the following guidelines when reviewing your Siebel Handheld application:

- Multi-value fields with links to search specifications have a negative impact on extraction time.
- For custom or heavily configured business components, it might be beneficial in some cases to clone the business component for the Siebel Handheld application to reduce the number of joins, links, and fields that are needed.
- Force Active and Link Spec properties force a field to be extracted to the Siebel Handheld application. If possible, avoid using these properties for fields that do not need to be downloaded to the handheld device.
- Siebel Handheld applications rely on specialized classes and user properties. These might require some Out-Of-Box (OOB) fields to be present when configuring the Siebel Handheld application. During development, ensure to keep the logging levels at the maximum level and review the log files for any missing fields.
- Avoid over-customization because it usually causes performance degradation and problems during an upgrade.

For more information, see [“Configuration Guidelines for the Siebel Handheld Client” on page 28](#).

10 Using Siebel Pharma Handheld

This chapter describes how to use some important features in your Siebel Pharma Handheld application. The following topics are covered in this chapter:

- [“Logging in to the Siebel Pharma Handheld Application” on page 226](#)
- [“Viewing the Homepage for Siebel Pharma Handheld” on page 226](#)
- [“About the Contacts Screen” on page 227](#)
- [“Managing Contacts with Siebel Pharma Handheld Applications” on page 227](#)
- [“Viewing and Creating Alerts with Siebel Pharma Handheld Applications” on page 233](#)
- [“Viewing Events in Siebel Pharma Handheld” on page 233](#)
- [“Viewing and Creating Assessments in Siebel Pharma Handheld” on page 234](#)
- [“About the Accounts Screen” on page 235](#)
- [“Managing Accounts with Siebel Pharma Handheld Applications” on page 235](#)
- [“Viewing Product Focus in Siebel Pharma Handheld” on page 239](#)
- [“Planning and Recording Calls in Siebel Pharma Handheld Applications” on page 241](#)
- [“Call Management Procedures in Siebel Pharma Handheld Applications” on page 243](#)
- [“Creating Contact Calls in Siebel Pharma Handheld” on page 244](#)
- [“Creating Contact Calls in Siebel Pharma Handheld” on page 244](#)
- [“Recording Contact Calls in Siebel Pharma Handheld Applications” on page 247](#)
- [“Recording a Contact Call for a Different Contact at the Time of Signature Capture” on page 252](#)
- [“Creating Account Calls in Siebel Pharma Handheld” on page 253](#)
- [“Recording Account Calls in Siebel Pharma Handheld Applications” on page 255](#)
- [“Creating General Activities in Siebel Pharma Handheld Applications” on page 257](#)
- [“Managing Samples in Siebel Pharma Handheld Applications” on page 259](#)
- [“Setting Up MedEd Events in Siebel Pharma Handheld Applications” on page 271](#)
- [“Creating Expense Reports in Siebel Pharma Handheld Applications” on page 273](#)

Logging in to the Siebel Pharma Handheld Application

To start using your Siebel Pharma Handheld application, you must enter a user ID and password. The following steps assume that your Handheld application is configured to require users to enter a user ID and password to log in. If the Auto Login feature is enabled, Handheld application users might be required to enter only a password to log in, or instead they might not have to enter any information at all.

To log in to a Siebel Pharma Handheld application

- 1 Start the Handheld application.
- 2 Enter your password.
- 3 Select the Save Password check box if you want to access the application directly from now on, rather than enter your password each time.

NOTE: Your system administrator might make this option unavailable in your application.

- 4 Tap OK.

NOTE: To make sure that your database is secure, always close the application from the application-level menu. For information about how to close the application, see the online help for the Siebel Handheld application.

Viewing the Homepage for Siebel Pharma Handheld

From the Pharma Handheld application, you can select the Homepage option from the drop-down list. Tap any of the following buttons

- Contacts
- Accounts
- Calendar
- Expense Reports
- MedEd Events
- Samples

The view also has tabs for the following applets:

- **What's new:** This applet is based on the Partner Message business component and shows the following fields: Priority, Abstract, and Message Body.
- **Alert:** This applet is based on the Action business component and shows Alerts generated by the application and those alerts manually created by users. This is a read-only applet.
- **Activities:** This applet shows the following fields: Activity Type, Description, Priority, Last Name, First Name.

About the Contacts Screen

The Contacts screen contains the following view options and associated subtabs:

- My Contacts
- Profile
 - Licences
 - Rankings and Ratings
 - Rx Data (Read-only view of the contact's historical prescription data)
 - Affiliations
- History
 - My Calls
 - Activities
 - Notes (These are public notes. Shared by all sales representatives who call the contact in context)
 - Best Times
- Communications
 - Alerts
 - Events
 - Messages
 - Assessments

Managing Contacts with Siebel Pharma Handheld Applications

A *contact* is typically a physician, nurse, pharmacist or other medical professional. In a more general sense, a contact is any individual a pharmaceutical company deems significant to their business process, such as a formulary director, contracts administrator, or medical education event speaker.

Sales representatives use Siebel Pharma Handheld to add and update contact profile information which might include adding new contacts, creating new contact addresses, tracking affiliations between contacts and accounts, recording the best times to call on a contact, transcribing relevant notes, and viewing prescription data.

This topic describes how to manage contact information using all of the views accessible from the Show drop-down list of the Contacts screen, except for Calls and Activities.

The following contact procedures are included in this topic:

- [“Adding Contacts in Siebel Pharma Handheld” on page 228](#)
- [“Adding Contact Addresses in Siebel Pharma Handheld” on page 229](#)

- [“Adding and Editing License Information in Siebel Pharma Handheld” on page 230](#)
- [“Specifying the Best Time to Call on a Contact in Siebel Pharma Handheld” on page 230](#)
- [“Adding Shared Notes About a Contact in Siebel Pharma Handheld” on page 231](#)
- [“Indicating Affiliations Between Contacts and Accounts in Siebel Pharma Handheld” on page 231](#)
- [“Viewing Prescription Data in Siebel Pharma Handheld” on page 232](#)

For information on the Calls and Activities views, see [“Creating Contact Calls in Siebel Pharma Handheld” on page 244](#) or [“Creating General Activities in Siebel Pharma Handheld Applications” on page 257](#).

Adding Contacts in Siebel Pharma Handheld

Using the Contacts screen and the My Contacts view, you can add contacts and update contact information directly in Siebel Pharma Handheld.

To add contacts

- 1 Navigate to the Contacts screen.
The My Contacts list appears.
- 2 On the toolbar, tap the New Record icon.
- 3 Complete the Last Name, First Name, Main Phone, Type, and Title fields for the new record.

To add additional contact information

- 1 Navigate to the Contacts screen
The My Contacts list appears.
- 2 Tap the Last Name link in the My Contacts list.
- 3 Add or edit information in the available fields. Some of the fields are described in the following table.

| Field | Comments |
|-------------------|---|
| Address | A contact can have more than one address. One address must be specified as primary. Addresses are entered using the Addresses view. For more information about adding addresses, see “Adding Contact Addresses in Siebel Pharma Handheld” on page 229 . |
| Last Call Date | Read only. The date of the most recent call that has been submitted. |
| OK to Sample | Checking this box indicates that it is permissible to drop samples off with this contact. |
| Primary Specialty | Read-only. Siebel administrators define this value. |

| Field | Comments |
|---------------|--|
| Rep Specialty | The specialty or area of expertise associated with this contact within the contact's organization. |
| Route | Allows users to select when to visit an account from a predefined route schedule. |
| Title | Must be filled in before a contact can sign for receipt of samples. |
| Type | Indicates the type of contact. |

Adding Contact Addresses in Siebel Pharma Handheld

Contact addresses are entered in the Addresses view. Since a contact can have multiple addresses, and multiple contacts might share an address, take care when editing an address as the changes might affect more than one contact.

When entering addresses, specify one address as the primary. Each sales representative can specify a different primary address for an assigned contact. For example, one representative might specify a private office as the primary address, while another representative might designate a hospital department as the primary address.

The More Info form only displays the primary address, not all the addresses associated with the contact.

To add contact addresses

- 1 Navigate to the Contacts screen
The My Contacts list appears.
- 2 Select a record, and from the View drop-down list, select Profile.
- 3 Tap the drop-down list and the application displays a list of addresses.
- 4 In the Addresses list, do one of the following:
 - Tap New to add a record, and then complete the necessary fields.
 - Tap an existing address in the list, tap OK, and the edit the address as required.
- 5 In the Address list, tap the check box in the record's Primary field to designate the record as the contact's primary address.

A check mark appears in the box to designate the primary address.

Adding and Editing License Information in Siebel Pharma Handheld

Using the License Numbers view, you can add the contact's professional license number, such as a state medical license number or a physician investigator's American Medical Association number.

Before a physician can sign for samples dropped, you must enter the physician's license numbers and additional license information. The contact must have a valid license number for the state corresponding to the address where samples are disbursed.

To add and edit license information

- 1 Navigate to the Contacts screen.
The My Contacts list appears.
- 2 Select a contact, and from the View drop-down list, select Profile.
The application displays the Licenses tab in the toolbar.
- 3 Tap the License tab, and then do one of the following:
 - On the toolbar, tap the New Record button, and then complete the fields for State, License number, License Expiration Date, and license Status.
 - Tap a License Number record to select it, and then edit the information in the fields.

Specifying the Best Time to Call on a Contact in Siebel Pharma Handheld

Using the Best Time to Call view, you can enter the best times to call on a contact based on a selected address. You can then query for contacts available during a specific time period.

To specify the best times to call on contacts

- 1 Navigate to the Contacts screen.
The My Contacts list appears.
- 2 Select a contact, and from the View drop-down list, select History.
The application displays the Contact and Best Time to call tabs at the bottom of the screen.
- 3 Tap the Best Times to Call tab to select it.
- 4 On the toolbar, tap the New Record button.

- 5 Complete the fields for the new record, as described in the following table.

| Field | Comments |
|------------|---|
| Address | Indicates which contact address pertains to this best time to call information. Select the address from the Pick Location dialog box, and tap OK. The Pick Location dialog box displays all addresses associated with that contact. |
| Start Day | Indicates the day that the contact is available for a call at that address. |
| Start Time | Indicates the start time for the selected day that the contact is available for a call. |
| End Time | Indicates the end time for contact availability. |
| Comments | Text field for information about time period constraints and other relevant notes. |

Adding Shared Notes About a Contact in Siebel Pharma Handheld

Using the Shared Notes view, you can create and update notes on contacts. Shared notes are available to other sales representatives calling on the contact.

To add notes about contacts

- 1 Navigate to the Contacts screen, and select a contact from the My Contacts list.
- 2 From the View drop-down list, select History.
The application displays the Notes tab on the toolbar.
- 3 Tap the Notes tab to select it.
- 4 On the toolbar, tap the New Record button.
A new row appears in the list.
- 5 Complete the necessary fields for the record.
Tap in the Note field, and then tap the ellipsis (...) to view the entire note while you are entering text.

Indicating Affiliations Between Contacts and Accounts in Siebel Pharma Handheld

Using the Affiliations view, you can associate a contact with an account. One reason to establish these affiliations is for call reporting purposes. Affiliations between accounts and contacts also can be established using the Accounts screen.

To specify affiliations between contacts and accounts

- 1 Navigate to the Contacts screen, and select a contact from the My Contacts list.
- 2 From the View drop-down list, select Profile.
- 3 Tap the Affiliations tab on the toolbar.
The Account Affiliations view appears.
- 4 Tap the Account Affiliations list to select it, and then tap the New Record button.
- 5 Complete the necessary fields for the new record.

Some of the fields are automatically populated based on the chosen Account. Other fields are described in the following table.

| Field | Comments |
|------------|---|
| Account | The account to be affiliated with the selected contact. Select a predefined account from the Pick Account list, and then tap OK. |
| Comments | Text field for user comments. |
| Direct | If a check mark appears in this field, profile data for the contact is routed to members of the team assigned to the affiliated account. If no check mark appears in this field, the profile data is routed only to the members of the team assigned to the contact. Tap the check box to specify a direct affiliation between a contact and an account. |
| Primary | If a check mark appears in this field, the account is the primary affiliation for the contact. Tap the check box to display a check mark. |
| Start Date | The beginning date on which the contact became affiliated with this account. |

Viewing Prescription Data in Siebel Pharma Handheld

Using the Rx Data view, you can view prescription data to better target medical professionals. Data about physicians and prescribed drugs helps pharmaceutical companies define their market for better strategic business planning and national account. Prescription data can be analyzed to measure market potential, target customers, align sales territories, and track market shifts.

To view prescription data

- 1 Navigate to the Contacts screen, and select a contact from the My Contacts list.
- 2 From the View drop-down list, select Profile.
- 3 Tap the Rx Data tab on the toolbar.
The Rx data associated with the contact appears in the RX Data list.
- 4 In the list, review the prescription data on a market-by-market basis.

Viewing and Creating Alerts with Siebel Pharma Handheld Applications

Alerts specific to each contact can be used to track and remind the user of important events.

To view and create alerts

- 1 Navigate to the Contacts screen and select Communications from the Views drop-down list.
- 2 Tap on the Alerts tab in the toolbar.

The application displays a list of alerts specific to the contact in context. Some of the fields are described in the following table.

| Field | Comments |
|------------|---|
| Category | Information or Warning. |
| Created By | User who created this alert. |
| Duration | Duration of the alert. |
| Expiration | The date and time on which the alert expires. |
| Message | Text of the alert message. |
| Start Date | Start date from which the alert is applicable. |
| Type | A drop-down list which displays all the available types of alerts in the system. |
| Unit Type | A drop-down list that is applicable to the duration of the alert. Value could be Days, Months, Weeks, or Years. |

Viewing Events in Siebel Pharma Handheld

Events specific to each contact can be viewed in Siebel Pharma handheld.

To view events

- 1 Navigate to the Contacts screen and select Communications from the Views drop-down list.

- 2 Tap on the Events tab in the toolbar.

The application displays a list of events specific to the contact in context. Some of the fields are described in the following table.

| Field | Comments |
|----------------|--|
| Event | A read only field indicating the name of the event |
| Start Date | A read only field which indicates the scheduled start date. |
| Session | Name of the session within a multiple-session event. |
| Invitee Status | A read only field. Possible values include Waiting, Confirmed and so on. |

Viewing and Creating Assessments in Siebel Pharma Handheld

Assessments allow users to calculate a single numerical value that expresses the fitness of a contact or account, according to a set of attributes defined in the assessment template. For more information on assessments, see *Applications Administration Guide*.

To view and create assessments

- 1 Navigate to the Contacts screen and select Communications from the Views drop-down list.
- 2 Tap on the Assessments tab in the toolbar.

The application displays a list of assessments. Some of the fields are described in the following table.

| Field | Comments |
|---------------|--|
| Description | Description of the assessment. |
| Name | Name for the assessment. |
| Percentage | Percentage as calculated by the Pharma Handheld application and shown in the Assessments list. |
| Template Name | Name of the template from a pre-existing list. |

- 3 Tap File, then New Record to create a new record.
- 4 Tap the Template Name field, and select a template from a list of templates displayed.
- 5 Tap OK.

The Assessment List view is displayed.

- 6 Enter a name for the assessment and drill down on the assessment.
You are prompted to answer some questions that are specific to the template.

About the Accounts Screen

The Accounts screen contains the following view options and associated subtabs:

- My Accounts
- Profile
 - Affiliations
 - Sales List
- History
 - My Calls
 - Activities
 - Notes (Add public notes related to the account in context from this view)
 - Best Times

Managing Accounts with Siebel Pharma Handheld Applications

An *account* is any health care business that generates sales for your company or that could potentially generate business. Typical examples of accounts include hospitals, clinics, HMOs, wholesalers, group purchasing organizations (GPOs), and pharmacies. An account can also be any organization with which multiple contacts are associated, such as a group practice or an assisted-living facility.

Sales representatives in the field use Siebel Pharma Handheld to add and update account profile information. Representatives use Siebel Pharma Handheld to add new accounts, create new account addresses, record the best times to call on an account, and record relevant notes. Sales representatives can also view sales data on Siebel Pharma Handheld to help them evaluate the business potential of accounts.

This topic describes how to manage account information using all of the Accounts views accessible from the Show drop-down list, except the Calls and Activities views. For information on those views, see [“Creating Account Calls in Siebel Pharma Handheld” on page 253](#) or [“Creating General Activities in Siebel Pharma Handheld Applications” on page 257](#).

The following account procedures are included in this topic:

- [Adding Accounts in Siebel Pharma Handheld on page 236](#)
- [Adding Account Addresses in Siebel Pharma Handheld on page 236](#)
- [Specifying Best Times to Call Accounts in Siebel Pharma Handheld on page 237](#)

- [Adding Notes About Accounts on page 238](#)
- [Indicating Affiliations Between Accounts and Professionals in Siebel Pharma Handheld on page 238](#)
- [Viewing Sales Data in Siebel Pharma Handheld on page 239](#)

Adding Accounts in Siebel Pharma Handheld

Accounts are businesses and organizations that either currently generate sales for your company or have the potential to do so. You can add or edit accounts using the My Accounts view.

NOTE: Only a Siebel administrator can delete an account.

To add an account

- 1 Navigate to the Accounts screen.
The My Accounts list appears.
- 2 Tap File, then New Record and complete the necessary fields for the new record.
- 3 To enter information in additional fields, click on the Account name and drill down to view further details.

Some of the fields are described in the following table.

| Field | Comments |
|---------------|--|
| Account | Account name. Tapping the account name hyperlink History opens the Call Activities view. |
| Address | The account can have more than one address. One address must be specified as primary. Account addresses must be entered in the Addresses view. For more information about adding addresses, see “Adding Account Addresses in Siebel Pharma Handheld” on page 236 . |
| Site | A unique location identifier that distinguishes this account from any other accounts with the same name. |
| Mkt Potential | Used to evaluate the business potential of the account. |

Adding Account Addresses in Siebel Pharma Handheld

Addresses for accounts are entered in the Addresses view. Since an account can have multiple addresses, care must be taken when editing an account address, as changes affect all accounts associated with the address.

When entering account addresses, designate one address as the primary. Each sales representative assigned to an account can specify a different primary address. For example, one representative might specify a private office as the primary address, while another representative might specify a hospital department as the primary address.

The My Accounts view shows the primary address for the Account, by default. The More Info form displays the primary address, not all the addresses for the contact.

To add account addresses

- 1 Navigate to the Accounts screen.
The My Accounts list appears.
- 2 Select a record, and tap Address.
All the addresses for the Account are displayed.
- 3 Tap the New Record button.
- 4 In the Add Address dialog box, do one of the following:
 - Tap New to create a new address record, and then complete the fields.
 - Select an existing address from the list, tap OK, then edit the address.
- 5 In the Primary field, tap the check box to display a check mark if this is the account's primary address.

Specifying Best Times to Call Accounts in Siebel Pharma Handheld

Use the Best Times to Call view to specify appropriate time periods for account calls.

To specify best times to call accounts

- 1 Navigate to the Accounts screen.
The My Accounts list appears.
- 2 Select an account, then from the View drop-down list, select History.
- 3 Tap on the Best Times tab and on the toolbar, tap the New Record button.
- 4 Complete the necessary fields for the new record. Some of the fields are described in the following table.

| Field | Comments |
|------------|--|
| Start Day | Indicates the days that the contact is available for a call. |
| Start Time | Indicates the time that the contact is available for a call. |
| End time | Indicates end of available time period. |
| Comments | Text field for schedule information. |

Adding Notes About Accounts

Using the Shared Notes view, you can create and update shared account notes. Shared notes are available to other sales representatives calling on the account.

To add account notes

- 1 Navigate to the Accounts screen.
The My Accounts list appears.
- 2 Select an account from the My Accounts list.
- 3 From the View drop-down list, select History:
- 4 Tap the Notes tab to select it.
- 5 On the toolbar, tap the New Record button, then complete the fields for the new record.
Tap in the Note field, and then tap the ellipsis (...) to view the entire note while you are entering text.

Indicating Affiliations Between Accounts and Professionals in Siebel Pharma Handheld

For accounts where you have contact with multiple professionals, you can indicate the affiliation of those professionals with the account by using the Affiliations view. Creating affiliations between accounts and contacts promotes efficiency in call reporting.

To indicate affiliations between accounts and contacts

- 1 Navigate to the Accounts screen
The My Accounts list appears.
- 2 Select an account from the My Accounts list, and from the View drop-down list, select Profile.
- 3 Tap the Affiliations tab.
- 4 Tap the Contact Affiliations list to select it.
- 5 On the toolbar, tap the New Record button.
- 6 Complete the necessary fields for the new record. Some of the fields are described in the following table.

| Field | Comments |
|-----------|--|
| Last Name | The contact affiliated with this account. Tap the field, select the contact from the Pick Contact list, and then tap OK. |

| Field | Comments |
|------------|---|
| Start Date | The beginning date on which the account became affiliated with this contact. |
| Direct | <p>Tap the check box to display a check mark.</p> <p>If a check mark appears in this field, profile data for that person is routed to all members of the team assigned to the affiliated account.</p> <p>If no check mark appears in this field, the profile data is routed only to members of the team assigned to the person.</p> |

Viewing Sales Data in Siebel Pharma Handheld

Sales data can be analyzed to measure market potential, target customers, align sales territories, and track market shifts. Sales data is typically syndicated information that is loaded periodically by the Siebel administrator.

You can review this information using the Sales Data view.

To view sales data

- 1 Navigate to the Accounts screen.
The My Accounts screen appears.
- 2 Select an account from the My Accounts list, and from the View drop-down list, select Profile.
- 3 Tap the Sales List tab.
- 4 Review the data in the list on a product-by-product basis.

Viewing Product Focus in Siebel Pharma Handheld

With the introduction of allocation functionality, administrators allocate products to sales representatives. Sales representatives can view the product allocation using the Product Focus view. For more information about the allocation functionality, see *Siebel Life Sciences Guide*.

The Product Focus view has four tabs as follows:

- Samples
- Sample Requests
- Products
- Promotional items

For sales representatives to be able to select samples at the time of call recording, the allocation period for the product must be active, and they must have some samples in their inventory.

To view samples allocated

- 1 Navigate to the Product Focus screen.
- 2 Tap on the Samples tab.

The application displays allocations with a Start Date less than or equal to today's date and End Date is Null, or End Date is greater than or equal to today date plus 30.

Some of the fields are described in the following table.

| Field | Comments |
|---------------|--|
| Description | Description of the sample. |
| Item | Displays the Lot number of the sample. |
| Order | Enter a number that indicates where this product appears in the drop-down list. This field appears in the Details list, Samples list, and Promotional Items list. For example, if you want a product in the Details list to appear first, enter 1. If this field is null for all products in the list, the products appear in alphabetical order. |
| Sample | The products to be dropped as samples during contact and account calls. |
| Stop Sampling | This is an allocation functionality related flag that can be set by the administrator. |

You can view all sample requests allocated.

To view sample requests allocated

- 1 Navigate to the Product Focus screen.
- 2 Tap the Sample Requests tab to view the sample requests allocated.

You can view products allocated.

To view products allocated

- 1 Navigate to the Product Focus screen.
- 2 Tap the Products tab to view the products allocated.

You can view promotional items allocated.

To view promotional items allocated

- 1 Navigate to the Product Focus screen.
- 2 Tap the Promotional Items tab to view the promotional items allocated.

Planning and Recording Calls in Siebel Pharma Handheld Applications

Planning calls in Siebel Pharma Handheld consists of creating account and contact calls. Recording calls involves entering information on products detailed, samples and promotional items dropped, and decision issues that occurred during the call.

- [Contact and Account Calls on page 241](#)
- [Smart Call and New Call on page 241](#)

Contact and Account Calls

Contact and account calls are types of activity records, although contact and account calls cannot be created from the Activities view.

You can create contact calls as follows:

- Tap the New Call button from the Contact form applet
- Tap the New call button from the Calendar > Daily view
- Create a new record from the Contacts screen's Calls view
- Create a new record from the Contact Calls screen

You can create account calls as follows:

- Tap the New Call button from the Account form applet
- Create a new record from the Accounts screen's Calls view
- Create a new record from the Account Call screen
- **Contact call.** Used to track a planned or past call activity with a contact. You can record details about contact calls such as the products detailed, the samples and promotional items dropped, and decision issues discussed.
- **Account call.** Used to track a planned or past call activity at an account. You can record products detailed, attendees, promotional items dropped, and decision issues discussed.

Smart Call and New Call

Siebel Pharma Handheld provides several features to manage contact and account call records:

- **Smart Call.** Specifying a Smart Call template for the call record associates a prepopulated call template with this call.

A Smart Call is a template that you can apply to contact calls and account calls to simplify and speed up the call-reporting process. A Smart Call defines a set of information about the products detailed, the samples distributed, samples requested, and the issues discussed on a call. Smart Call templates are created in the Siebel Pharma application, not in the Siebel Pharma Handheld application.

Smart Calls can be also used for calls related to a particular campaign or promotional effort, or for everyday call reporting. They are intended for situations where sales representatives report details about a number of calls that are identical in content (the same products are detailed in the same order, the same samples are dropped, and so on). After you apply a Smart Call template to a call, you can make additional edits in any fields in the Contact Call screen's Details view.

- **New Call.** A user can create a Call record directly from within this Contacts List applet. The New Call button on the Contacts form applet creates a Contact Call record with the Contact Ids populated. The created Call records are displayed in the Contacts Call Detail view under Activities.

For more information about planning and recording calls in Siebel Pharma Handheld, see [Scenarios for Calls in Siebel Pharma Applications on page 242](#).

Scenarios for Calls in Siebel Pharma Applications

The following scenarios depict typical work sequences followed by sales managers and representatives using the Siebel Pharma application and the Siebel Pharma Handheld application. Your company might follow a different sequence according to its business requirements.

Scenario for Planning Calls in the Siebel Pharma Application and Recording Calls in the Siebel Pharma Handheld Application

This topic gives one example of how you might plan and record calls in Siebel Pharma Handheld applications. You might plan and record calls differently, depending on your business model.

A sales representative for a pharmaceutical company wants to set up visits to their customers over the next few days. To do so, they first use the Siebel Pharma application on their laptop to plan contact calls. Then they synchronize the data with the Siebel Pharma Handheld application on their handheld device, which they carry around with them to record call details.

To plan calls, the sales representative begins by analyzing the physicians' prescribing habits. They then query on geographical location, specialties, and time lapse since they last called on them to determine which physicians they must call on.

Using the Siebel Pharma application, the sales representative applies a Smart Call template to those calls so that information about the samples products is automatically copied into the call records. They have now prepopulated the calls with the details that they want to speak about with each physician as well as the samples they want to drop off.

The sales representative synchronizes their Handheld application so that they can carry these details with them and record their progress throughout the day. The information that is available on their Handheld application includes previous calls and other activities, notes, and physician prescription history.

Before meeting with the physician, the sales representative goes to the samples closet in their office and drops off the samples. They record the lot numbers on their handheld device.

When the sales representative meets with the doctor, they describe the samples that they have for them, explain the benefits of key products, and discuss the doctor's concerns. If the doctor expresses interest in additional product samples, the sales representative records additional samples to the contact call, and then captures the doctor's signature on the handheld device.

After the meeting, the sales representative records the information on the products detailed and promotional items and issues discussed. After the sales representative has reviewed the information and verified that it is accurate, they submit the call.

Scenario for Planning and Recording Calls

This topic gives another example of how you might plan and record calls in Siebel Pharma Handheld applications. You might plan and record calls differently, depending on your business model.

That same day, another sales representative sets off to visit several clinics. Unlike the first sales representative, this representative does not schedule their calls on their laptop beforehand for two reasons. First, they plan to call on group practices with many physicians, and they cannot be sure which physician will be signing for the samples. Secondly, they prefer to enter sample information manually rather than using a Smart Call template.

Before they enter the first clinic, they query for physicians associated with that address. Seven names appear. When walking into the first clinic, they encounter one of the doctors so they select that physician's record and adds a new call. While speaking with the doctor about the benefits of the product, the sales representative hands the doctor some samples and records the samples dropped information. The sales representative then obtains the doctor's signature on the handheld device. After adding a comment about the call and recording products detailed, the sales representative submits the call.

Call Management Procedures in Siebel Pharma Handheld Applications

This topic provides the following procedures related to planning, recording, and submitting calls.

- Creating, recording, and submitting contact calls:
 - [To add a contact call from the Contacts view on page 244](#)
 - [To add a contact call from the Contact Calls view on page 246](#)
 - [To add a contact call from the Calendar screen on page 247](#)
 - [To apply a Smart Call template on page 247](#)
 - [To record information about products detailed on page 247](#)

- To record information about samples dropped on page 248
- To record information about promotional items dropped on page 249
- To record information about decision issues discussed during the call on page 249
- To capture a signature on page 250
- To submit a completed contact call on page 251
- To create a call based on an existing contact call for a new contact on page 252
- To change the contact or address of an existing call on page 253
- To create a new Sample Drop contact call based on an existing call on page 253
- Creating, recording, and submitting account calls:
 - To add an account call from the Accounts view on page 253
 - To add an account call from the Account Call view on page 255
 - To enter products detailed information for an account call on page 255
 - To enter attendee information for an account call on page 256
 - To submit a completed account call on page 256

Creating Contact Calls in Siebel Pharma Handheld

You can create contact calls and update contact call details using Siebel Pharma Handheld. Because contact calls are a type of activity, they also appear in the Calendar view.

As a shortcut for creating contact calls, you optionally can apply a Smart Call to the new record. For more information about contact calls, see [“Smart Call and New Call” on page 241](#).

To add a contact call from the Contacts view

- 1 Navigate to the Contacts screen.
The My Contacts list appears.
- 2 Select a contact in the Contacts list.
- 3 In the Contacts list, tap the New Call button and complete the fields in the new form. The Execute view is displayed by default. Some of the fields are described in the following table: .

| Field | Comments |
|-----------|--|
| Last Name | The contact to be visited during this call. |
| Ref # | A reference number for this call. This reference number becomes the transaction number (Transaction # field) for a samples transaction corresponding to this call. |

| Field | Comments |
|----------------|--|
| Smart Call | Smart Call is a prepopulated call template that can be associated with this call. Make a selection from the Pick Smart Call dialog box. |
| Status | <p>The status of this call. The default value is Planned. After synchronization, this value changes to Synchronized for calls that have not been submitted.</p> <p>You cannot select the value Submitted. To change the status of the call to Submitted, you must submit the call.</p> |
| Paper Sign | Indicates that the contact provided a paper signature for a sample disbursement. |
| Remake Receipt | <p>Used to create an updated sample receipt and signature in the event that the original call information was incorrect.</p> <p>For example, if a DoctorA mistakenly signs for the receipt of samples on a request form having a different physician's name (such DoctorB), then the sample form is inaccurate. In this example, the sales representative must change the name on the original call to the correct physician (DoctorA) and have DoctorB sign again.</p> <p>Using the Remake Receipt check box, the sales representative can capture a second signature on the corrected sample form. Siebel Pharma Handheld applications automatically tracks the appropriate audit trail information required by the FDA in the remake receipt functionality.</p> |

- 4 Tap the Details tab under the Execute view to specify which products to detail during the call. Some of the fields are described in the following table.

| Field | Comments |
|------------|--|
| Name | Name of the product being detailed. |
| Priority | Priority of the product being detailed. By default, the application displays a value of zero for the field. You can enter a different priority. All records which have priority of greater than zero are captured in the call. |
| Indication | Select an appropriate reason from the drop-down list. |

- 5 Tap the Samples tab to specify which samples to drop off during the call.

The application displays two buttons as follows:

- a Best Lots: Tap to view the products which are current and are before the cut off date.

- b** Show All: Tap to view the products which are within the allocation period or are short dated (past the cut off date but before the expiration date).

The application displays a list applet with the following fields:

| Field | Comments |
|----------|---|
| Name | Name of the product being sampled. |
| Lot # | Lot to which this particular sample belongs to (if applicable). |
| Quantity | This is an editable which field which you can use to enter the number of samples dropped at the physician's office. |

- 6** Enter quantities for each of the Samples.
- 7** Tap the Sample Requests tab to specify samples requested by the physician for direct send (head office to disburse samples directly to the physician)
- 8** Enter quantities for each of the Samples.
- 9** Tap on the Promotional Items tab in the Call Details view to specify which products are promotional items.
- 10** Enter quantities for each of the Promotional Items.
- 11** To specify which issues were discussed during a call:
 - a** From the Show drop-down list, select Issues.

NOTE: A contact might raise decision issues as objections when evaluating a product before making a decision on a prescription.
 - b** Tap the Issues list to select it, add a new record, and complete the necessary fields.

For more information about how to enter information, see ["Recording Contact Calls in Siebel Pharma Handheld Applications" on page 247.](#)

NOTE: If you apply a Smart Call to the call, you might only need to verify that the information in the four lists is accurate.

To add a contact call from the Contact Calls view

- 1** Navigate to the Contact Call screen.
- 2** Drill down on the Last Name field in the Contacts view, choose the contact you want and tap OK.
- 3** Complete the necessary fields.

To add a contact call from the Contacts list

- 1** Navigate to the Contacts screen.
- 2** Select the Contact and tap the New Call button.
- 3** Complete the necessary fields.

To add a contact call from the Calendar screen

- 1 Navigate to Calendar screen.
- 2 Navigate to the Daily view for the day you want to schedule your call.
- 3 Highlight time frame for the Contact Call you want to schedule.
- 4 Tap New Call.
The Contact Call Screen appears with Date/Time field prepopulated with the date/time you selected.
- 5 Complete the necessary fields.

To apply a Smart Call template

- 1 Navigate to the Contact Call screen, and tap the Contact Call form to select it.
- 2 Drill down in the call details.
- 3 In the Smart Call field, tap the drop-down list arrow.
The Pick Smart Call list appears.
- 4 Select a Smart Call template, and tap OK.

Recording Contact Calls in Siebel Pharma Handheld Applications

Contact calls are used to track a planned or past call activity with a contact. You can enter contact call details directly in Siebel Pharma Handheld applications.

To record a contact call, users must complete the following tasks:

- Enter information on products detailed.
- Enter information about samples dropped.
- Enter information about samples requested.
- Enter information about promotional items dropped (optional).
- Record any decision issues (optional).
- Capture signatures.
- Submit the call.

The next topic explains how to perform each of these tasks.

To record information about products detailed

- 1 Navigate to the Contact Call screen, and select the desired contact call.
- 2 From the View drop-down list, select Execute.

- 3 Tap the Details tab.
The Product Details view appears.
- 4 Tap the Product Details list to select it.
- 5 Complete the fields for each record. Some of the fields are described in the following table.

| Field | Comments |
|------------|--|
| Indication | The purpose or therapeutic area for which a drug is designed. |
| Priority | Priority of this product relative to others discussed during the call. |
| Name | The name of the product detailed during the call. |

To record information about samples dropped

- 1 Navigate to the Contact Call screen, and select the desired contact call by selecting the correct record from View option in the toolbar.
- 2 From the View drop-down list, select Execute.
- 3 Tap the Samples Dropped tab.
The Samples Dropped list appears.
- 4 Tap the Samples Dropped list to select it, and complete the fields for each product dropped during the call. Some of the fields are described in the following table.

| Field | Comments |
|----------|--|
| Lot # | The lot number (if applicable) of the products dropped during the call. If tracking samples by lot number, select a lot number in the Pick Lot dialog box. |
| Name | The name of the products dropped during the call. The products appearing in this drop-down list are based on the products available in your samples inventory. For more information about samples, see “Managing Samples in Siebel Pharma Handheld Applications” on page 259 . |
| Quantity | The quantity of samples dropped during the call. |

NOTE: If the samples you dropped do not appear in the Name field drop-down list, you might need to acknowledge the receipt of the samples within the system. Go to the Samples Receipts screen and submit any records with a status of In Progress. For more information about acknowledging receipts, see [“Acknowledging Receipt of a Samples Transfer in Siebel Pharma Handheld” on page 269](#). If the samples you dropped still do not appear, contact your samples administrator.

To record information about samples requested

- 1 Navigate to the Contact Call screen, and select the desired contact call.
- 2 From the View drop-down list, select Execute.

- 3 Tap the Samples Dropped tab.
The Samples Dropped list appears.
- 4 Tap the Samples Dropped list to select it, and complete the fields for each product dropped during the call.

To record information about promotional items dropped

- 1 Navigate to the Contact Call screen, and select the desired contact call.
- 2 From the View drop-down list, select Execute.
- 3 Tap the Promotional Items tab.
The Promotional Items view appears.
- 4 In the Promotional Items list, complete the necessary fields. Some of the fields are described in the following table.

| Field | Comments |
|----------|---|
| Name | The promotional item dropped during the call. The items appearing in this drop-down list are based on items added to your Personal Promotional Items list. For more information about the Personal Promotional Items list, see Creating Contact Calls in Siebel Pharma Handheld on page 244 . |
| Quantity | The quantity of the promotional item dropped during the call. |

To record information about decision issues discussed during the call

- 1 Navigate to the Contact Call screen, and select the desired contact call.
- 2 Tap the Issues tab
The Professional Issues view appears.
- 3 Tap the Issues list to select it.
- 4 On the toolbar, tap New Record to add a record for each issue discussed during the call.
- 5 Complete the necessary fields for each new record.

NOTE: A contact might raise decision issues as objections when evaluating a product before making a prescribing decision.

You can use Siebel Pharma Handheld to electronically capture signatures. For the application to accept a signature:

- Make sure the title field for the contact from Contacts, My Contacts view is populated.
- Make sure the contact's license number appears in the License # field of the Contact view. Licenses are associated with the contact's address.

To capture a signature

- 1 Navigate to the Contact Call screen, and select the desired parent Contact Call form.
- 2 From the View drop-down list, select Profile, Wrap-up, or Execute as appropriate.
- 3 Tap the Sign button to launch the Signature Capture view.

The Signature Capture view contains the company name, samples recorded when the call gets submitted, a disclaimer (as set up by the administrator on the Web Client), and contact information. For more information about disclaimers, see *Siebel Life Sciences Guide*.

NOTE: The Pharma Handheld application displays only static disclaimers.

If you want to invert the signature display, tap the Signature Flip button instead of the Sign button. Inverting the signature display allows you to extend the handheld device toward the contact for a signature without turning the device around.

- 4 In the Signature Capture view, have the contact sign on the white surface.
- 5 Perform one of the following actions:
 - Tap OK to commit the record.
 - Tap Cancel to abort the procedure.
 - Tap Clear to erase the signature area.
 - Tap Change Contact to select the correct contact whose signature you are capturing.

NOTE: The application automatically locks the keyboard hot keys when the Signature Capture view is displayed.

To validate the call

- 1 Navigate to Call Details screen.
- 2 Tap on the Validate button.

When the validation completes, a dialog either displays that there are no errors or displays all the errors from the underlying validation engine (LS Pharma Validation Service - CE).

To create private notes for the call

- 1 Navigate to Contact Call screen.
- 2 Select Wrap-up from the View drop down list.
- 3 Tap the Call Notes tab.
- 4 Enter details to the following fields to make private notes for the calls:
 - Comment
 - Next Call Date
 - Next Call Objective

To create public notes for the call

- 1 Navigate to the Contact Call screen.
- 2 Select Profile from the View drop down list.
- 3 Tap the Notes tab.

The Pharma Handheld application displays the notes list specific to the call.

- 4 Select New Record from the menu and create a note.

Changing Contacts While in Signature Capture View

Sales representatives might find themselves in a position where they need to switch contacts in order to get a required signature. You can change a contact within the signature capture screen.

To change contact while in Signature Capture view

- 1 In signature capture screen, tap the Change button.
- 2 From the Pick Contact list, select the desired contact.

NOTE: By default only the contacts that are affiliated to the accounts this contact is affiliated to will show on the pick applet. Users can select All Contacts button to show all contacts regardless of their affiliation

- 3 Tap Pick and Sign.

The signature capture view appears with the new contact information just selected.

- 4 Tap OK to commit the signature.

To submit a completed contact call

- 1 Navigate to the Contact Call screen, and select the desired contact call.

NOTE: You cannot submit a call with a future date.

- 2 Verify that the information for the Contact Call is accurate and complete. For more information about verifying contact call information, see:

- To record information about products detailed on page 247
- To record information about samples dropped on page 248
- To record information about promotional items dropped on page 249
- To record information about decision issues discussed during the call on page 249

NOTE: After you submit a call, you cannot modify any fields except the Comment field.

- 3 If you dropped samples when making the call, verify that you entered a samples reference number in the Ref # field in the Contact Call form.

CAUTION: If you are a mobile user, submit your calls before connecting to the server, and then synchronize. Mobile users must not submit calls while connected to the server or they run the risk of introducing errors into their inventory counts.

- 4 Contact Call form, tap Submit.

If the call passes all of the validations, the application:

- Creates a samples disbursement transaction with a line item for each dropped sample recorded for the call.
- Submits a disbursement transaction to update the samples inventory. If the submission of the disbursement transaction is successful, the application changes the call's Status field to Submitted.
- Sets appropriate fields in the call record to read-only.

For more information about submitting records, see ["How Siebel Samples Work with Calls" on page 262.](#)

Recording a Contact Call for a Different Contact at the Time of Signature Capture

A sales representative creates a call for a contact, but prior to saving the signature might need to choose a different contact or change the address of the call. The representative might need to create a call for another contact that they met while waiting for the first contact to arrive.

A representative can change an existing call and tap the Pick and Sign button to quickly navigate back to the Signature Capture view.

A representative can also create a new Call record based on an existing call directly from within the Contact Call view. The created Call records are displayed in the Contacts Call Detail view under Activities. The user is prompted to select the contact name, and tap the Pick and Sign button to capture the signature.

To create a call based on an existing contact call for a new contact

- 1 Navigate to the Contacts screen.
- 2 Select a Contact record from the list and tap New Call.

A new contact CALL record is created and the Contact Call Detail view is displayed.

- 3 Add sample drop records based on the personal list, or choose a Smart Call Template to prepare the sample drops, and tap the Sign button.

If all the business rules are valid, the signature capture window is launched.

- 4 Verify the contact's name, address, Drug Enforcement Agency (DEA) number, and license number displayed in the call header.

To change the contact or address of an existing call

- 1 Tap Cancel on Signature Capture view.
- 2 Tap the small icon on the Last Name field.
The Pick Contact applet is displayed. You can query All Contacts and Affiliated Contacts to find the relevant contacts.
- 3 Select a contact from the list, and tap the Pick and Sign button to navigate to the Signature Capture view.
- 4 Tap Change to display the Pick Contact applet.
- 5 Tap the small icon on the Address field.
- 6 Select the correct address from the list, and tap OK.
- 7 Tap Sign to navigate back to the Signature Capture view.

To create a new Sample Drop contact call based on an existing call

- 1 Tap Cancel on Signature Capture view.
- 2 Tap New Sample Drop in the Contact Call Detail View.
The Pick Contact Applet is displayed. You can query All Contacts, Affiliated Contacts, and By Location to find the relevant contacts.
- 3 Select a contact from the list, and tap the Pick and Sign button to navigate back to the Signature Capture view.
NOTE: Tapping Sample Drop copies the Sample Drop Applet records. Details, Promotional Items and Decision Issues are not copied.
- 4 Save the signature of the contact that decrements the inventory and creates sample transaction history records.

Creating Account Calls in Siebel Pharma Handheld

Account Calls are used to track a planned or past call activity at an account. You can enter account call details directly in Siebel Pharma Handheld applications. For more information about entering call details, see [“Recording Account Calls in Siebel Pharma Handheld Applications” on page 255](#).

To add an account call from the Accounts view

- 1 Navigate to the Accounts screen.

- 2 In the Accounts list, tap the New Call button and complete the fields in the new form. Some of the fields are described in the following table.

| Field | Comments |
|------------|--|
| Name | The account to be visited during this call. Make a selection from the Pick Account dialog box. |
| Site | A unique location identifier that distinguishes this account from any other accounts with the same name. |
| Smart Call | <p>The Smart Call template to be associated with this call. Make a selection from the Pick Smart Call dialog box.</p> <p>A Smart Call is a template that you apply to contact calls that defines information about the products detailed, samples and promotional items distributed, and issues discussed on a call. After you apply a Smart Call template to a call, you can make additional edits.</p> |
| Status | The status of this call. The default value is Planned. You cannot select the value Submitted. To change the status of the call to Submitted, you must submit the call. For more information about submitting a call, see "To submit a completed account call" on page 256 . |

- 3 From the Show drop-down list, select the following and add records, if necessary:

- Details
- Attendees

- 4 From the parent Account Call form, select the Submit button.

For more information about entering information in the Details and Attendees lists, and submitting a call, see ["Recording Account Calls in Siebel Pharma Handheld Applications" on page 255](#).

To validate the call

- 1 Navigate to Account Call Details screen.
- 2 Tap on the Validate button.

When the validation completes, a dialog either displays that there are no errors or displays all the errors from the underlying validation engine (LS Pharma Validation Service - CE).

To create private notes for the call

- 1 Navigate to Account Call screen.
- 2 Select Wrap-up from the View drop down list.
- 3 Tap the Call Notes tab.
- 4 Enter details to the following fields to make private notes for the calls:

- Comment
- Next Call D
- Next Call Objective

To create public notes for the call

- 1 Navigate to the Contact Call screen.
- 2 Select Profile from the View drop down list.
- 3 Tap the Notes tab.

The Pharma Handheld application displays the notes list specific to the call.

- 4 Select New Record from the menu and create a note.

To add an account call from the Account Call view

- 1 Navigate to the Account Call screen.
- 2 On the toolbar, tap the New Record button.
- 3 In the Account Call form, complete the necessary fields.

Recording Account Calls in Siebel Pharma Handheld Applications

Account calls are used to track a planned or past call activity with an account. You can enter account call details directly in Siebel Pharma Handheld.

To record an account call, users must complete the following procedures:

- Enter information on products detailed.
- Enter attendee information (the names of contacts met and any samples dropped).
- Enter information about any activities related to the call (optional).
- Submit the call.

The next topics explain how to perform each of these tasks.

To enter products detailed information for an account call

- 1 Navigate to the Account Call screen, and select the desired account call.
- 2 From the View drop-down list, select Execute.
- 3 Tap the Details tab.

- 4 In the Details list, complete the necessary fields. Some of the fields are described in the following table.

| Field | Comments |
|------------|--|
| Indication | The purpose or therapeutic area for which a drug is designed. |
| Priority | Priority of this product relative to others discussed during the call. |
| Name | The product detailed during the call. The products appearing in this drop-down list are based on the products added to your Personal List. |

To enter attendee information for an account call

- 1 Navigate to the Account Call screen, and select the desired account call.
- 2 From the View drop-down list, select Execute.
- 3 Tap the Attendee tab.
- 4 On the toolbar, tap the New Record button to add a new record for each attendee.
NOTE: By default, the pick list shows only contacts that are affiliated to the chosen Account. To show all contacts regardless of affiliation, tap the All Contacts button in the Pick Contact list.
- 5 Complete the fields for each attendee.
 - a In the Last Name field, tap the drop-down list arrow.
 - b From the Pick Contact dialog box, select those contacts you want to add as attendees.

The application creates an attendee call record (similar to a contact call) for each contact you include as an attendee of the account call. These attendee call records are child records of the account call and can be accessed by going to the Account Call and finding the associated attendee. Drilling down on the attendee last name hyperlink enable you to navigate to and access the appropriate attendee call details view.
- 6 To create an attendee call, do one of the following:
 - Drill down on the Last Name hyperlink and complete the fields in the Contact Calls details.

To submit a completed account call

- 1 Navigate to the Account Call screen, and select the desired account call.
NOTE: You cannot submit a call with a future date.
- 2 Verify that the information for the Account Call is accurate and complete. For more information about verifying account call information, see:
 - [To enter products detailed information for an account call on page 255](#)
 - [To enter attendee information for an account call on page 256](#)
CAUTION: If you are a mobile user, submit your calls before connecting to the server, and then synchronize. Mobile users must not submit calls while connected to the server. If they do, they run the risk of introducing errors into their inventory counts.

- 3 In the Account Call form, select Submit.

If the call passes all of the validations, the application:

- a Creates a samples disbursement transaction with a line item for each dropped sample recorded for the call.
- b Submits a disbursement transaction to update the samples inventory. If the submission of the disbursement transaction is successful, the application changes the call's Status field to Submitted.
- c Sets many fields in the call record to read-only.

NOTE: If attendees are recorded in an account call, an attendee call is created for each attendee, and the process described above is performed for each attendee call. Attendee calls are the same as contact calls except that they do not appear in the Activities view and they are submitted automatically when the account call is submitted.

Creating General Activities in Siebel Pharma Handheld Applications

Sales representatives use general activities to manage time and schedule demands and to keep their manager informed of their workload. A general activity in Siebel Pharma Handheld is used to track non-call tasks such as to-do items, personal events, and correspondence.

In a typical scenario, a sales manager for a pharmaceutical company uses the Activities screen to create and track the activities on which they plan to spend their time during the next month. After scheduling a contact call with a customer, the sales manager uses the Calendar to review scheduled activities for the upcoming weeks.

The following procedures for adding activities from the Activities and Calendar views are described in this topic:

- [Adding General Activities from Activities in Siebel Pharma Handheld on page 257](#)
- [Adding General Activities Using Calendar in Siebel Pharma Handheld on page 258](#)

Adding General Activities from Activities in Siebel Pharma Handheld

All activities with valid dates and times appear in the Calendar views of Siebel Pharma Handheld as well as in the Activities views. The Calendar views provide a visual representation of each user's scheduled activities.

NOTE: The Planned Start and Duration fields in general activities and appointments must contain entries in order for the appointments and activities to appear in the correct time period in the Daily Calendar view. While the Planned Start field is visible in the Activities screen's My Activities view, this field is not visible in the Contacts screen's Activities and Accounts > Activities views. The Planned Start field might be exposed during application configuration.

To add a general activity from the Activities view

- 1 Navigate to the Activities screen.
The My Activities list appears.
- 2 On the toolbar, tap the New Record button.
- 3 In the new row, complete the necessary fields. Some of the fields are described in the following table.

NOTE: Select **More Info** from the **Show** drop-down list to edit additional fields for each record.

| Field | Comments |
|---------------|---|
| Activity | Activities entered here are set as "General" and are not editable. Calls created from the New Call button are set as "Contact Call" or "Account Call" depending on whether these calls were created for a contact or an account. Previously created Contact Calls and Account Calls are displayed on the Activities view. |
| Activity Type | The type of activity. The values displayed in this drop-down list depend on the value of the Activity field. For example, the activity types that you can select for a contact call are different from those that you can select for a general activity. |
| Assigned To | The sales representative responsible for the activity. The Assigned To field in all Activities views (Activities, My Activities list; Activities, More Info form; Contacts, Activities list; and Accounts, Activities list) is read-only. |
| Duration | The length of time, in minutes, that you expect the activity to take. |
| Last Name | The last name of the contact associated with the activity. The Last Name field on the Activities, My Activities view and the Accounts, Activities view is read-only. |

Adding General Activities Using Calendar in Siebel Pharma Handheld

General activities and appointments must have the Planned Start and Duration fields populated in order to be displayed in the correct time slot on the Daily Calendar view.

To add a general activity in the Calendar view

- 1 Navigate to the Calendar screen.
- 2 From the View drop-down list, select Daily, Weekly, or Monthly.
- 3 For Weekly or Monthly, select the day and time on which you want to schedule a general activity.
- 4 On the toolbar, tap the New Record button or tap twice on a row.

- 5 In the new record, complete the necessary fields. Some of the fields are described in the following table.

| Field | Comments |
|---------------|---|
| Account | Complete this field if the activity is related to a particular account. |
| Planned Start | If you started from the Daily Calendar view, this date reflects the time selected when you created the new appointment. If you started from the Weekly Calendar or Monthly Calendar view, the first time slot of the day is supplied. |
| Type | The type of activity. The values displayed in this drop-down depend on the kind of activity. |
| Repeat | Checking this box allows you to set the end date of a repeating appointment (Until). If you do not select a value, 12 more appointments will be scheduled. |

- 6 To specify contact participants:
- a From the Show drop-down list, select Contact Participants.
 - b Tap the Contacts list to select it.
 - c On the toolbar, tap the New Record button.
 - d In the Add Contacts dialog box, select participants and then tap OK.
- NOTE:** You can also create a new contact from the Add Contacts dialog. The newly created contact gets updated on you contacts list.
- 7 To specify employee participants:
- a From the Show drop-down list, select Employee Participants.
 - b Tap the Employee Participants list to select it.
 - c On the toolbar, tap the New Record button.
 - d In the Add Employees dialog box, select participants, and then tap OK.

Managing Samples in Siebel Pharma Handheld Applications

Most pharmaceutical manufacturers provide free samples to health care professionals in an effort to influence the prescribing habits of physicians and accounts across the United States and the world. In the United States, samples distribution is regulated by the FDA and must be reconciled at least once a year.

Siebel Pharma Handheld maintains a samples transaction record for samples transfer, receipt, disbursement, order, and inventory adjustment. The application assigns each samples transaction record a unique identifier for tracking purposes. Depending on the type of transaction, you can display this identifier as a transaction number, an order number, an invoice number, or a transfer number.

This topic describes managing and maintaining an electronic inventory of samples and promotional items within Siebel Pharma Handheld. As part of managing samples, you can track an electronic inventory of samples stock, create request orders and transfers for samples, and acknowledge receipt of sample transfers directly in the Handheld application. You can use either the Siebel Pharma application or the Siebel Pharma Handheld application to manage your samples, however you cannot use both applications to handle samples transactions. Reconciling inventory must be performed in the Siebel Pharma application.

The following information about managing samples in Siebel Pharma Handheld applications is included in this topic.

- [“Scenario for Managing Samples in Siebel Pharma Handheld” on page 260](#)
- [“Keeping the Samples Records in the Server and the Siebel Pharma Handheld Application Synchronized” on page 261](#)
- [“How Siebel Samples Work with Calls” on page 262](#)
- [“About Inventory Periods in Siebel Pharma Handheld” on page 263](#)
- [“About Establishing an Initial Inventory Count in Siebel Pharma Handheld” on page 263](#)
- [“About Samples Views in Siebel Pharma Handheld” on page 266](#)
- [“Creating a Samples Order in Siebel Pharma Handheld” on page 267](#)
- [“Creating a Samples Transfer in Siebel Pharma Handheld” on page 268](#)
- [“Acknowledging Receipt of a Samples Transfer in Siebel Pharma Handheld” on page 269](#)
- [“Adjusting a Samples Transaction in Siebel Pharma Handheld” on page 270](#)

Scenario for Managing Samples in Siebel Pharma Handheld

This topic gives one example of how you might manage samples in Siebel Pharma Handheld applications. You might manage samples differently, depending on your business model.

Near the end of the quarter, a sales representative in the U.S. prepares for FDA-mandated inventory reconciliation. The sales representative is required to take a physical inventory of their samples and make sure the quantities match the electronic inventory. To do so, they need to make sure that all of their sample transactions are up-to-date for the current period before submitting the inventory count.

The sales representative remembers that they made a sample disbursement mistake earlier in the day. They accidentally had a physician sign for 10 samples when really only five were dropped off. Therefore, the sales representative needs to make a samples adjustment for this transaction.

In Siebel Pharma Handheld, the sales representative finds the sample transaction record. Then, the representative selects the adjust quantity feature, which automatically shifts the transaction status to In Progress for the new adjustment record and to Adjusted for the already-submitted sample transaction. When the representative submits the adjustment record, Siebel Pharma Handheld updates their inventory to reflect the lower quantity that was disbursed and changes the sample quantity for the actual call from ten to five.

The sales representative also had some samples that had expired, which they physically mailed back to their samples administrator. The representative now has to record this transaction into their electronic inventory. In the Samples Transfer view, the representative creates a new record, then adds three new sample line item records, and selects the Submit button. This changes the status to submitted, locks the records, and updates the inventory for the Siebel Handheld application to reflect the transfer.

The sales representative is now ready for the actual inventory count. While physically standing in their personal samples closet with their handheld device, the representative opens the Samples Inventory Count view. From this view, they see the products listed in their personal inventory according to the electronic count. They go down the list, product by product, lot by lot, counting each product and lot number combination.

The sales representative enters the actual quantity that they find into Siebel Pharma Handheld (using their handheld device) and notices that the application automatically updates the difference field to reflect discrepancies between the on-hand and electronic inventory. When finished, the representative verifies that the differences are zero. The representative is now ready to submit their inventory count. Submitting causes a new inventory period to open, displaying the current date and time.

Keeping the Samples Records in the Server and the Siebel Pharma Handheld Application Synchronized

Administrators can manage a samples inventory using either the Siebel Pharma Handheld application or the Siebel Pharma application, but must not use both. In order to keep sample inventories synchronized, users must not submit a samples transaction on the server when the same record is still editable on their handheld device.

When a user submits a record, the status changes to Submitted, and it becomes read-only. Read-only means that no additional operations are allowed on the record. By updating a user property and writing two small Visual Basic scripts, you can specify that synchronized records are also read-only.

To configure synchronized records as read-only

- 1 First, the administrator changes the value of the user property Update Status To Synchronized to Y.
Making this change invokes the UpdateStatusToSynchronized method on the Sample Txn business component.
- 2 Then, the administrator writes a small Visual Basic script for this method to change the status on the Sample Txn to Synchronized.

- 3 Finally, the administrator writes another Visual Basic script to specify that the record becomes read-only when its status changes to Synchronized.

This prevents users from accidentally submitting any sample transactions on the server while the same record is still editable on their handheld device.

NOTE: Sample transactions are not logged upon submitting a sample drop. They only appear on the handheld device after synchronization has been carried out.

How Siebel Samples Work with Calls

This topic describes how Siebel Samples work with the Contact Call Detail and Account Call Detail views. For more information about recording calls, see [“To view sample requests allocated” on page 240](#).

NOTE: If the Disburse on Signature Capture system parameter is set to TRUE, the application decrements the inventory count for Samples Dropped only when the user taps the Save button on Signature Capture view. If samples were dropped, tapping the Submit button creates a Samples Transaction History record only if the Disburse on Signature Capture system parameter is set to FALSE.

When you select the Submit button in either the Contact Call and Account Call forms, the application verifies that:

- The call's date and time is the same as or earlier than the current date and time. A call cannot be submitted with a future date.

If dropped samples are recorded for the call, the application checks that:

- The Ref# field is not empty. This is controlled by the user properties setting. If the user property for reference number required is set to Y, then the application displays an error message if this field is not populated by the user. Otherwise, the application will automatically populate this field.
- The samples exist in the inventory for the period indicated by the call's date and time.
- The period into which the samples will be recorded is an unreconciled period. If it is a reconciled period, a warning message appears.
- A valid lot number has been specified for dropped samples tracked by lot number.

If the call passes these checks, the application:

- 1 Creates a samples disbursement transaction with a line item for each dropped sample recorded for the call.
- 2 Submits a disbursement transaction to update the samples inventory. If the submission of the disbursement transaction is successful, the application changes the call's Status field to Submitted.
- 3 If sample are requested (direct send), a new Disbursement Request transaction is created.

- 4 Converts many fields in the call record to read-only.

An administrator can change the fields that become read-only after a call is submitted.

NOTE: If attendees are recorded in an account call, an attendee call is created for each attendee, and the process described in the previous paragraph is performed for each attendee call. Attendee calls are the same as contact calls except that they do not appear in the Activities view and they are submitted automatically when the account call is submitted.

About Inventory Periods in Siebel Pharma Handheld

Siebel Samples provide three types of inventory periods for maximum flexibility. These period types are described in [Table 34](#).

Table 34. Samples Inventory Periods

| Period Type | Description |
|-------------|---|
| Active | The current period; it shows no end date and it is not reconciled. You can enter any type of samples transaction, including adding new products; inventory counts will be adjusted accordingly. |
| Inactive | A past, unreconciled period; it shows an end date but has not been reconciled yet. Except for adding new products into inventory, you can enter any type of samples transaction. |
| Reconciled | A past, inactive period; reconciled periods will not display on the Handheld application. |

About Establishing an Initial Inventory Count in Siebel Pharma Handheld

Establishing an initial master inventory is only necessary if you are manually entering and tracking samples. Either an administrator or a user might establish an inventory. However, because Siebel Samples route and maintains transactions by Employee ID, only the user who creates an inventory (the initial inventory count) can create or manipulate that inventory.

Before adding products to the master inventory, you must verify that the products are correctly defined for inventory tracking. In particular, make sure that the products are Orderable and have either Inventory or Lot # tracking.

NOTE: Users do not need to perform the procedures described in this topic if their initial counts consist only of samples transfers routed to them by their samples administrator. Instead, they must simply acknowledge the receipt of those shipments. For more information about acknowledging receipts, see [“To acknowledge receipt of a samples transfer” on page 270](#). However, users do need to perform the procedures described in this topic if their initial counts contain any records that they entered manually. Users who enter records manually must submit those counts and reconcile the initial inventory period.

To establish an initial master inventory:

- **Count products on hand.** Establish an inventory count by physically counting the products currently on hand. This process creates a list of the products in your inventory and an inventory period solely for your initial count.
- **Add products and submit count.** Add products to the inventory and submit an initial count. (See [“To add products to inventory and submit an initial closeout count”](#) on page 264.)
- **Submit adjustments.** Create and submit adjustments for the initial count. For more information about creating and submitting adjustments, see [“To create and submit initial count adjustments”](#) on page 265.
- **Reconcile.** Reconcile the initial inventory period using the Siebel Pharma application. For more information about reconciling the initial inventory, see *Siebel Life Sciences Guide*.

To add products to inventory and submit an initial closeout count

- 1 Navigate to the Inventory screen.

The Inventory Count list appears.

- 2 From the View drop-down list, select Count View.

The current active period is displayed.

- 3 Tap the Inventory Count tab.

All the samples for the active period are displayed. The table has the following columns

| Field | Comments |
|------------------------|---|
| Current Physical Count | Number of samples found at the time of performing the physical count. |
| Count | The amount of your initial inventory. |
| Cut Off Date | Cut off date for the lot number. |
| Difference | Shows a negative number after you enter a value in the Count field. |
| Last Physical Count | Number of samples found when the previous count was conducted. |
| Lot | The lot number for this sample. |
| Sample | The name of the sample to be added. |

- 4 (Optional). On the toolbar, tap the New Record button to add a new row and select a different sample.

NOTE: You cannot sort on the Difference and On Hand Qty fields in the Inventory Count applet. You can conduct an interim count periodically. The main difference between Interim and Closeout is that an Interim count does not close the period. An Interim count ensures that huge difference do not exist when you close a period by performing a Closeout count.

5 Tap Closeout.

The Difference field is cleared for all records. The application transfers the number you entered in the Count field to the On Hand Quantity field and clears the Count field.

Once you submit an initial count, the Siebel Samples deactivates the inventory period into which you entered your initial counts and creates a new active inventory period. You can view this change using the My Samples History filter.

To view an inventory audit report

- 1 Navigate to the Inventory screen.
- 2 Select the Inventory Audit option from the drop down list.
- 3 View the existing reports.

You cannot create a new report. The application automatically creates these reports when an Interim or a Closeout count is performed.

- 4 (Optional) Tap on the Inventory Count tab to view the samples and their count submitted as part of the report.

To view and create a new location audit report

- 1 Navigate to the Inventory screen.
- 2 Select the Location Audit option from the drop down list.
- 3 View the existing reports.

You can create a new report.

- 4 (Optional) Tap on the Inventory Count tab to view the samples and their count submitted as part of the report.

NOTE: Location audit reports are periodically done by the sales representative to ensure that the storage location for the samples is safe and secure.

To create and submit initial count adjustments

- 1 Navigate to the Samples screen.
- 2 From the drop-down list, select Adjustments.
The Samples Adjustments list appears.
- 3 On the toolbar, tap the New Record button add a new record.
The Sample Adjustment Details form and list appear.

- 4 Complete the necessary fields. Some of the fields are described in the following table.

| Field | Comments |
|-------------|---|
| Txn. # | Transaction Number. The application automatically generates an unique identifier. You can change this value. |
| Txn. Date | Transaction Date. The date of the adjustment transaction. Make sure that this date falls within the valid date ranges for the appropriate Inventory Period. |
| Adj. Reason | Adjustment Reason. Select Initial Count from the drop-down list. |
| Status | The application automatically changes this field value to Submitted when you submit the adjustment. |

- 5 Tap the Details tab.

- 6 On the toolbar, tap the New Record button.

- 7 Complete the necessary fields.

Repeat the previous steps to enter a line item for each sample needing adjustment.

- 8 Tap Submit.

The values in the Transaction Status field (in the Samples Adjustment form) and the Item Status fields (in the Line Items list) change to Submitted.

NOTE: You must reconcile adjustments using the Siebel Pharma application. For more information about reconciling adjustments, see *Siebel Life Sciences Guide*.

About Samples Views in Siebel Pharma Handheld

Siebel Pharma Handheld contains these main screens related to samples:

- **Samples.** Navigate to this screen to drill down further into the Sample History view.
- **Sample Management.** Navigate to this screen to adjust sample transaction records, such as correcting quantities dropped off, or to display your on-hand inventory count. This screen has four views as follows:
 - **Received Samples:** Navigate to this view to acknowledge the receipt of a samples transfer or to display a list of receipts, and to review receipt details.
 - **Sent Samples:** Navigate to this view to create a transfer, to display a list of transfers to or from your inventory, and to review transfer details.
 - **Disbursements:** Navigate to this view to display samples involved in all the Sample Drop transactions, and to make adjustments to them.
 - **Adjustments:** Navigate to this view to make adjustments such as Found, Lost, Initial Count, or Request.
- **Samples Receipts.** Navigate to this screen to acknowledge the receipt of a samples transfer or to display a list of receipts, and to review receipt details.

- **Samples Transfers.** Navigate to this screen to create a transfer, to display a list of transfers to or from your inventory, and to review transfer details.
- **Samples Orders.** Navigate to this screen to create an order for samples, to display a history of orders, and to review order details.

Creating a Samples Order in Siebel Pharma Handheld

You can request additional samples by creating a samples order using Siebel Pharma Handheld.

After synchronizing, the Siebel Pharma application routes the samples order to the samples administrator. The samples administrator then creates a samples transfer that is routed back to the requestor. After the requestor acknowledges receipt of the transfer, Siebel Samples automatically enters the new inventory into their inventory counts.

To create a samples order

- 1 Navigate to the Samples Orders screen.

The My Sample Orders list appears.

- 2 On the toolbar, tap the New Record button.
- 3 In the new record, complete the necessary fields.

The application automatically sets the Order Date field to a date and time in the active period. It also automatically generates values for the Order # field, but it can be edited. You might want to edit this value to have the items appear in a particular order (other than the order in which you entered them).

NOTE: It is not necessary to change the Order Type field. You can include both samples and promotional items in a single order.

- 4 Tap on the Details tab.
- 5 On the toolbar, tap the New Record button.
- 6 Complete the necessary fields for the new record. Some of the fields are described in the following table.

| Field | Comments |
|-------------|--|
| Sample | The product to be ordered. Tap the arrow in the Sample field, choose the product from the Pick Product list, and then tap OK. |
| Quantity | The sample quantity that you are ordering. |
| Item Status | Initially set to In Progress. The application automatically changes this field value to Submitted when you submit the transaction. |

- 7 Repeat the previous steps to enter a line item for each product or promotional item you want to order, and tap the Submit button.

NOTE: You cannot modify or delete an order after it has been submitted.

The values of the Status field (in the form) and the Item Status field (in the list) change to Submitted.

Creating a Samples Transfer in Siebel Pharma Handheld

You can create a samples transfer to exchange samples between one another or to return samples to the home office.

After the samples transfer is submitted into the system, the recipient receives a samples transfer. After the recipient acknowledges the receipt of the transaction, the application enters the new inventory into their inventory count.

NOTE: Samples returns are defined as transfer transactions for which corresponding received shipment transactions have *not* been created. Returned samples are subtracted from the sender's inventory counts but are *not* added to the master inventory, as they are typically expired or damaged products.

To create a samples transfer

- 1 Navigate to the Samples screen.
- 2 Select Sent Samples from the drop-down list.
The Transfers list appears.
- 3 On the toolbar, tap the New Record button.
- 4 In the new row, complete the necessary fields. Some of the fields are described in the following table.

| Field | Comments |
|---------------|--|
| Transfer # | The application automatically generates a unique identifier. You can change this value. |
| Transfer Date | The application automatically generates a date and time value that falls in the active period. |

| Field | Comments |
|-------------------|--|
| Transfer Type | <p>Select the appropriate value:</p> <ul style="list-style-type: none"> ■ Transfer Out - Indicates you are transferring samples to another employee. ■ Return - Indicates you are returning samples to the home office. Samples are typically returned because they have expired or are damaged products that must be destroyed. Transfers with the Transfer Type field set to Return do not generate shipment transactions (because there is no recipient), and they are <i>not</i> added to the master inventory counts. |
| Status | The application automatically changes this field value to Submitted when you submit the transaction. |
| Transfer Lst Name | Transfer Last Name. The employee to whom you are transferring the samples. If the value of the Transfer Type field is Transfer Out, then you must complete this field. Leave this field empty if the value of the Transfer Type field is Return. |

- 5 Tap on the Details tab.
- 6 On the toolbar, select the New Record button.
- 7 In the new row, complete the necessary fields. Some of the fields are described in the following table.

| Field | Comments |
|-------------|--|
| Sample | The samples being transferred. |
| Quantity | The quantity that you are transferring. |
| Lot # | The lot number for this samples transfer. |
| Item Status | The application automatically changes this field value to Submitted when you submit the transaction. |

- 8 Repeat the previous steps to enter a line item for each product or promotional item you are transferring.
- 9 Tap Submit.

NOTE: You cannot modify or delete a transfer transaction after it has been submitted. The transaction will appear in the Samples History list as part of the Inventory List view.

Acknowledging Receipt of a Samples Transfer in Siebel Pharma Handheld

Siebel Samples supports two ways of electronically moving samples inventory:

- By creating a samples transfer (see [“Creating a Samples Transfer in Siebel Pharma Handheld” on page 268.](#))
- By creating a samples order (see [“Creating a Samples Order in Siebel Pharma Handheld” on page 267.](#))

In either case, the application creates a samples transfer transaction and routes it to the recipient for acknowledgement. After the recipient submits a receipt, the application transfers the new inventory into the recipient’s active inventory period. You can then view the new inventory in the Samples History view.

To acknowledge receipt of a samples transfer

- 1 Navigate to the Samples screen.
- 2 Select Received Samples from the drop-down list.
- 3 Select the incoming transaction and tap on the Transfer In Item List.
The Receipts Details view appears. In the form, samples transfers that have not been acknowledged as received show an Invoice Status value of In Progress. Those that have been acknowledged as received show an Invoice Status value of Submitted.
- 4 In the Transfer In Item list, enter values for the Received quantity.
- 5 Navigate back to the main view (received transactions list) and tap on Acknowledge.
If the value is the same as the value shipped, then, the status is either Submitted (if approval process is enabled), or Processed.
If the value is different from the shipped value, the application displays a message that there is a difference in the count. You can either tap Cancel to keep the transaction open or tap OK to mark the transaction as Processed With Discrepancies.
- 6 To verify the submitted transfer was recorded:
 - a Navigate to the Samples screen.
 - b From the Show drop-down list, select Samples History.
The Samples History view appears.
 - c Highlight the appropriate inventory period to view the line items of the received transfers.

Adjusting a Samples Transaction in Siebel Pharma Handheld

You can adjust inventory counts of samples stock by making adjustments to past samples transactions directly in Siebel Pharma Handheld. Common reasons for adjusting past transactions include product loss or expiration, counting mistakes, or data entry errors. Users can add products or adjust quantities for any type of samples transaction (disbursement, transfer, order, and so on).

NOTE: Users cannot add a product to your inventory that did not exist when the original transaction was created.

The inventory list view displays all the samples for the selected inventory period.

To adjust a quantity in a samples transaction

- 1 Navigate to the Samples screen.
The Samples History screen appears.
- 2 In the Inventory Periods list, select the period containing the transaction.
The samples for the selected period are displayed in the list below.
- 3 In the Samples History list, select the transaction you would like to update.
You can identify the transaction record in the Samples History list by the Transaction Date, Last Name, or Transaction # field values.
- 4 Tap Adjust.
The application copies the line-item record for the transaction in the Samples History list and sets the line-item record's Item Status field to In Progress.
- 5 In the Samples History list, enter the correct quantity in the Quantity field.
- 6 Tap Submit, and then tap OK.

The application adds the new item with an Item Status of Submitted and changes the original item record (whose quantity you adjusted) to an Item Status of Adjusted. The new quantity is reflected in the on-hand quantities in the Inventory Count view.

Setting Up MedEd Events in Siebel Pharma Handheld Applications

This topic describes how a pharmaceutical company can use Siebel Pharma Handheld to plan a medical education (MedEd) event. A MedEd event refers to a promotional or educational event attended by physicians and medical professionals. An event functions as a forum for exchanging information on a particular disease, illness, therapeutic class or product. A medical education event can be as simple as a lunch-and-learn in a physician's office or as complex as a seminar series or national sales meeting.

Sales representatives in the field can use Siebel Pharma Handheld to view and create MedEd events and invite contacts to MedEd events.

The following topics provide instructions for MedEd procedures.

- [Creating a MedEd Event in Siebel Pharma Handheld on page 271](#)
- [Inviting Professionals to MedEd Events Using Siebel Pharma Handheld on page 272](#)

Creating a MedEd Event in Siebel Pharma Handheld

You can create MedEd events directly in Siebel Pharma Handheld.

To create a MedEd event

- 1 Navigate to the MedEd screen.
- 2 From the Show drop-down list, select My MedEd Events.
- 3 On the toolbar, select the New Record button.
- 4 In the new list, complete the necessary fields. Some of the fields are described in the following table.

| Field | Comments |
|--------------|---|
| Budget | The amount of money budgeted for the event. |
| CME Credit | Number of CME (Continued Medical Education) credits that can be earned at the event. |
| Created By | User name for the person who created the MedEd event. |
| End Date | Date and time that the event ends. Set the time fields before selecting a calendar day. |
| Start Date | Date and time that the event starts. Set the time fields before selecting a calendar day. |
| Event Status | Indicates whether the event is active, cancelled, completed, in progress, inactive, or planned. |
| Type | The category of medical education event being offered. |

Inviting Professionals to MedEd Events Using Siebel Pharma Handheld

When you have scheduled a MedEd event and have decided what sessions will be offered, you can invite health care professionals to attend.

To invite a contact to a MedEd event

- 1 Navigate to the MedEd screen.
- 2 From the Show drop-down list, select Invitees.
- 3 Tap the Contacts list to select it.
- 4 On the toolbar, select the New Record button.
- 5 In the new row, complete the fields. Some of the fields are described in the following table.

| Field | Comments |
|------------|---|
| First Name | Completed based on the last Name selection. |
| Last Name | The contact/invitee. Select from the Pick Contact list. |

| Field | Comments |
|----------------|---|
| Invitee Status | Indicates whether the invitee has confirmed attendance, is on a wait list, and so on. Pending is the default value. |
| Role | Describes the selected contact's role at the session. |

Creating Expense Reports in Siebel Pharma Handheld Applications

You can track expenses and create expense reports using Siebel Pharma Handheld.

To create an expense report

- 1 Navigate to the Expense Reports screen.
- 2 From the Show drop-down list, select My Expense Reports.
- 3 On the toolbar, tap the New Record button.
- 4 In the new row, complete the necessary fields.
- 5 From the Show drop-down list, select More Info to view additional fields.

NOTE: You must complete the Submit To field or you will get an error message and your expense report will not be saved.

To add detailed line items

- 1 Navigate to the Expense Reports screen.
- 2 From the Show drop-down list, select My Expense Reports.
- 3 In the Expense Reports list, drill down on the Name hyperlink.
- 4 Tap the Line Items list to select it.
- 5 On the toolbar, tap the New Record button.
- 6 In the new row, complete the necessary fields.

You can enter Line Items whose Start or End Date is outside of the Expense Report period.

- 7 Repeat Step 4 through Step 6 to add line items.

A Troubleshooting

This appendix addresses some of the common issues—including installation, logging, and performance—faced while deploying a Siebel Handheld application. The audience for this appendix is Siebel database administrators, system administrators, and application administrators. The following topics are covered in this appendix:

- [“Troubleshooting Siebel Handheld Application Installation” on page 275](#)
- [“Troubleshooting Siebel Handheld Application Synchronization” on page 276](#)
- [“Siebel Handheld Application Log Files” on page 279](#)
- [“Using SQLTrace with Siebel Handheld Applications” on page 282](#)
- [“End User Error Messages” on page 283](#)
- [“Web Server Timeout Errors” on page 283](#)
- [“Handheld Device Timeout Errors” on page 283](#)
- [“Siebel Handheld Application Performance” on page 284](#)

Troubleshooting Siebel Handheld Application Installation

Siebel Handheld applications support a specific and limited set of devices, Handheld application OS, desktop OS, server OS, server databases, and languages. This range of devices and platform support varies widely across the range of Siebel Handheld application releases and must be carefully checked to confirm support. For information about supported handheld devices, operating systems, desktop OS, server OS, and server databases, see Siebel System Requirements and Supported Platforms on Oracle Technology Network for your Siebel application.

Client Installation, Backup, and Restore

If you encounter problems while installing the Siebel Handheld application on client devices from a desktop PC, refer to the following checklist. Most installation problems are covered by this checklist.

- Confirm that the handheld device is certified and supported. Refer to Siebel System Requirements and Supported Platforms on Oracle Technology Network.
- Confirm that the handheld device has 64 megabytes (MB) or higher of random access memory (RAM) with the majority of it available for the Siebel Handheld application.
- Make sure an ActiveSync connection is running (green icon). If not, check your computer to make sure that the serial or USB port is activated and functioning correctly.

- If a previous version of Siebel Handheld application is installed on the device, make sure you uninstall the application. For more information about uninstalling applications, see [“Uninstalling the Siebel Handheld Application” on page 201](#).

The binaries for the Siebel Handheld application can also be installed to a CompactFlash card to conserve (RAM) memory. The application will still be loaded into memory when it executes. To install on the CompactFlash card, set the ApplicationBinariesLocation and ApplicationDataLocation parameters in the setup.ini file. For more information about installing on a CompactFlash card, see [“Setting Up Siebel Handheld Application Installation on a CompactFlash Card” on page 182](#).

The Siebel Handheld application only (no data) can also be copied onto a CompactFlash card to aid in deployments. The initial backup image must be made of a device that has not done an initial synchronization. An initial synchronization will still be required by each user to receive the configuration and data. For information on backup utilities, see the documentation accompanying your handheld device.

It is also possible to back up your Siebel Handheld application on a PC using the Backup/Restore feature within ActiveSync.

Troubleshooting Siebel Handheld Application Synchronization

Synchronization problems can be caused by incorrect settings in the Siebel Handheld application setup.ini or .cfg files. Be sure that application names, data sources, and .srf files are correct.

Make sure to set the SyncURL setting in the client setup.ini file for all synchronization methods. Also, be sure to grant the user visibility to these views in the Administration - Application screen, then the Responsibilities view.

NOTE: Check the log files and confirm the accuracy of the generated connect string.

To troubleshoot synchronization for your Siebel Handheld application, review the following topics:

- [“Companion Synchronization Checklist” on page 276](#)
- [“Direct Server Synchronization Checklist” on page 277](#)
- [“Resynchronizing Problems” on page 278](#)
- [“Synchronization Timeout Parameters” on page 278](#)

Companion Synchronization Checklist

Use the following list to troubleshoot problems you encounter using Companion Synchronization.

- Is the latest version of ActiveSync (version 3.5 or later) installed and running?
- Is ActiveSync connected (green icon)?
- Do you have the correct data source defined in the .cfg file of your Siebel Handheld application?

- If you have problems logging in, check to make sure that you can use the same credentials to log into the desktop client.
- Remember to keep a 1:1 ratio of users to devices. Only a single user can log in to synchronize each device. Failure to do so causes data corruption.

When using a SRF with a custom configuration and a synchronization error occurs during initialization, follow the steps below to troubleshoot problems:

To troubleshoot Companion Synchronization configuration errors

- 1 Add HandheldSyncLogLevel = 5 to the [pda] section in the .cfg and look for more error details in the CompSync log.
- 2 Use the Validate function in Siebel Tools to look for a configuration error.
- 3 Remove Siebel Handheld application views from user responsibility until synchronization passes the error point.
- 4 Remove DefaultBusObjs from the Administration - Mobile screen, Application Administration, then the Settings view until synchronization passes the error point.

Direct Server Synchronization Checklist

Use the following checklist to troubleshoot problems you encounter using Direct Server Sync.

- Did you install all client and server components successfully?
- Is there a PDA Object Manager (OM) running on the Siebel Server?
- Did you bounce the IIS, SWE, and Siebel Server after installation?
- Do you have network access to the URL for DSS? Can you ping the IIS server from inside and, if applicable, from outside the firewall?
- Can you successfully run a SWE command?
- Do you have the correct SyncURL defined in the setup.ini file?
- Is the network connection working from the device? Try Pocket Internet Explorer.
- If you have problems logging in, check to make sure that you can use the same credentials to log into the desktop client.
- Remember to keep a one to one ratio of users to devices. Only a single user can log in to synchronize each device. Failure to do so causes data corruption.

When using a SRF with a custom configuration and a synchronization error occurs during initialization, follow the steps below to troubleshoot problems:

To troubleshoot Direct Server Synchronization configuration errors

- 1 Turn on synchronization logging for your Siebel Handheld application to 5 and look for more error details in the Synchronization OM log of your Siebel Handheld application.

For more information about logging, see [“Configuring Logging Levels for Direct Server Synchronization” on page 171](#).

- 2 Use the Validate function in Siebel Tools to look for a configuration error.
- 3 Remove the Siebel Handheld application views from user responsibility until synchronization passes the error point.
- 4 Remove DefaultBusObjs from the Administration - Mobile screen, Application Administration, then the Settings view until synchronization passes the error point.

Resynchronizing Problems

Generally if, for whatever reason, synchronization is terminated, the user is prompted with a message and the synchronization process ends. In this situation the synchronization session is allowed to close. However, if the connection with the server is unintentionally broken, the session might not have a chance to close before the connection is broken. Therefore, if the user tries to immediately synchronize again, they might get an error message and be unable to synchronize. If this occurs, the user must wait for a period of time (generally, 15 minutes) to allow the session to time out and terminate. Once the session terminates, the user can synchronize again.

Synchronization Timeout Parameters

The following parameters control the timeouts for the synchronization process:

- Companion Synchronization timeout parameters are stored in the laptop registry under HKEY_CURRENT_USER, Software, Oracle, Siebel Handheld, and include the following:

SyncTimeout—This sets the timeout value for the entire synchronization process (from the handheld device to the laptop to the server and back again). The default value, 1800000 milliseconds, is 30 minutes.

RemoteTimeout—This is the length of time for the handheld device Syncmanager process to wait for a remote synchronization to complete, before timing out. The default value, 1800000 milliseconds, is 30 minutes.

RemoteTimeout can also be controlled from the setup.ini file. For more information about the setup.ini parameters, see [“Siebel Handheld Application setup.ini Parameters” on page 176](#).

NOTE: RemoteTimeout only applies to synchronizations using Companion Synchronization and when remote synchronization is enabled.

- Direct Server Synchronization timeout parameters are stored in the .cfg file and include:
 - IIS Timeout—This is the length of idle time between communications from a handheld device to the server, before a timeout occurs.
 - OM Timeout—An OM (object manager) timeout is triggered by an idle timeout on the server or any loss of communication with the OM for the defined period.
- For more information about changing these timeout settings, see [“To change the IIS timeout value” on page 279](#) and [“To change the OM timeout value” on page 279](#).

To change the IIS timeout value

- 1 Open IIS.
- 2 Go to Default Web Site, right click, and open Properties.
- 3 Change the value for the Connection Timeout setting.

The default value for Connection Timeout is 900000 milliseconds, or 15 minutes.

NOTE: This parameter only applies to synchronizations using Direct Server Sync.

To change the OM timeout value

- 1 Open the .cfg file for your handheld device.
- 2 Under the Siebel section, change the entry for SrvrTimeOutInterval.

The default value for SrvrTimeOutInterval is 600000 milliseconds, or 10 minutes.

NOTE: This parameter only applies to synchronizations using Direct Server Sync.

Siebel Handheld Application Log Files

This topic provides information about the different types of log and output files that are generated by your Siebel Handheld application. This topic includes the following information:

- [“Direct Server Synchronization Log and Output Files” on page 280](#)
- [“Companion Synchronization Log and Output Files” on page 280](#)
- [“Handheld Device Log” on page 281](#)

A new handheld synchronization log file, *hhsync.log*, is created on the first synchronization of each day, and when the synchronization node id changes in the Siebel Handheld application. The synchronization node id changes when you re-install or restart the Siebel Handheld application on your handheld device. Where multiple synchronization node folders are created under the user folder (for example, as follows in DSS), then this means that the synchronization node id has changed:

```
\Si ebsrvr\hhsync\phce\userName\syncNode
```

Direct Server Synchronization Log and Output Files

The name of the Direct Server Synchronization Log is the OM name plus the task ID for that log session. The log is found in the log subdirectory under the Siebel root directory. Depending on the amount of detail enabled, this file can give a good perspective of what has happened during synchronization regarding the Siebel OM.

To assist in service request (SR) resolution, collect the following files when logging a Siebel Handheld application service request. The following files are found on the machine where Siebel Server is installed:

- All server synchronization user files in the *Siebel Root\si ebsrvr\hhsync\application directory\user id\node id* directory.
- Server application event log extract.
- Server system event log extract.
- All files in the *Siebel Root\si ebsrvr\log* directory including CompName_task.log.

where CompName is the name of an application server component, and task is a Siebel task number. Each synchronization session is assigned a new Siebel task, so there will be many of these. An example CompName value is SalesCE_enu (note that language is part of the component name itself).

- All files from all subdirectories under the *Siebel Root\si ebsrvr\logarchive* directory.
- Server performance and monitoring log files (showing memory, network, disk performance), if available.

The following files are found on the machine where the Web Server (that is, IIS) is installed:

- SWE log files from *Siebel Root\appweb\log* (all files).
- IIS server application event log extract.
- IIS server system event log extract.
- IIS server performance and monitoring log files (showing memory, network, disk performance), if available.

For help with resolution, create a SR on *My Oracle Support*.

Companion Synchronization Log and Output Files

The name of the Companion Synchronization Log is compsync.log. It is found in the log subdirectory under the Siebel root directory. Depending on the amount of detail enabled, this file can give a good perspective of what has happened during synchronization regarding the Siebel object manager:

- *Siebel Root\client\hhsync\application directory\user id\node id* directory.
- NT application event log extract.
- NT system event log extract.
- All files in the *Siebel Root\client\log* directory.

- Performance and monitoring (Perfmon) log files (showing memory, network, disk performance), if available.

Handheld Device Log

The siebel_log.txt file is found in the /Program Files/Siebel Handheld directory on the handheld device. It is controlled by reg key DefaultLogLevel, with values 0-4. For more information about DefaultLogLevel, see [“Siebel Handheld Application setup.ini Parameters” on page 176](#).

The default value of the reg key during installation is 2. This file logs all of the following:

- Database objects that were imported during synchronization.
- All SQL statements generated by the application. You must set a parameter in the setup.ini file to turn this log on. DefaultLogLevel=3 and 4 will log SQL statements to the device log.

For more information about setting DefaultLogLevel, see [“Siebel Handheld Application setup.ini Parameters” on page 176](#).

- All system error messages. You must set a parameter in the setup.ini file to turn this log on. DefaultLogLevel=4 will give the most detailed log with debug info.

For more information about setting DefaultLogLevel, see [“Siebel Handheld Application setup.ini Parameters” on page 176](#).

- All of the messages displayed on the Sync Manager dialog box as well as some basic information about synchronization, such as data source and connect string. You can set the logging level for this file using the DefaultLogLevel parameter in the setup.ini file.

Logging Synchronization Errors for Siebel Handheld Applications

Synchronization for a Siebel Handheld application logs all the strings that the user sees in the synchronization message window, as well as many informational messages that do not get displayed in the UI. These informational messages are used to help track down where an error occurred.

Synchronization for a Siebel Handheld application determines which messages get logged based on a value stored in the registry under the key DefaultLogLevel. The values range from 0 to 4, where 0 logs all error messages and strings that the user sees, and 4 logs all data that is sent and received during synchronization.

If synchronization for your Siebel Handheld application fails, check the siebel_log.txt file immediately after the failure. If you need more detail, increase the logging level of DefaultLogLevel and run synchronization for the Siebel Handheld application again. For more information about changing DefaultLogLevel, see [“Siebel Handheld Application setup.ini Parameters” on page 176](#).

You can determine when and where an error occurs during synchronization of your Siebel Handheld application by referring to siebel_log.txt and the Sync Object Manager log.

The device log, siebel_log.txt, shows the last HTTP command that was sent to server, for example, ExecuteLogin, Init, ProcessTransactionEx, ExtractDB, and so on. The Sync Object Manager log receives the request and invokes the corresponding method.

For example, in the device log (siebel_log.txt), the following SWEEExtCmd=Init code indicates that the device is sending an initialization request to the Siebel Server:

```
INF00x6F46107E03/30/2005 16:58:27http://PPP_PEER:2323/phce/  
start.swe?SWEEExtSource=siebel_pda&SWEEExtCmd=Init&PatchLevel=7.7.2.2PDA%5b18356_1]  
8.471262E-309NU&SyncFormat=CEO&SyncNodeld=&TimeStamp=1112230707
```

```
INF00x6F46107E03/30/2005 16:59:20<Init ReturnCode="OK" Complete="100">
```

The server log (Sync Object Manager log, that is, CompNameObjMgr_enu_XXXXX.log), shows that the Init request is received by the server and the Init method is invoked:

```
ObjMgrBusServiceLogInvokeMethod402005-03-23 10:45:27Begin: Business Service  
'Handheld Synchronization' invoke method: 'Init' at 10c2bc68
```

```
HandheldSyncLogDebug502005-03-23 10:45:27Request Init received
```

(Where CompName is the server component name of the synchronization for the specific Siebel Handheld application.)

In addition to the device log, check the CompSync log in laptop (for Companion Synchronization) or SalesCEObjMgr_enu_XXXXX.log in Siebel Server (for DSS). For more information about changing logging levels, see [“Configuring Logging Levels for Direct Server Synchronization” on page 171](#) and [“Configuring Logging Levels for Companion Synchronization” on page 172](#).

Using SQLTrace with Siebel Handheld Applications

Use SQLTrace for diagnostic purposes if you encounter serious configuration errors or performance problems and cannot otherwise diagnose the problem. The problem can quickly become apparent in the log entries from SQL tracing.

NOTE: SQLTrace can also be enabled for Companion Synchronization too.

To enable SQL tracing

CAUTION: When SQL tracing is enabled, synchronization is severely and adversely affected. Therefore, it is recommended that you turn SQL tracing on for a limited time to diagnose a specific problem, then turn SQL tracing off.

- 1 Log into srvmgr prompt.
- 2 At the DOS prompt, run server manager (srvmgr) from the <siebsrvr/bi n> directory:

```
srvmgr /g gateway /e enterprise /u sadmin /p sadmin
```
- 3 Run: change evtloglvl ObjMgrSqlLog=4 for comp [OMName], where [OMName] = the OM name (for example, PhCEObjMgr_enu).
- 4 Log entries are written as described in [“Siebel Handheld Application Log Files” on page 279](#).

End User Error Messages

In addition to errors found in the siebel_log.txt file, you might receive a Transaction Errors Reported message during synchronization. These errors are logged in TXNError.hh. The message appears on the handheld device in the Syncmanager Status box. You can double-click the message at any time to see the errors. You can set the following settings in setup.ini:

- TXNErrorLevel = 0 (no log information)
- TXNErrorLevel = 1 (report txn errors in the client)

For more information about user messages, see [“Troubleshooting Synchronization with the Siebel Handheld Application Client” on page 219](#).

Web Server Timeout Errors

If synchronization times are long due to the size of the database file or the speed of the network connection, your users can encounter Web Server timeout errors and synchronization failures. First try to resolve this problem by reducing the size of your extract and configuring your server performance. You can reduce the size of the extract by fine-tuning filters, or if appropriate filters have already been defined, select ones that will return less data. If after doing this, your users continue to experience timeout errors, then increase the timeout parameter on IIS for the default Web Server.

Follow the procedure for your platform to change the timeout parameter.

To change the timeout parameter on Windows

- 1 From the Start menu, select Programs, Administrative Tools, then Internet Services Manager.
- 2 From the Internet Information Services window, select the local machine, right-click on the Default Web Site, and select Properties from the pop-up list.
- 3 From the Web Site Properties window, select the Web Site tab.
- 4 Under the Connections heading, locate the Connection Timeout property and increase the number of seconds.
- 5 Click OK.

Handheld Device Timeout Errors

The following three parameters are stored in the device registry. They control the device timeout. For more information about device timeout errors, see the following parameters in [“Siebel Handheld Application setup.ini Parameters” on page 176](#):

- InternetOptionReceiveTimeout = 600000
- InternetOptionSendTimeout = 600000
- InternetOptionDataExtractionTimeout = 600000

Siebel Handheld Application Performance

This topic discusses ways you can optimize your client performance, server performance, and network performance.

Client Performance

Client performance is affected primarily by the device capability (minimum 206 MHz processor and 64 MB RAM), configuration, and the quantity of data loaded onto the device. In general, screen-to-screen and view-to-view navigation times of 3 to 4 seconds or less are considered normal.

It is strongly recommended that you follow these general guidelines to optimize the performance of your Siebel Handheld applications:

- Keep configuration under 30 views.
- Keep RML file size under 1.75 MB.
- Keep database files (dbfile.txt) under 3 MB.
- Keep the number of records for each business component under 2,500.
- Avoid installing other applications on the device at the same time.
- Users must synchronize their data frequently.

Additional configuration information can be found in Bulletin 476964.1 on *My Oracle Support*. This document was previously published as Siebel Technical Note 405.

The Siebel Handheld Client benefits from periodically closing and restarting the Siebel Handheld application to release memory consumed by the operating system. Close the Siebel Handheld application from the application-level menu. For information about how to close the application, see the online help for the Siebel Handheld application.

If an external VGA driver is or has been loaded on the device in the past and is no longer needed, be sure to unload it, because it is known to have significant performance impacts on handheld devices.

Server Performance

Server performance affects the performance and scalability of Direct Server Synchronization processes. Customers can engage Expert Services to do a Scalability Review. The following guidelines are general in nature and pertain to the synchronization components of your Siebel Handheld application running on a Siebel Server:

- Use Performance Monitor to track memory and processor performance on the Siebel Server.
- Make sure that the Servers (IIS, GW, OM and DB) are on the same high speed LAN segment.
- The synchronization components of your Siebel Handheld application will stress Siebel Server and database performance because of the intensive interactions during large concurrent synchronizations.
- Estimate that 64 MB is used for each Siebmtsh.exe instance.

- To calculate the theoretical maximum concurrent users:

$$\text{Numusers} = \text{Server_Memory} * \text{threads_per_process_ratio} / 64 \text{ MB}$$
- Optimize the Threads for each Process Ratio (Max Tasks/Max MT Servers) according to Technical Note 405.
- Set Max Tasks=Max. number of concurrent users (for example, 150).
- Set MinMTServers=MaxMTServers (for example, 30 for 5:1 ratio, if this is determined to be appropriate).

To optimize and tune a Siebel DB server:

- Use Performance Monitor to track memory and processor performance on the DB server.
- Turn on SQL tracing and search for long-running SQL calls.
- Estimate 64 MB of RAM for each concurrent user.
- Consider creating indexes to tune the queries. For help with creating indexes to tune queries, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- Consider using a RAID 0+1 disk array with an expanded number of spindles to optimize database performance.

NOTE: Make sure that DB Server settings are set to recommended values. Set timeouts at the Siebel Server. Be careful that the timeout is set longer than the longest SQL query.

Network Performance

The condition of the network affects Direct Server Synchronization performance and scalability. In general, a low-latency, high-bandwidth network is preferred with dedicated rather than shared network connections. The following are general network considerations:

- Make sure there is adequate bandwidth throughout the entire round-trip of network packets and estimate where bottlenecks will occur due to high concurrent load. For example, if 50 users are concurrently synchronizing 1 MB each, 50 MB of data must pass through the network. If this data is passing through a single leased 56 KB/second line, then this data will minimally take $500,000 \text{ KB} / 56 \text{ KB/sec} = 8928 \text{ seconds} = 148 \text{ minutes}$. In reality, a 56 KB per second line will offer only 30-40 KB per second average throughput, which causes further bandwidth constraints. Check the total available bandwidth of network providers (private or public ISP).
- Verify that you have low latency lines with ping round trips of less than 1 second (1000 ms) with no dropped packets.
- Verify if roundtrips occur with less than 10 hops.
- Check the actual throughput of network connections by performing simple file transfers with no Siebel software.
- VPN software from third parties accounts for 30 to 60 percent of additional packet overhead because of the encryption.

Network problems can usually be resolved by working with network providers to size network connections and optimize network routing. Consider the use of burst networks, quality of service contracts, private networks, and dedicated modem banks directly into a high-speed LAN environment.

B

Screens and Views

The Siebel Handheld Client supports a subset of the screens and views supported by the Siebel Web Client. Other screens and views can be configured for your Siebel Handheld application if they are based on currently supported classes. The following topics are covered in this appendix:

- [“Screens and Views” on page 287](#)

Application developers and application administrators will find the information in this appendix useful.

NOTE: If you configure screens and views based on unsupported classes, you might see anomalous results. For information on how to determine the classes used by other screens and views, see *Siebel Object Types Reference*.

Screens and Views

The screens and views shipped with the Siebel Pharma Handheld sample application are listed in [Table 35](#).

To find the view name in Siebel Tools that corresponds to a view name on the UI, use the following procedure.

To find Tools name and UI name of a view

- 1 Launch Siebel Tools.
- 2 In the Object Explorer, select Screen.
The Screens list appears in the Object List Editor.
- 3 In the Object List Editor, query for the screen you want, and then select it.
- 4 In the Object Explorer, expand the Screen object, and then select the Screen View object.

The Screen Views list appears in the Object List Editor. The Name property shows the Tools name of the view, and the Viewbar Text property shows the UI name.

You can find the Viewbar Text property either by scrolling to the right in the Object List Editor or by looking in the Properties window when the view is selected. (If the Properties window is not visible, from the View menu, choose Windows > Properties Window.)

For more information about the Object Explorer, see the chapter that describes the Siebel Tools user interface in *Using Siebel Tools*.

Table 35. Siebel Pharma Handheld Screens and Views

| Screen | View Name on UI |
|------------|--------------------|
| Contacts | My Contacts |
| | More Info |
| | Private Notes |
| | Shared Notes |
| | License Numbers |
| | Activities |
| | Calls |
| | Affiliations |
| | Best Time to Call |
| | Addresses |
| | Rx Data |
| Accounts | My Accounts |
| | Shared Notes |
| | More Info |
| | Private Notes |
| | Activities |
| | Best Times to Call |
| | Addresses |
| | Calls |
| | Affiliations |
| | Sales Data |
| Activities | My Activities |
| | More Info |

Table 35. Siebel Pharma Handheld Screens and Views

| Screen | View Name on UI |
|---------------------|---|
| Calendar | Daily |
| | Weekly |
| | Monthly |
| | Contact Participants |
| | Employee Participants |
| | Add/Modify Activity (note: not displayed from the drop-down menu) |
| Contact Call | Details |
| | Promotional Items |
| | Issues |
| | Samples Dropped |
| Account Call | Details |
| | Attendees |
| Home Page | Home Page |
| Product Focus | Details |
| | Promotional Items |
| | Samples |
| Med Ed | My Med Ed Events |
| | Invitees |
| Expense Reports | My Expense Reports |
| | More Info |
| | Expense Items |
| Samples | Samples History |
| | Inventory List |
| Samples Adjustments | Inventory Count |
| | Adjustments |
| | Adj. Details |
| Samples Receipts | Receipts |
| | Receipt Details |
| Samples Transfers | Transfers |
| | Transfer Details |

Table 35. Siebel Pharma Handheld Screens and Views

| Screen | View Name on UI |
|----------------|-----------------|
| Samples Orders | Orders |
| | Order Details |

C

Business Components and Classes

The Siebel Handheld Client supports a subset of the business components and classes supported by the Siebel Web Client application. This appendix lists the supported business components, business component classes, and applet classes. The following topics are covered in this appendix:

- [“Business Components for Siebel Pharma Handheld” on page 291](#)
- [“Business Component Classes for Siebel Pharma Handheld” on page 297](#)
- [“Applet Classes for Siebel Pharma Handheld” on page 301](#)
- [“Siebel Pharma Handheld Objects that Must Not Be Modified” on page 308](#)

Business Components for Siebel Pharma Handheld

Table 36 lists the supported business components for Siebel Pharma Handheld applications.

Table 36. Business Components for Siebel Pharma Handheld

| Supported Business Components | Description |
|-------------------------------|--|
| Account | Holding all account related information, this business component supports Find enhanced functionality - which provides the ability to search for accounts based on entered search criteria. |
| Account Contact Affiliation | Holds contact affiliation information in relation to accounts. |
| Account Note | Holds note information in relation to an account. |
| Account Private Note | Holds private note information in relation to an account. |
| Action | Holding all activity related information, this business component supports the creation of new account calls from the daily calendar view, creation of new Med Ed records, and dynamic drilldown from the daily calendar view to Med Ed Event. |
| Action Contact | Holds action information in relation to contacts. |
| Action Employee | Holds action information in relation to employee. |
| Admin SEA Field | Holds administration SEA field information. |

Table 36. Business Components for Siebel Pharma Handheld

| Supported Business Components | Description |
|---------------------------------|--|
| CSLS Contact Best Times - CE | Holds information in relation to the best time to call a contact. |
| CUT Address - CE | The Pick Applet based on this business component is used in the HH Contact Find Form Applet - CE, Contact Find Popup Applet, HH Account Find Form Applet - CE, and Account Find Popup Applet. This business component supports multi select and displays address related fields, for example, Postal code and City. The user can either select one or more values. |
| CUT Address for Account/Contact | Holds address related information (such as, postal code and city) in relation to accounts/contacts. |
| Contact | Holds contact related information, such as, first name, last name, title, phone, and fax details. This business component does not push contacts to the device which are marked as <i>Personal</i> . NOTE: Contacts are individuals with whom a pharmaceutical company currently does business or with whom it expects to do business in the future. A typical contact is a physician, nurse, pharmacist, or any other medical professional. |
| Contact Account Affiliation | Holds information in relation to the best time to call a contact. |
| Contact (All) | This is similar to the Contact business component, except that it pushes contacts to the device which are marked as <i>Personal</i> and owned by the user. |
| Currency | Holds currency information. |
| Database Version | Holds database version information. |
| Decision Issue | Holds decision issue information. |
| Employee | Holds employee related information, such as, user ID, first name, last name, and position. NOTE: Employees are individual users in the Company who have access to the Company data. |
| Expense | Holds expense related information, such as, name, submitted to, pay to, and period for which the expense has been submitted. |
| Expense Item | Holds details about the individual expense items that are being reported in an expense report. The details include date on which the expense was incurred, amount, type of expense, and reimbursable flag. |

Table 36. Business Components for Siebel Pharma Handheld

| Supported Business Components | Description |
|---------------------------------|--|
| Expense Item Type | Holds information about types of expense items. |
| Internal Division | Holds information about internal divisions. |
| Internal Product | Holds details about a pharmaceutical company's products, such as, samples, promotional items, and product details. The details include product name, type, and inventory tracking flag. |
| Language Def | Holds information in relation to languages defined. |
| List of Values | Holds information in relation to list of values. |
| List of Values Child | Holds <i>child</i> information in relation to list of values. |
| List of Values Dynamic | Holds <i>dynamic</i> information in relation to list of values. |
| Opportunity | Holds opportunity related information, such as, accounts associated, sales team members, close date, and opportunity objectives. NOTE: Opportunity is any potential revenue generating event. |
| Period | Holds information in relation to inventory periods. |
| Pharma Account Call - CE | Holds information about account call activities, including information about the account targeted in the call, call date, and location. |
| Pharma Address | The Pick Applet based on this business component is used in the HH Contact Find Form Applet - CE and Contact Find Popup Applet. This business component supports multi select and displays all the accounts that are affiliated to a contact. The user can either select one or more values. |
| Pharma Address Inter Table - CE | Holds address information in relation to pharma contacts. |
| Pharma Area | Holds area information in relation to pharma contacts. |
| Pharma Call Decision Issue | Holds decision issue information in relation to pharma calls. |
| Pharma Call Decision Issue - CE | Holds information about any objections that a medical professional raises about a product in marketing calls. The information stored includes name and a description of the issue. |
| Pharma Call Products Detailed | Holds product details in relation to pharma calls. |

Table 36. Business Components for Siebel Pharma Handheld

| Supported Business Components | Description |
|--|--|
| Pharma Call Products Detailed - CE | Holds information about pharma call products of type <i>Detail</i> . These are the samples that are described to medical professionals in marketing calls. |
| Pharma Call Sample Dropped - CE | Holds information about pharma call products of type <i>Sample</i> . The sampling of these products is governed by regulatory acts. |
| Pharma Call Signature | Holds the Signature of the contact who signed for the samples dropped in a contact call or an account call along with Signature Header, Disclaimer Text, and necessary contact details. |
| Pharma GA Markets | Holds information in relation to pharma GA markets. |
| Pharma GA Rx Consumption | Holds information in relation to pharma GA Rx consumption. |
| Pharma Internal Product | Holds information about a pharmaceutical company's internal products. |
| Pharma Inventory Period | Holds information about a pharmaceutical company's inventory period. |
| Pharma ME Event | Holds details about the Medical Education events to which a contact is invited. This business component also supports Calendar enhanced functionality, which provides the ability to create an activity for every new Med Ed Event record created. The activity displays on the calendar view and enables the user to drill down from the Daily calendar view to Med Ed Event. |
| Pharma Personal Inventory - CE | Holds information about a pharmaceutical company's personal inventory. |
| Pharma Personal Products - CE | Holds the personal list data for an employee including all products of type <i>Detail</i> , which can be described in calls with medical professionals. |
| Pharma Personal Promotional Items - CE | Holds the personal list data for an employee including all products of type <i>Promotional Item</i> , which can be given to medical professionals during calls to promote the prescription of samples. |
| Pharma Personal Samples - CE | Holds the personal list data for an employee including all products of type <i>Sample</i> , which can be disbursed to medical professionals during calls. |
| Pharma Product Issue | Holds product issue information for a pharmaceutical company. |
| Pharma Professional Addresses - CE | Holds professional address information for a pharmaceutical company. |

Table 36. Business Components for Siebel Pharma Handheld

| Supported Business Components | Description |
|---|---|
| Pharma Professional Call - CE | Holds information about contact call and attendee call activities, including information about the contact targeted in the call, call date, and location. |
| Pharma Professional Note | Holds note information for a pharmaceutical company. |
| Pharma Professional Position | Holds position information for a pharmaceutical company. |
| Pharma Professional Primary Specialty | Holds primary specialty information for a pharmaceutical company. |
| Pharma Professional Private Note | Holds private note information for a pharmaceutical company. |
| Pharma Professional State Licenses | Holds state licenses information for a pharmaceutical company. |
| Pharma Professional Items Dropped - CE | Holds information about items dropped for a pharmaceutical company. |
| Pharma Sales Consumption | Holds information about sales consumption for a pharmaceutical company. |
| Pharma Sample And Promo Order | Holds information about sample orders and promotion orders for a pharmaceutical company. |
| Pharma Sample And Promo Order Item | Holds individual order item information about sample orders and promotion orders for a pharmaceutical company. |
| Pharma Sample Order And Adjustment | Holds sample order and adjustment information for a pharmaceutical company. |
| Pharma Sample Order And Adjustment Item | Holds individual item information about sample orders and adjustments for a pharmaceutical company. |
| Pharma Sample Stocked Lots | Holds information about sample stocked lots for a pharmaceutical company. |
| Pharma Sample Stocked Products | Holds information about sample stocked products for a pharmaceutical company. |
| Pharma Sample Transaction History Item - CE | Holds the transactional data related to the samples dropped in the contact or attendee call. Data includes sample name, date on which the transaction occurred, and status of the transaction (In Progress, Submitted, Adjusted). |
| Pharma Samples | Holds information about samples for a pharmaceutical company. |

Table 36. Business Components for Siebel Pharma Handheld

| Supported Business Components | Description |
|-------------------------------|---|
| Pharma Signature Disclaimer | Holds the signature disclaimer information for a pharmaceutical company. |
| Pharma Template Call | Holds template call information for a pharmaceutical company. |
| Picklist Generic | Holds information about generic picklists for a pharmaceutical company. |
| Picklist Hierarchical | Holds hierarchical details about picklists for a pharmaceutical company. |
| Position | Holds position information for a pharmaceutical company. |
| Price List | Holds price list information for a pharmaceutical company. |
| Query List | Holds query list information for a pharmaceutical company. |
| Retail Outlet Best Call Time | Holds information about the best time to call on retail outlets for a pharmaceutical company. |
| SIS HH Contact Activity | Holds activity information in relation to contacts for a pharmaceutical company. |
| System Preferences | Holds system preference information for a pharmaceutical company. |
| Time Expense Map | Holds information about time expense map for a pharmaceutical company. |
| Time Zone | Holds time zone information for a pharmaceutical company. |

Other business components supported by Siebel Pharma Handheld include the following:

- CUT Address Inter Table - CE
- Com Organization
- Employee (MM)
- LS Product Indiction
- Pharma Inventory Period - CE
- Pharma ME Event Professional Invitee Inter Table
- Shift
- Shift Exception

Business Component Classes for Siebel Pharma Handheld

This topic lists the supported business component classes for Siebel Pharma Handheld applications.

NOTE: The business component classes for Siebel Handheld applications are based on the classes for the Siebel Web Client. However, in most instances, because of the limitations of Siebel Handheld applications, only a subset of the class functionality that is required for the Siebel Handheld application is supported. All business components are shared by the Siebel Handheld application and the Siebel Server.

CSSBCAccountSIS

This class provides specialized functionality for managing a hierarchical account through its life cycle. This class is one of many in the hierarchy of classes that make up the Account Module in Siebel Industry Applications.

Usage Guidelines: Use this class in Pharma Handheld to create a new account, by copying from an existing account. The new account created inherits some attributes from the original account, for example, Price List, Tax List, and Product Distribution records. This allows a Rep to place orders for the new account straight away.

NOTE: For more information about the CSSBCAccountSIS class, see *Siebel Developer's Reference*.

Parent: CSSBCBase

Accessible Methods: Not Applicable

BC User Properties: Not Applicable

Field User Properties: Not Applicable

Dependencies & Limitations: This class can only be used for copying an account and assumes the existence of Tax List, Price List Fields, and Retail Audit Product BC.

CSSBCHHPharmaCall

This is a specialized business component class (and the main class) for call reporting.

Usage Guidelines: Use this class to perform validation and submission of calls, and to ensure that Submit is retriggered on the server during synchronization by logging sample transactions.

Parent: CSSBCPharmaCallComponent

| | |
|--|---|
| Accessible Methods: | The following methods are accessible from CSSBCPharmaCall: <ul style="list-style-type: none">■ ValidateOnSignBtnClick: Validates the sampling rules as defined in the business component's user properties. The rules validated include SubmitByPosition, Contact OK to Sample, Valid Contact Profile, and Sample Drops.■ BCUpdateInventoryonOK: Performs a fake inventory update on the Handheld application based on the products disbursed in the contact calls. This update does not create transaction history records; history records are created on the server during synchronization. |
| BC User Properties: | Not Applicable |
| Field User Properties: | Not Applicable |
| Dependencies & Limitations: | None |

CSSBCHHSampleItem

This is a specialized business component class for transactions.

| | |
|--|---|
| Usage Guidelines: | Use this class to implement the functionality necessary to adjust sample transactions or add new transactions to already submitted contacts or attendee calls. |
| Parent: | CSSBCSampleItem |
| Accessible Methods: | The following methods are accessible from CSSBCHHSampleItem: <ul style="list-style-type: none">■ Submit: Submits transaction items, marking the status as <i>Submitted</i>; the original transaction item is marked as <i>Adjusted</i>.■ Adjust: Copies the selected transaction item and makes the quantity editable in the copy.■ AddTxn: Adds a new transaction item to an existing transaction.■ IsItemSubmitted: Returns TRUE/FALSE depending on whether the transaction item is <i>Submitted</i> or not.■ IsTxnTypeDisbursement: Returns TRUE/FALSE depending on whether the transaction item type is <i>Disbursement</i> or not.■ IsSampleBeingAdjusted: Returns "1" or "2" depending on whether there is an adjustment item created for the selected transaction item. |
| BC User Properties: | Not Applicable |
| Field User Properties: | Not Applicable |
| Dependencies & Limitations: | None |

Other Supported Business Component Classes

Other supported business components classes for Siebel Pharma Handheld applications include the following:

- CSSBCBSampleLot
- CSSBCContactSIS
- CSSBCDivision
- CSSBCDocInvoice
- CSSBCExpitm
- CSSBCExprpt
- CSSBCFINOppty
- CSSBCFINSActivity
- CSSBCHHInventoryPeriod
- CSSBCHHSample
- CSSBCInventoryPeriod
- CSSBCPeriod
- CSSBCPharmaAddress
- CSSBCPharmaCall
- CSSBCPharmaCallComponent
- CSSBCPharmaDisclaimer
- CSSBCPharmaSign
- CSSBCPosition
- CSSBCPriceList
- CSSBCProdRev
- CSSBCProductDetailed
- CSSBCPromoItemsDropped
- CSSBCSample
- CSSBCSampleDropped
- CSSBCSampleInventory
- CSSBCStateModel
- CSSBCStateModelState
- CSSBCStateTransition
- CSSCommandShell
- CSSExpenseAutoGenService

- CSSOMAlarmMgr
- CSSPreferencesService
- CSSQueryExporter
- CSSSISOMPricerService
- CSSSMCacheService
- CSSTimeSheetAutoGenService

The following classes are supported by Siebel Pharma Handheld, but there is no specialized code in the Siebel Handheld application (specific to Siebel Pharma Handheld) for these classes.

- CSSBCAutoProduct

NOTE: It is required that you use the same class name for CSSBCAutoProduct on both the Siebel Handheld application and the Siebel Server. However, unlike the class on the Siebel Server, there is no functionality in the class on the handheld application.

- CSSBCBase

NOTE: CSSBCBase is a base class from which other business component classes are derived. This class provides functionality through business component user properties and invokes methods that are useful in many common situations. For more information about the CSSBCBase class, see *Siebel Developer's Reference*.

- CSSBCUser

NOTE: CSSBCUser is the base class for people-related business components, such as, User, Employee, Contact, and Personal Contact. This class provides functionality for user creation and management, external security adapter integration, Sync List for PIM devices, Personal Contact promotion, and other general contact and employee tasks. For more information about this class, see *Siebel Developer's Reference*.

- CSSBusComp

NOTE: CSSBusComp is a base class from which other business component classes are derived. It provides functionality through business component user properties and an object interface. For more information about this class, see *Siebel Developer's Reference*.

Applet Classes for Siebel Pharma Handheld

You can create additional screens and views for your Handheld application. Any new screens and views must be based on classes and business components that are supported for the Siebel Pharma Handheld application.

This topic lists the supported applet classes for Siebel Pharma Handheld applications.

CSSFrame

This is a non-specialized frame class that implements the base functionality required for applets. All applet classes derive from this class. This class handles the rendering of controls and processes all events raised at the applet level.

NOTE: For more information about the `CSSFrame` class, see *Siebel Developer's Reference*.

CSSFrameAcctCall

This is a specialized business component class for the account list applet. It supports new account call creation, automatic call creation, and automatic update of created calls.

Usage Guidelines: Use this class to invoke the `FindView` method which navigates to another view. The View name is configurable. The `GetViewBOBCName` method allows the user to define the view, `BusObj`, and `BusComp` name from the `auxconfig.txt` file. If no views are specified, it navigates to the default view, which is: Pharma Account Find View - CE, `BusObj`: Account, `Buscomp`: Account.

Parent: `CSSFrameList`

Accessible Methods: `AutoCall`, `AutoUpdate`, `NewCall`, `FindView`, `GetViewBOBCName`, `GetStrFromFile`, `GetNextLine`, `CreateProductsDetailed`, `UpdateProductsDetailed`, `UpdateAttendeeCall`, `UpdateAttendeeCallChildren`, `RemoveRecords`.

Applet User Properties: Not Applicable

Control User Properties: Not Applicable

Dependencies & Limitations: None

CSSFrameCEGridDay

This frame class is used to render the Calendar control for displaying appointments in a day. The day is divided into intervals (according to user preference) of 15, 30, 60, and 120 minutes.

CSSFrameCEGridMonth

This frame class renders the Calendar control for displaying appointments in a month. The user can click on any of the days in a month (any date) to navigate to that day's view for the selection.

CSSFrameCEGridWeek

This frame class renders the Calendar control for displaying appointments in a week. Seven intervals are displayed with each slot representing a day of the week. The user can click on any of the days in a week (any date) to navigate to that day's view for the selection.

CSSFrameCEHome

This is a specialized frame class used for the Home Page view of the Siebel Pharma Handheld application. This frame class renders buttons that are large in size - up to a maximum of 30 pixels (height) x 100 pixels (width). This class also suppresses the record count information in the track text.

CSSFrameFindContact

This is a specialized business component class supporting search functionality based on field values in the frame for *Contacts*. This class is derived from the class "[CSSFrame](#)" on page 301.

Usage Guidelines: Use this class to invoke search functionality on an applet. Upon clicking the Search button, it finds records based on the entered field values on a form. Upon clicking the All button, all contacts are displayed irrespective of the search criteria entered in the field values on a form.

Parent: CSSFrame

Accessible Methods: SetFrameActive, PickMultiple, SearchContacts, AllContacts, InitializeFrame, GetStrFromFile, GetNextLine, GetFldName, GetLinkName.

Applet User Properties: The following user properties are available:

- Name: PickApplet 1
Value: Account Find, Name
- Name: PickApplet 2
Value: Postal Code Find, Postal Code
- Name: SHCEHideToolBar
Value: TRUE
- Name: SHCEHideStatusBar
Value: TRUE

Control User Properties: Not Applicable

Dependencies & Limitations: Not Applicable

CSSFrameFindAccount

This is a specialized business component class supporting search functionality based on field values in the frame for *Accounts*. This class is derived from *CSSFrame*.

Usage Guidelines: Use this class to invoke search functionality on an applet. Upon clicking the Search button, it finds records based on the entered field values on a form. Upon clicking the All button, all accounts are displayed irrespective of the search criteria entered in the field values on a form.

Parent: *CSSFrame*

Accessible Methods: *SetFrameActive*, *PickMultiple*, *SearchAccounts*, *AllAccounts*, *InitializeFrame*, *GetStrFromFile*, *GetNextLine*, *GetFldName*, *GetLinkName*.

Applet User Properties: The following user properties are available:

- Name: *PickApplet 1*
Value: City Find, City
- Name: *PickApplet 2*
Value: Postal Code Find, Postal Code
- Name: *SHCEHideToolBar*
Value: TRUE
- Name: *SHCEHideStatusBar*
Value: TRUE

Control User Properties: Not Applicable

Dependencies & Limitations: Not Applicable

CSSFrameListSampleItem

This is a specialized business component class to create and adjust sample transactions for contact or attendee calls.

Usage Guidelines: Use this class to implement the functionality which enables users to add items to sample transactions, adjust the quantity of transaction items, and submit transaction items.

Parent: Not Applicable

| | |
|--|--|
| Accessible Methods: | The following methods are accessible from CSSFrameListSampleItem: <ul style="list-style-type: none">■ ItemSubmit: Submits a new transaction item or an adjusted transaction item.■ AdjustTxn: Makes a copy of the selected transaction and enables the user to modify the quantity in the copied record. Once the modified record is submitted, the original transaction is marked as <i>Adjusted</i> and the modified record status is marked as <i>Submitted</i>.■ AddTxn: Creates a new transaction item so that the user can add a different sample to an existing transaction. |
| Applet User Properties: | Not Applicable |
| Control User Properties: | Not Applicable |
| Dependencies & Limitations: | None |

CSSFrameLSGridDay

This is a specialized business component class for the daily calendar applet. It supports new contact call creation using the New Call button on the calendar.

| | |
|--|--|
| Usage Guidelines: | Use this class to invoke the New Account Call and the Med Ed Event methods. New Account Call creates a new account call and populates the call time with the timeslot selected in the calendar. Med Ed creates a new record for Med Ed Event and populates the Start time with the timeslot selected in the calendar. |
| Parent: | CSSFrameCEGridDay |
| Accessible Methods: | CalNewCall, CalNewAcctCall, CalMedEd. |
| Applet User Properties: | Not Applicable |
| Control User Properties: | The following control user properties are available: <ul style="list-style-type: none">■ Name: Account Call View Name Value: SIS HH Pharma Account Call Products Detailed View - CE■ Name: Account Call BO Name Value: Pharma Account Call - CE■ Name: Med Ed View Name Value: My Med Ed Events View - CE■ Name: Med Ed BO View Value: Pharma ME Event - CE |
| Dependencies & Limitations: | None |

CSSFrameMultiplePick

This is a specialized business component class supporting multiselect functionality in pick applets. This class is derived from CSSFrameListPick.

Usage Guidelines: Use this class to invoke multiple value selection on a Pick Applet, and to support sequential and random selection of records on a pick applet.

Parent: CSSFrameListPick

Accessible Methods: OnPickMultipleRecord, LoadNonSeq.

Applet User Properties:

Control User Properties: The following control user property is available:

- Name: AutoSelect

Value: TRUE

Dependencies & Limitations: Not Applicable

CSSFramePopupAccount

This is a specialized business component supporting search functionality based on field values in the frame for *Accounts*. This class is derived from CSSFramePopup.

Usage Guidelines: Use this class to invoke search functionality on an applet. Upon clicking the Search button, it find records based on the entered field values on a popup applet. Upon clicking the All button, all accounts are displayed irrespective of the entered search criteria in the field values on a popup applet.

Parent: CSSFramePopup

Accessible Methods: InitializeFrame, PickMultiple, CalculateAppletWidth, SearchAccounts, AllAccounts, NotifyNewQuerySpec, GetFldName, GetLinkName, GetStrFromFile, GetNextLine.

Applet User Properties: The following user properties are available:

- Name: PickApplet 1

Value: City Find, City

- Name: PickApplet 2

Value: Postal Code Find, Postal Code

Control User Properties: Not Applicable

Dependencies & Limitations: Not Applicable

CSSFramePopupContact

This is a specialized business component class supporting search functionality based on field values in the frame for *Contacts*. This class is derived from *CSSFramePopup*.

Usage Guidelines: Use this class to invoke search functionality on an applet. Upon clicking the Search button, it find records based on the entered field values on a popup applet. Upon clicking the All button, all accounts are displayed irrespective of the entered search criteria in the field values on a popup applet.

Parent: *CSSFramePopup*

Accessible Methods: *InitializeFrame*, *PickMultiple*, *CalculateAppletWidth*, *SearchContacts*, *AllContacts*, *NotifyNewQuerySpec*, *GetFldName*, *GetLinkName*, *GetStrFromFile*, *GetNextLine*.

Applet User Properties: The following user properties are available:

- Name: *PickApplet 1*
Value: City Find, City
- Name: *PickApplet 2*
Value: Postal Code Find, Postal Code

Control User Properties: Not Applicable

Dependencies & Limitations: Not Applicable

CSSFrameProfCall

This is a specialized business component class for the account list applet. It supports new contact call creation, automatic call creation, and automatic update of created calls.

Usage Guidelines: Use this class to invoke the *FindView* method which navigates to another view. The View name is configurable. The *GetViewBOBCName* method allows the user to define the view, *BusObj*, and *BusComp* name from the *auxconfig.txt* file. If no view is specified, it navigates to View: "Pharma Professional Find View - CE", *BusObj*: *Contact*, *BusComp*: *Contact*.

Parent: *CSSFrameList*

Accessible Methods: *AutoCall*, *AutoUpdate*, *NewCall*, *FindView*, *GetViewBOBCName*, *GetStrFromFile*, *GetNextLine*, *CreateProductsDetailed*, *UpdateProductsDetailed*, *UpdateAttendeeCall*, *UpdateAttendeeCallChildren*, *RemoveRecords*.

Applet User Properties: Not Applicable

Control User Properties: Not Applicable

Dependencies & Limitations: None

Other Supported Applet Classes

Other supported applet classes for Siebel Pharma Handheld applications include the following:

- CSSDurationFrameList
- CSSFrameCEMultiPart
- CSSFrameListExpitm
- CSSFrameListPharmaActivity
- CSSFrameListSample
- CSSFrameNewCall
- CSSFrameSISAction
- CSSFrameSample
- CSSSWEFrameListPick
- CSSSWEFrameListTargetAssoc
- CSSSWEFrameListVisibilityAssoc
- CSSSWEFrameListVisibilityPick
- CSSSubmitFrame

The following classes are supported by Siebel Pharma Handheld, but there is no specialized code in the Siebel Handheld application (specific to Siebel Pharma Handheld) for these classes:

- CSSFrameBase

NOTE: *CSSFrameBase* provides functionality through applet user properties and invokes methods that are useful in many common situations. The *FrameListBase* classes contain the same functionality and behaviors as their *FrameBase* counterparts. For more information about this class, see *Siebel Developer's Reference*.

- CSSFrameCECaIAddModify
- CSSFrameList

NOTE: *CSSFrameList* is a specialized applet class that represents a list applet object in the Siebel UI Frameworks. For more information about this class, see *Siebel Developer's Reference*.

- CSSFrameListBase

NOTE: *CSSFrameListBase* is a specialized applet class used for base frame functionality in list applets. For more information about this class, see *Siebel Developer's Reference*.

- CSSFramePopupCurrency

Siebel Pharma Handheld Objects that Must Not Be Modified

There are certain objects in the Siebel Pharma Handheld Client that must not be changed. [Table 37](#) lists the objects that must not be changed in Siebel Pharma Handheld applications.

CAUTION: If these objects are changed, the behavior of the Siebel Pharma Handheld application cannot be predicted.

NOTE: For guidelines on modifying existing objects, see *Configuring Siebel Business Applications*.

Table 37. Siebel Pharma Handheld Application Objects that Must Not Be Modified

| Type of Object | Objects | |
|-------------------------------|--|----------------------|
| Business Objects | Action - Account - CE | |
| | Action - Asset - CE | |
| | Action - Contact - CE | |
| | Action - Internal Product - CE | |
| | Action - Service Request - CE | |
| | Asset - Contact - CE | |
| | Asset - FS Inventory Location - CE | |
| | FS Activity Rec Parts & Tools - Order - CE | |
| | FS InvLoc Product - Internal Product - CE | |
| | FS Part Browser - FS Inventory Location - CE | |
| | Line Items - Action - CE | |
| | Line Items - Internal Product - CE | |
| | Order - Asset - CE | |
| | Order - Service Request - CE | |
| | Service Request - Account - CE | |
| | Service Request - Asset - CE | |
| | Service Request - Contact - CE | |
| | Service Request - Internal Product - CE | |
| | Business Components | Action - CE |
| | | Service Request - CE |
| Hidden Picklists BusComp - CE | | |

D

User Properties

This appendix documents the specific user properties for Siebel Pharma Handheld applications. The following topics are included in this appendix:

- ["User Properties for Siebel Pharma Handheld Applications" on page 309](#)

User Properties for Siebel Pharma Handheld Applications

This topic lists the user properties for Siebel Pharma Handheld applications.

AlphaTab Named Search User Property

Use this property to specify the field on which you want to implement the Alpha Tab Search feature.

Values: Fields available in your application.

Parent Object Type: List applet.

Usage: Select the field from the drop-down list.

For example, if you specify the Account Name field as the AlphaTab Names Search, the alpha search is applied to the Account Name field in the business component for the list applet, and the list applet displays an alpha tab search bar.

Note: You can configure as many AlphaTab Named Search properties as you require, but you can define only one for each list applet.

Attendee Call ReadOnly User Property

Use this property to specify the fields that are read-only when a call is an attendee call.

Values: Fields available in your application.

Default Value: ,Start Date,Start Time,Call Status,Sub Type,Address,Duration Minutes,Last Name, (default)

Parent Object Type: Pharma Professional Call—CE business component.

Usage: Use the same field names that are defined in Siebel Tools.

Note: A comma must always precede the first entry, and follow the last entry. Do not enter a space before or after the commas.

Example: ,Start Date,Start Time,Call Status,SubType,Address,Duration Minutes,

AutoPopupField User Property

Use this property to configure a view so that a pop-up list for a particular field automatically opens when the user opens the view.

Values: Fields available in your application.

Parent Object Type: View.

Usage: If the field is in a list applet, set the AutoPopupField property of the view to the Business Component field name.

If the field is in a form applet, set the AutoPopupField property of the view to the name of the control.

AutoSelect User Property

Use this property to enable users to select multiple items in a list control.

Values: TRUE or FALSE.

Parent Object Type: Control.

Usage: Set to TRUE to enable users to select multiple items.

BarcodeNewRecord User Property

Use this property to specify the behavior you require for new record creation when a scan is performed.

Values: Y or N.

Parent Object Type: Applet.

Usage: If you specify Y, when the user performs a parseable scan, a new record is created for the scan, and that record is populated with the values parsed from the barcode.

If you specify N, when the user performs a nonparseable scan, a new record is not created for the scan, but the active field is populated with the barcode number.

BC Read Only Field User Property

For information about this user property, see *Siebel Developer's Reference* and *Configuring Siebel Business Applications*.

BorderStyle User Property

Use this property to specify the border style of a signature control.

Values: Double or Single.

Parent Object Type: Signature control applet.

Usage: Specify the style you require for the control.

BrowserPath User Property

Use this property to specify the application to start when the user taps a URL or attachment in the application. By default, when users tap a URL or attachment, the item is displayed in Pocket Internet Explorer.

Values: String.

Parent Object Type: Application.

Usage: The default value for this property makes attachments and URLs open in Pocket Internet Explorer.

Cascading Fields User Property

Use this property to specify the fields to copy from the parent account call to the attendee call. These fields must be forced to be active in the child business component.

Values: Fields available in your application.

Default Value: ,Start Date,Start Time,Account Id,Address,Account Address Id,Account State,Duration Minutes,Brick Id,

Parent Object Type: Pharma Account Call—CE business component.

Usage: Use the same field names that are defined in Siebel Tools. A comma must always precede the first entry, and follow the last entry. Do not enter a space before or after the commas.

ConfirmAccept User Property

Use this property to specify whether to display a confirmation message after the user taps the Accept button on the signature control.

Values: Y or N.

Parent Object Type: Signature control applet.

Usage: Set this property to Y to display a confirmation message after the user taps the Accept button.

Set this property to N to display no confirmation message.

DefaultViewType User Property

Use this property to configure how applets are displayed in an application. For more information about applet display modes, see [Designing Screens and Views for Siebel Handheld Applications on page 32](#).

Values: Tabbed or Traditional.

Default Value: Traditional

Parent Object Type: Application.

Usage: If set to Tabbed, the first applet in the frame array of a view definition is treated as the parent applet. The parent applet is displayed when users navigate to this view. All other applets in the view are treated as child applets and appear on the tab bar of the parent applet. A parent context bar also appears at the top of the view.

If set to Traditional, the application displays the applets in the traditional two-applet view. To switch between the two applets, choose View, then Toggle.

In portrait orientation, the default tabbed applet behavior for the application, when not explicitly specified, is Traditional. You can change the default tabbed applet behavior to Tabbed, if required.

In landscape orientation and square screens, the default tabbed applet behavior for the application is Tabbed, and you cannot change this value to Traditional.

Note: This user property can be overridden by the view-level user property ViewType. For more information about the ViewType user property, see [ViewType User Property on page 332](#).

Disable Second Applet User Property

Use this property when you configure a tree applet to suppress the dynamically created second applet so that only a single applet is displayed.

Values: TRUE or FALSE.

Default Value: FALSE

Parent Object Type: View.

Usage: If set to TRUE, the second dynamic applet is not displayed.

If set to FALSE, only a single applet is displayed.

DisableAlphaTabGroups User Property

Use this property to specify how to display alpha tabs in ungrouped format.

Values: TRUE or FALSE.

Parent Object Type: Applet.

Usage: If set to TRUE, the alpha tabs are displayed in ungrouped format.

If set to FALSE, the alpha tabs are displayed in grouped format.

DisableParentBar User Property

Use this property to specify not to display the parent context bar in a view that displays multiple applets in tabbed applet mode. You might use this property if a view contains applets that are not linked in a parent-child relationship.

Values: TRUE or FALSE.

Parent Object Type: View.

Usage: If set to TRUE, the parent context bar is not displayed. When a user selects an applet, the tab of the applet is highlighted, but no parent context bar is displayed.

If set to FALSE, the parent context bar is displayed.

Electronic Signature Read Only User Property

Use this property to specify the fields that are read-only after the user captures a signature.

Values: Fields available in your application.

Default Value: ,Last Name,Address,Chosen Template,Start Date,CallStatus,Paper Signature,

Parent Object Type: Pharma Professional Call—CE business component.

Usage: Use the same field names that are defined in Siebel Tools.

Note: A comma must always precede the first entry, and follow the last entry. Do not enter a space before or after the commas.

EncryptKey User Property

Use this property to specify the default key to use when encrypting signatures. If you do not specify a value for this property, an internal mechanism is used to generate the encryption key.

Values: Integer, between 1 and 99999999.

Parent Object Type: Signature control applet.

Usage: Enter the encryption key you require as the value for this property.

Explorer Applet Size User Property

Use this property when you configure a tree applet to specify how much space you want the tree applet to occupy.

Values: Integer, between 1 and 100.

Default Value: 50

Parent Object Type: View.

Usage: Enter the percentage of space that you want the tree applet to occupy.

Field Read Only Field *fieldname* User Property

For information about this user property, see *Siebel Developer's Reference* and *Configuring Siebel Business Applications*.

FormUI User Property

Form UI is a technique of applet design that enables Siebel Handheld application developers to create applets that are larger than normal form applets. For more information about Form UI applets, see [Configuring Form UI Applets for the Siebel Handheld Client on page 40](#).

Use this property to specify whether an applet is a Form UI applet or not.

Values: TRUE or FALSE.

Parent Object Type: Applet.

Usage: If set to TRUE, makes the applet a form UI applet.

If set to FALSE, makes the applet a normal applet.

HandheldInsertFailMode User Property

Use this property to specify how to handle record insert failures for the business component. For more information about how insert failures are handled, see ["Insert Failure Processing in Siebel Handheld Applications" on page 198](#).

Values: NoFail or FailOnError.

Default Value: NoFail

Parent Object Type: Business component.

Usage: If you specify NoFail, insert failures are logged and synchronization continues.

If you specify FailOnError, insert failures are logged and synchronization terminates.

HandheldSyncPickMode User Property

Use this property to enable Extended Pick processing for specified fields in a business component. For more information about Extended Pick processing, see ["Extended Pick Processing in Siebel Handheld Applications" on page 195](#).

Values: *Fieldname_1|Mode_1, Fieldname_2|Mode_2 ... Fieldname_n|Mode_n*

Parent Object Type: Business component.

- Usage:** Specify the field name and the mode for each field for which you want to specify this property. Specify one of the following modes:
- NoFail
 - FailOnError
- If set to NoFail, enables Extended Pick processing for the specified field.
- If set to FailOnError, disables Extended Pick processing for the specified field.
- Note:** The setting on the business component takes precedence over the setting defined in the Settings view tab on the Administration - Mobile screen, then the Application Administration view.
- You can use the business component setting to nullify the global application of Extended Pick. Conversely, you can enable Extended Pick processing on specific business components if the setting in the Settings view tab has disabled the functionality.

InkColor User Property

Use this property to specify the color to use on the signature control applet.

- Values:** Hexadecimal value from 0x000000 to 0xFFFFFFFF.
- Parent Object Type:** Signature control applet.
- Usage:** The six hexadecimal digits in the value that you specify represent the color to use. The six digits comprise three groups of two digits. Each group of two digits is the value for one of the colors: blue, green, and red, in that order. For example, the value 0xFF0000 represents the color blue.

InkWidth User Property

Use this property to specify the width of the line that is created in the signature control applet.

- Values:** 1-5.
- Parent Object Type:** Signature control applet.
- Usage:** A typical line width is 2.

Max Auto Generated Priority User Property

Use this property to specify the maximum priority that is generated automatically for products detailed on a contact or account call.

- Values:** Integer.
- Default Value:** 3
- Parent Object Type:** Pharma Call Products Detailed—CE business component.
- Usage:** No additional information.

Maximum Periods User Property

Use this property to specify the number of unreconciled periods, including the active period.

- Values:** Integer.
- Default Value:** 2
- Parent Object Type:** Pharma Inventory Period—CE business component.
- Usage:** No additional information.

MaximumFooterLines User Property

Use this property to specify the maximum height of the footer display control.

- Values:** Integer.
- Parent Object Type:** Signature control applet.
- Usage:** Specify the height as the number of text lines. If the footer text exceeds this number, the user can use the pen to scroll vertically. The footer display control does not display a scroll bar.

MaximumHeaderLines User Property

Use this property to specify the maximum height of the header display control.

- Values:** Integer.
- Parent Object Type:** Signature control applet.
- Usage:** Specify the height as the number of text lines. If the header text exceeds this number, the user can use the pen to scroll vertically. The header display control does not display a scroll bar.

Multicurrency User Property

Use this property to enable multicurrency support for a frame. This property overrides the multicurrency behavior of the application.

- Values:** Y or N.
- Parent Object Type:** Frame.
- Usage:** Set to Y to enable multicurrency support.

Must Detail Products User Property

Use this property to specify whether a user must enter detailed information for a product before a call can be submitted.

- Values:** Y or N.
- Default Value:** Y

Parent Object Type: The following business components:

- Pharma Account Call—CE
- Pharma Professional Call—CE

Usage: Set to Y to specify that the user must enter detailed information.

On Field Update Set *n* User Property

For information about this property, see *Siebel Developer's Reference*.

OnlySubmitByPosition User Property

Use this property to specify whether the only position that can submit a call is the position that created the record.

Values: Y or N.

Default Value: Y

Parent Object Type: The following business components:

- Pharma Account Call—CE
- Pharma Professional Call—CE

Usage: Set to Y to prevent submission of calls by a position different to the one that created the record.

OverrideDefaultApplet User Property

Use this property to override the default behavior to give focus to an applet. This user property is typically used with the `GotoViewNewRecord` method.

Values: 0 or 1.

Parent Object Type: Control.

Usage: If set to 0, gives the focus to the parent applet.

If set to 1, gives the focus to the child applet.

OverridePopupVisibility*n* User Property

Use this property to specify fields that override the business component Popup Visibility Type setting in an applet. Use this property when the visibility you require for a field differs from the Popup Visibility Type of the business component, and you do not want to change the Popup Visibility Type of the business component. For more information about visibility types, see [Designating Visibility on page 164](#).

Values: *Business_Component_Field*, *Visibility_Type*

Parent Object Type: Applet control or column.

Usage: The name of the OverridePopupVisibility parameter must be unique, so you must append a unique number to the name of the parameter for any other items, for example, OverridePopupVisibility1, OverridePopupVisibility2, and so on.

Assign a value to this property that contains the name of the business component field, followed by a comma, and then the visibility type you require for the field. [Table 23](#) summarizes popup visibility types and the corresponding visibility codes.

Paper Signature Read Only User Property

Use this property to specify fields that are read-only when the Paper Signature field for the call is set to TRUE.

Values: Fields available in your application.

Default Value: ,Call Status,Signature Captured,

Parent Object Type: Pharma Professional Call—CE business component.

Usage: Use the same field names that are defined in Siebel Tools.

Note: A comma must always precede the first entry, and follow the last entry. Do not enter a space before or after the commas.

Paper Reference Number Required User Property

Use this property to specify whether a reference number is required to submit a call with a paper signature.

Values: Y or N.

Default Value: Y

Parent Object Type: Pharma Professional Call—CE business component.

Usage: Set to Y to specify that a reference number is required.

PrintForms User Property

You can implement multiple printing of forms in Siebel Handheld applications. To do this, you must add a control that invokes the PrintBatch method.

Use the PrintForms user property to specify the forms to print, the print templates, and a default number of copies of each form to print. You can specify up to 10 sets of forms, print templates, and default numbers.

For more information about how to configure multiple-form printing and multiple-quantity printing, see [Configuring Multiple Form and Multiple Quantity Printing for the Siebel Handheld Application on page 87](#).

Values: *form_name_1, template_filename_1, default_number_of_copies_1, form_name_2, template_filename_2, default_number_of_copies_2, ... form_name_10, template_filename_10, default_number_of_copies_10,*

Parent Object Type: Control.

Usage: Enter the sets of forms, print templates, and default numbers that you require.

Note 1: Do not enter a space before or after the commas. A comma must always follow the last entry.

Note 2: All of the applets that are specified in the following print tags must be configured in the view from which the PrintBatch method is invoked:

- Applet
- List
- GetTotal
- GetCount

In other words, all of the applets that are specified in these print tags, in all of the print templates that are used in the PrintBatch method, must be configured in the view.

Use the GetDirect print tag to get the value of a field from a business component that is not configured in the view.

Example: You might have forms with templates as follows:

- Order with Price, OrderwithPrice.txt
- Order without Price, OrderwithoutPrice.txt
- Temporary ID Card, TemporaryIDCard.txt

To print one copy of the first form only, specify the following value for this property:

```
Order wi th Pri ce, Orderwi thPri ce. txt, 1, Order wi thout
Pri ce, Orderwi thoutPri ce. txt, 0, Temporary ID
Card, TemporaryI DCard. txt, 0,
```

Propagate Address BC Fields User Property

Use this property to specify the fields to update in the S_ADDR_PER table that are propagated to the intersection table.

Values: Fields available in your application.

Parent Object Type: The following business components:

- CUT Address—CE
- Pharma Address

Usage: Enter the list of fields, separated by commas.

Note: Do not enter a space before or after the commas.

Example: City,Postal Code,State,Street,Address,Country,Phone#

Propagate Address Fields To Parent BC User Property

Use this property to specify the source and destination fields that are propagated to the parent business component.

Values: Fields available in your application.

Parent Object Type: The following business components:

- CUT Address—CE
- Pharma Address

Usage: Separate the source and destination fields with a vertical bar (|). Separate the pairs of fields with commas.

Examples: **CUT Address—CE**

City|City CE, Postal Code|Postal Code CE, State|State CE,Street Address|Street Address CE,Country|Country CE

In the example City|City CE, *City* is the source field, and *City CE* is the destination field.

Pharma Address

City|City, Postal Code|Postal Code, State|State,Street Address|Street Address, Country|Country

Read Only If Status Synchronized User Property

Use this property to specify whether to set records to read-only status if the call status is Synchronized.

Values: Y or N.

Default Value: N

Parent Object Type: The following business components:

- Pharma Account Call—CE
- Pharma Professional Call—CE

Usage: Set to Y to set all records in the business component to read-only when the call status is Synchronized.

Sample Disbursed Required User Property

Use this property to specify whether the user must have dropped samples before a call is submitted.

Values: Y or N.

Default Value: N

Parent Object Type: Pharma Professional Call—CE business component.

Usage: Set to Y if the user must have dropped samples before a call is submitted.

SHCE COM Disable User Property

Use this property to disable support for the COM feature for a Siebel Handheld application.

Values: Y or N.

Parent Object Type: Application.

Usage: Set to Y to disable the COM feature.

SHCEDailySlotHeight User Property

Use this property to change the time slot spacing in the Daily Calendar Applet.

Values: Integer, between 14 and 50.

Default Value: 15

Parent Object Type: Daily calendar applet.

Usage: No additional information.

SHCEHideNavBar User Property

Use this property to hide the navigation bar in a calendar applet or find applet.

Values: TRUE or FALSE.

Parent Object Type: Applet.

Usage: If set to TRUE, hides the navigation bar.

If set to FALSE, does not hide the navigation bar.

SHCEHideStatusBar User Property

Use this property to hide the status bar in a calendar applet or find applet.

Values: TRUE or FALSE.

Parent Object Type: Applet.

Usage: If set to TRUE, hides the status bar.

If set to FALSE, does not hide the status bar.

SHCEHideToolBar User Property

Use this property to hide the toolbar in a calendar applet or find applet.

Values: TRUE or FALSE.

Parent Object Type: Applet.

Usage: If set to TRUE, hides the toolbar.

If set to FALSE, does not hide the toolbar.

SHCEMenuBackButton User Property

Use this property to add the Go Back button to the menu bar in a calendar applet or find applet. This property enables you to have no toolbar on the applet to save screen space, and to still have a Go Back button in the applet.

Values: TRUE or FALSE.

Parent Object Type: Applet.

Usage: If set to TRUE, adds the Go Back button to the menu bar.

If set to FALSE, does not add the Go Back button to the menu bar.

SHCEMenuForwardButton User Property

Use this property to add the Go Forward button to the menu bar in an applet. This property enables you to have no toolbar on the applet to save screen space, and to still have a Go Forward button in the applet.

Values: TRUE or FALSE.

Parent Object Type: Applet.

Usage: If set to TRUE, adds the Go Forward button to the menu bar.

If set to FALSE, does not add the Go Forward button to the menu bar.

SHCEMenuNewButton User Property

Use this property to add the New Record button to the *menu bar* in an applet. This property enables you to have no toolbar on the applet to save screen space, and to still have a New Record button in the applet.

Values: TRUE or FALSE.

Parent Object Type: Applet.

Usage: If set to TRUE, adds the New Record button to the menu bar.

If set to FALSE, does not add the New Record button to the menu bar.

ShowData User Property

Use this property to specify if the content of the signature field can be displayed. If you specify not to display signature data, the ellipses in the signature applet control are disabled.

Values: Y or N.

Parent Object Type: Signature field in a signature capture applet.

Usage: Set this property to Y to allow the content of the signature field to be displayed.

Set this property to N to prevent the display of signatures.

Signature Body Text Field: *n* User Property

Use this property to specify the fields to appear in the signature body text.

Values: Fields available in your application.

Default Value: **Signature Body Text Field: 1**

Quantity

Signature Body Text Field: 2

Name

Signature Body Text Field: 3

Vendor

Parent Object Type: Pharma Call Sample Dropped—CE business component.

Usage: Field name that must appear in the signature body text.

Signature Date User Property

Use this property to specify whether to add the current date to the disclaimer.

Values: Y or N.

Parent Object Type: Pharma signature capture applet.

Usage: No additional information.

Signature Time User Property

Use this property to specify whether to add the current time to the disclaimer.

Values: Y or N.

Parent Object Type: Pharma signature capture applet.

Usage: No additional information.

Signature User Property

Use this property to specify if a field can store a signature.

Values: Y or N.

Parent Object Type: Signature capture applet.

Usage: Set this property to Y to make a field a signature field.
Set this property to N if you do not want a field to store signatures.

SignatureDataField User Property

Use this property to specify a destination field for the captured signature if you do not want to save the signature in the user interface field.

Values: A field name in the current business component.

Parent Object Type: Signature capture applet.

Usage: Enter the name of the business component field where you want to save the signature data.

SignatureField User Property

Use this property to specify the name of the user interface field where the signature data is stored.

Values: UI field name.

Parent Object Type: Signature button control.

Usage: Enter the name of the field where you want to store the signature data.

SignatureFooterField User Property

Use this property to specify the field from which to copy the content for the signature footer.

Values: A field name in the current business component.

Parent Object Type: Signature capture applet.

Usage: No additional information.

SignatureFooterSaveField User Property

Use this property to specify the field to which the content for the signature footer is saved.

Values: A field name in the current business component.

Parent Object Type: Signature capture applet.

Usage: If the SignatureDate user property is set to Y, the date is included in the data that is saved to this field.

If the SignatureTime user property is set to Y, the time is included in the data that is saved to this field.

SignatureHeaderField User Property

Use this property to specify the field from which to copy the content for the signature header.

Values: A field name in the current business component.

Parent Object Type: Signature capture applet.

Usage: No additional information.

SignatureHeaderSaveField User Property

Use this property to specify the field to which the content for the signature header is saved.

Values: A field name in the current business component.

Parent Object Type: Signature capture applet.

Usage: No additional information.

SignatureHeight User Property

Use this property to specify the height of the signature capture control.

Values: Integer, between 1 and 200.

Parent Object Type: Control on signature capture applet.

Usage: Specify the height in pixels.

SignatureWidth User Property

By default, signature controls occupy the full width of the screen on the handheld device. Use the SignatureWidth user property to set the width of a signature control.

Values: Integer, between 1 and 240.

Parent Object Type: UI field.

Usage: Enter the width of the screen.

SingleSelect User Property

Use this property to restrict to one the number of records that the user can select from a control.

Values: TRUE or FALSE.

Parent Object Type: Control.

Usage: If set to TRUE, disables multiselect mode and autoselect mode, and enables the user to select one record only.

If set to FALSE, enables multiselect mode and autoselect mode.

Skip Conflict Resolution for Duplicate Records User Property

Use this property to specify the behavior of the object manager when it encounters a duplicate record during synchronization. When a record is inserted which is considered a duplicate due to key constraints, the object manager can ignore the record from the Siebel Handheld application. Conflict resolution is not performed on duplicate records.

Values: Y or N.

Default Value: N

Parent Object Type: The following business components:

- Pharma Call Products Detailed—CE
- Pharma Call Sample Dropped—CE

Usage: Set to Y to make the object manager ignore duplicate records from the Siebel Handheld application.

SRE Professional Types User Property

Use this property to create a list of types of profession for the application. You can use this field to make sure that only a person from the appropriate profession can make a Signature Recorded Electronically (SRE) in the Siebel Handheld application.

Values: *,Profession_1, Profession_2, ... Profession_n,*

Parent Object Type: Pharma Professional Call—CE business component.

Usage: Specify a list of professions.

Note: A comma must always precede the first entry, and follow the last entry. Do not enter a space before or after the commas.

Example: *,Physician,Central Lab Technician,*

SRE Reference Number Required User Property

Use this property to specify whether an applet must contain a reference number before the user can submit the data in the applet.

Values: Y or N.

Default Value: N

Parent Object Type: Pharma Professional Call—CE business component.

Usage: Set to Y if the applet must contain a reference number.

SubCall Component CE User Property

Use this property to specify the child business components to which the Siebel Handheld application database changes are copied.

Values: Business components available in your application.

Default Value: Pharma Professional Call - CE

Parent Object Type: Pharma Account Call—CE business component.

Usage: Use the same business component names that are defined in Siebel Tools.

Submit At Handheld Sync User Property

Use this property to specify that contact calls are automatically submitted when a user synchronizes the Siebel Pharma Handheld application.

Values: Y or N.

Default Value: Y

Parent Object Type: Pharma Account Call—CE business component.

Usage: Set to Y to specify that contact calls are automatically submitted.
Set to N to specify that contact calls are not automatically submitted.

TaskUI User Property

Use this property to specify that a view contains a Siebel Task. For more information about Siebel Tasks, see [Developing Tasks for Task UI in Siebel Handheld Applications on page 77](#).

Values: TRUE or FALSE.

Parent Object Type: View.

Usage: If set to TRUE, specifies that the view contains a Siebel Task.
If set to FALSE, specifies that the view is not a Siebel Task.

Template Components User Property

Use this property to specify the business components to copy from the call template when a Smart Call is associated with a call.

Values: Business components available in your application.

Default Value: **Pharma Account Call—CE**

Pharma Call Products Detailed - CE

Pharma Professional Call—CE

Pharma Call Products Detailed - CE, Pharma Call Sample Dropped - CE

Parent Object Type: The following business components:

- Pharma Account Call—CE
- Pharma Professional Call—CE

Usage: Use the same business component names that are defined in Siebel Tools.

Template Fields User Property

Use this property to specify the fields to copy from the call template when a Smart Call is associated with a call.

Values: Fields available in your application.

Default Value: **Pharma Call Decision Issue—CE**

,Name,

Pharma Call Products Detailed—CE

,Name,Priority,Indication

Pharma Call Sample Dropped—CE

,Name,Lot Name,Quantity,

Pharma Promotional Items Dropped—CE

,Name,Quantity,

Parent Object Type: The following business components:

- Pharma Call Decision Issue—CE
- Pharma Call Products Detailed—CE
- Pharma Call Sample Dropped—CE
- Pharma Promotional Items Dropped—CE

Usage: Use the same field names that are defined in Siebel Tools.

Note: Do not enter a space before or after the commas. A comma must always precede the first entry, and follow the last entry.

TriggerFindRecord User Property

For information about how to use this property, see [Enhancing the Behavior of the Barcode Scan on page 119](#).

Update After Recreate Receipt User Property

Use this property to specify the fields that users can edit when the Recreate Receipt field is selected.

Values: Fields available in your application.

Default Value: ,LastName,Address,Start Date,Comment,Signature Captured,

Parent Object Type: Pharma Professional Call—CE business component.

Usage: Use the same field names that are defined in Siebel Tools.

Note: Do not enter a space before or after the commas. A comma must always precede the first entry, and follow the last entry.

Update After Submit User Property

Use this property to specify the fields that users can edit after a call is submitted.

Values: Fields available in your application.

Default Value: ,Comment,

Parent Object Type: The following business components:

- Pharma Account Call—CE
- Pharma Professional Call—CE

Usage: Use the same field names that are defined in Siebel Tools.

Note: Do not enter a space before or after the commas. A comma must always precede the first entry, and follow the last entry.

Update Inter Table BC Fields User Property

Use this property to specify a list of fields that must be updated in the inter table, that is, in S_CON_ADDR.

Values: Fields available in your application.

Parent Object Type: Pharma Address business component.

Usage: Do not enter a space before or after the commas. A comma must always precede the first entry.

Example: DEA#,DEA Expr

Update Status To Synchronized User Property

Use this property to specify whether to update the status, on the Siebel Server, of the call records that users download to the Siebel Pharma Handheld application.

Values: Y or N.

Default Value: Y

Parent Object Type: The following business components:

- Pharma Account Call—CE
- Pharma Professional Call—CE

Usage: Set to Y to specify to update the status of the call records on the Siebel Server.

Note: The status of records on the Siebel Server is updated as follows:

| Current Status | Status Changes To |
|-----------------------|--------------------------|
| Submitted | Does not change |
| Signed | Signed or Synchronized |
| Any other status | Synchronized |

The calls where the status is Synchronized are read-only on the Siebel Server, if the user property, Read Only If Status Synchronized, is set to Y. For more information about the Read Only If Status Synchronized property, see [Read Only If Status Synchronized User Property on page 320](#).

UseSystemLocale User Property

Use this property if you want a Calendar applet to display the dates according to the regional date settings on the handheld device where the application is installed.

Values: TRUE or FALSE.

Parent Object Type: Applet.

Usage: If set to TRUE, the applet uses the regional date settings on the handheld device to display dates on the calendar.

If set to FALSE, the applet does not use the regional date settings.

Validate DEA Number User Property

Use this property to specify that the DEA Number is validated in the following events:

- Sign button is checked.
- User updates a DEA number.
- Any other action that causes the field to be saved.

An example of a valid DEA number is: AA1234563.

Values: Y or N.

Default Value: Y

Parent Object Type: The following business components:

- Pharma Address
- Pharma Address Inter Table—CE
- Pharma Professional Call—CE

Usage: Set to Y to specify that the DEA Number must be validated.

Set to Y to validate DEA numbers.

Validate Product Priority User Property

Use this property to display an error message during submission if the product detail priority is not in a sequential order that begins with 1.

Values: Y or N.

Default Value: N

Parent Object Type: The following business components:

- Pharma Account Call—CE
- Pharma Professional Call—CE

Usage: Set to Y to display an error message for products with invalid priority information.

Validation User Properties for Pharma Professional Call—CE Business Component

This section describes several validation-related user properties for the Pharma Professional Call—CE business component.

Values: Y or N.

Default Value: Y

Parent Object Type: Pharma Professional Call—CE business component.

User Property:

| | |
|------------------------------------|---|
| Validate DEA Number Expiration | Set to Y to specify that the application must validate the DEA Expiration Date for a call before the user can sign and submit the call. |
| Validate License Number | Set to Y to specify that the application must validate the License Number for a call before the user can sign and submit the call. |
| Validate License Number Expiration | Set to Y to specify that the application must validate the License Number Expiration Date for a call before the user can sign and submit the call. |
| Validate License Number Status | Set to Y to specify that the application must validate the License Number Status for a call before the user can sign and submit the call. |
| Validate OK to Sample | Set to Y to specify that the user can drop a sample to a contact in the call. |
| Validate Professional Profile | Set to Y to specify to check that the Last Name, First Name, Title, and Address fields are populated for a professional before a user can sign the handheld device. |
| Validate Sign Lot Number | Set to Y to specify to check that the Lot Number field is populated before a user can sign the handheld device. |

View User Property

Use this property to specify the name of the view to display when a user selects the control.

Values: Views available in your application.

Parent Object Type: Control.

Usage: No additional information.

ViewType User Property

Use this property to configure how applets are displayed in a view. For more information about applet display modes, see [Designing Screens and Views for Siebel Handheld Applications on page 32](#).

Values: Tabbed or Traditional.

Parent Object Type: View.

Usage: For information about the usage of this property, see [DefaultViewType User Property on page 311](#).

Note: ViewType overrides the DefaultViewType property in portrait orientation. In landscape orientation and on square screens, the default applet display mode is Tabbed.

WritingGuide User Property

Use this property to display information in the applet. This information provides instructions on how to create a signature on a handheld device.

Values: Y or N.

Parent Object Type: Signature control applet.

Usage: Set this property to Y to display the writing guide in the applet.

Set this property to N if you do not want to display the writing guide in the applet.

E

Print Tagging Language

This appendix describes the print tagging language that is used to create print templates for your handheld device. The audience for this appendix is Siebel Handheld application developers. The following topics are covered in this appendix:

- ["Overview of Print Tagging Language for Siebel Handheld Applications" on page 333](#)
- ["Print Mode Tags" on page 334](#)
- ["Data Output Tags" on page 335](#)
- ["Print Formatting Tags" on page 341](#)
- ["Get and Set Tags" on page 354](#)
- ["Control Positioning Tags" on page 361](#)
- ["Data Manipulation and Logic Tags" on page 365](#)
- ["Using Variables in Print Templates" on page 369](#)

Overview of Print Tagging Language for Siebel Handheld Applications

You can create the print template file in any text editor:

- Tags are enclosed in angle brackets and identified with the keyword TAG, followed by a colon. For example, <TAG: . . . >.
- Parameters are specified with a keyword, followed by an equals sign (=). Arguments are in uppercase, surrounded by quotation marks. For example, Font = "FONT FACE".
- Values are in uppercase and lowercase, surrounded by quotation marks. For example, Font = "Hel veti ca".
- Tags can appear in any order in the file. The only required tag is the Title tag.
- Variables within a print template must be uniquely named. Variable names must be one-word strings and spaces are not allowed.

Print Mode Tags

The following two tags determine the mode of printing. The print mode determines how printing is going to be handled by the Siebel Handheld application.

- [“Absolute Tag” on page 334](#)
- [“Relative Tag” on page 334](#)

Absolute Tag

The Absolute tag sets the print mode to absolute. Print mode determines how printing is going to be handled by your Handheld application. Absolute mode-only tags and parameters are enabled and processed accordingly while in absolute mode.

Usage

The Absolute tag can appear anywhere in the template for any number of times to switch between print modes. Print mode defaults to Relative if none is specified.

Syntax

To use this tag, use the following syntax:

```
<TAG:Absolute>
```

| Parameter | Description |
|--------------|------------------------------------|
| TAG:Absolute | (Required) Absolute tag indicator. |

Relative Tag

The Relative tag sets the print mode to relative, cancelling the absolute mode. Absolute mode-only tags and parameters are disabled and ignored while in Relative mode.

Usage

The Relative tag can appear anywhere in the template for any number of times to switch between print modes. Print mode defaults to Relative if none is specified.

Syntax

To use this tag, use the following syntax:

```
<TAG:Relative>
```

| Parameter | Description |
|--------------|------------------------------------|
| TAG:Relative | (Required) Relative tag indicator. |

Data Output Tags

The following tags are used to control data output for printing:

- [“Applet Tag” on page 335](#)—Specifies which applets are used in the document.
- [“Cell Tag” on page 337](#)—Creates cells of data in a row in a table.
- [“Picture Tag” on page 339](#)—Adds graphic images to the document.
- [“Static Text Tag” on page 341](#)—Adds text that does not change to the document.

Applet Tag

The Applet tag specifies the applet to be printed.

Usage

Any data that appears in your document must be generated from the applets in the view. Therefore, you must create applets that produce the data required for your document.

Use the Applet tag to specify the number of list or form applets within the template and to locate the applet in your document. In the form view, a maximum of two fields for each row are supported.

Syntax

To use this tag, use the following syntax:

```
<TAG:Applet name="APPLET NAME" query="QUERY" caption="CAPTION"
headerdivider="HEADERDIVIDER" recorddivider="RECORDDIVIDER" wrap="WRAP" grid="GRID"
linewidth="THICKNESS" offset="OFFSET" width="WIDTH" sql="SQL_Statement"
maxlines="MAXLINES" startat="Number">
```

| Parameter | Description |
|------------|---|
| TAG:Applet | (Required) Applet tag indicator. |
| name | (Required) Applet name as defined in Siebel Tools. |
| query | (Optional) Specifies the query string directly set to the business component supporting the print template. If the query string is not specified, the currently active query string in the view is applied to expose the relevant records. |
| caption | (Optional) Specifies whether the captions in the form applets are suppressed. Valid values are TRUE (default) and FALSE. This parameter does not apply to list applets. Column headings in list applets are always enabled. |

| Parameter | Description |
|---------------|--|
| headerdivider | <p>(Optional) Specifies whether a divider is printed below the column heading in list applets.</p> <p>Valid values are TRUE and FALSE (default). This parameter does not apply to form applets.</p> |
| recorddivider | <p>(Optional) Specifies whether a horizontal divider is printed between records in list applets.</p> <p>Valid values are TRUE and FALSE (default). This parameter does not apply to form applets.</p> |
| wrap | <p>(Optional) Specifies whether to enable word wrap for text field values.</p> <p>Valid values are TRUE and FALSE (default). This parameter does not apply to numeric fields.</p> |
| grid | <p>(Optional) Specifies whether a grid is used to display data in list applets.</p> <p>Valid values are TRUE and FALSE (default). If TRUE, a grid is drawn between fields, rows, and columns in list applets. This parameter does not apply to form applets.</p> |
| linewidth | <p>(Optional) Specifies grid thickness in millimeters.</p> <p>Valid values are 1-5. The default value is 1, which produces a line 0.2 mm wide. Increasing the value by 1 increases the width by 0.2 mm. A 5 produces a line that is 1 mm wide.</p> |
| offset | <p>(Optional, absolute mode-only) Specifies the distance, in millimeters, to the right from the left margin. This value can be a number ("40") or a variable that contains a number value ("_X" or "posx").</p> |
| width | <p>(Optional) Specifies the width of the applet in millimeters.</p> |
| sql | <p>(Optional) Where clause for the SQL statement. For example, <code>sql="\status"=\'good\'</code> selects all records where status field is set to good.</p> <p>NOTE: Enclose the field name in double quotes and use backslash followed by double quotes to include a double quote within the parameter, then enclose the value in single quotes.</p> |
| maxlines | <p>(Optional) Specifies the maximum number of records to print for list applets. This parameter does not apply to form applets.</p> |
| startat | <p>(Optional) Starts printing from the specified record in the row set.</p> |

Example

The following example prints the same applet twice side-by-side starting at record 10 in the list:

- The first time CS HH Product Bucket List Applet is printed, it is printed with gridlines of 0.4 mm in thickness, and 0 mm from the left edge of the printable area of the page (on the left margin).
Only records where Status=Defective are displayed. Up to two records are printed (Maxlines="2").
- The second time CS HH Product Bucket List Applet is printed, it is printed with gridlines where the left side of the applet is positioned 80 mm from the left margin.
Only records where Status=Good are displayed. Up to two records are printed (Maxlines="2").

```
<TAG: Applet name="CS HH Product Bucket List Applet" sql="\Status\"='Defective'" maxlines="2"
startat="10" recorddivider="FALSE" wrap="TRUE" grid="TRUE" offset="0" width="75"
linewidth="0.4">
```

```
<TAG: SETVARIABLE variable="var_equal" name="">
<TAG: SETVARIABLE variable="var_fieldname" name="\Status\">
<TAG: SETVARIABLE variable="var_criteria" name="Good">
<TAG: CONCAT variable="var_equal" prepend="<variable>var_fieldname"
append="<variable>var_criteria">
<TAG: SETY y="0">
```

```
<TAG: Applet name="CS HH Product Bucket List Applet" sql="<variable>var_equal" maxlines="2"
startat="10" recorddivider="FALSE" wrap="TRUE" grid="TRUE" offset="80" width="75">
```

Cell Tag

The Cell tag prints the string stored in the name="STRING" parameter or variable="VARIABLE NAME" parameter at the specified location on the current line.

Usage

Use the Cell tag to create cells of data in a row in a table. Use it with the EndOfLine tag. If the data exceeds the width of the cell specified by the Startat and Endat parameters, then the data wraps to the next line.

Syntax

To use this tag, use the following syntax:

```
<TAG: CELL startat="START" endat="END" name="STRING" align="ALIGNMENT" left="LEFT"
top="TOP" right="RIGHT" bottom="BOTTOM" box="BOX" linewidth="LINEWIDTH">
<TAG: CELL startat="START" endat="END" variable="VARIABLE NAME" align="ALIGNMENT"
left="LEFT" top="TOP" right="RIGHT" bottom="BOTTOM" box="BOX" linewidth="LINEWIDTH">
```

<TAG:EndOfLine>

| Parameter | Description |
|-----------|--|
| TAG:Cell | (Required) Required tag indicator. |
| startat | (Required in relative mode) Specifies the starting point of the cell, expressed as a percentage of the printable page width. For example, if START =10, then the cell starts at the point that is 10% from the left edge of the printable width. |
| endat | (Required in relative mode) Specifies the end point of the cell, expressed as a percentage of the printable page width. For example, if END=50, then the cell ends at the point that is 50% from the left edge of the printable width. |
| name | (Optional) String value. If a variable parameter is not specified, then name is a required parameter. |
| variable | (Optional) Variable where the value is stored. If a name parameter is not specified, then variable is a required parameter. |
| align | (Optional, Existing) Specifies the alignment of the string. Valid values are LEFT (default), CENTER, and RIGHT. |
| left | (Optional, absolute mode-only tag) Specifies the distance, in millimeters, of the left side of the cell from the left margin. If omitted, the current head position is used and RIGHT parameter is used to specify the width of the cell. |
| top | (Optional in absolute mode) Specifies the distance, in millimeters, of the top side of the cell from the top margin. If omitted, the current head position is used and the BOTTOM parameter is used to specify the height of the cell. |
| right | (Required in absolute mode) Specifies the distance, in millimeters, of the right side of the cell from the left margin. |
| bottom | (Required in absolute mode) Specifies the distance, in millimeters, of the bottom side of the cell from the top margin. |
| box | (Optional) Specifies whether to draw a box around the printed cell data in form applets. Valid values are TRUE and FALSE (default). This parameter does not apply to list applets. |
| linewidth | (Optional in absolute mode) Specifies the thickness, in millimeters, of the line drawn if BOX is set to TRUE. If omitted, line width is 0.2 millimeters. |

Example

The following example prints "Text In A Box" inside a rectangular box 10 millimeters (mm) long x 30 mm wide, which is 200 mm from the top margin, and 10 mm from the left margin.

```
<TAG:Cell name="Text In A Box" top="200" left="10" bottom="210" right="60" box="TRUE"
linewidth="0.1" align="LEFT">
<TAG:EndOfLine>
```

Picture Tag

The Picture tag specifies the graphic files that are included in the print template. When used in absolute print mode (with absolute mode-only tags), graphics are printed from the specified file at the specified position, and according to the specified dimensions.

Usage

Use the Picture tag to add graphic images, such as a company logo, to your documents. Supported file formats are BMP and JPG.

Graphics files are located in the `\Program Files\Siebel Handheld\templates` directory on the handheld device. Specify the full path name of the file for the Name parameter. Scaling or graphics operations, such as inverse image, are not supported.

The *alignment* and *position* parameters are used together to place the graphic. Alignment specifies the left edge, the center, or the right edge of the graphic. Position is the distance, specified in millimeters, from the left edge of the paper.

Example

- If `align="Left"` and `position="30"`, then the left edge of the graphic is set 30 mm from the left edge of the paper.
- If `align="Center"` and `Position="30"`, the center of the graphic is set 30 mm from the left edge of the paper.
- If `align="Right"` and `Position="30"`, the right edge of the graphic is set 30 mm from the left edge of the paper.

If the position is not specified, the position defaults to 0. Depending on the alignment setting, the graphic is left-aligned (`align="Left"`), right-aligned (`align="Right"`), or centered (`align="Center"`).

- In absolute mode-only, if `left="20"`, `top="40"`, `right="80"`, and `bottom="80"`, then the picture is drawn 20 mm from the left margin and 40 mm from the top margin, and is 60 x 40 millimeters in size.
- In the absolute mode-only, if `right="80"` and `bottom="80"`, then the picture is drawn at the current location and is 80 x 80 mm in size.

When `left="x"` and `top="y"` parameters are not specified, the picture is drawn at the current head position. In such cases, the `right="x"` and `bottom="y"` parameters specify the width and height of the image respectively.

Syntax

To use this tag, use the following syntax:

```
<TAG:Picture name="FILE NAME" align="ALIGNMENT" position="POSITION">
```

| Parameter | Description |
|-------------|----------------------------------|
| TAG:Picture | Required. Picture tag indicator. |

| Parameter | Description |
|-----------|---|
| name | (Required) Specifies the full path and name of the graphics file. |
| align | (Optional) Specifies the left edge, the center, or the right edge of the graphic. Used with the Position parameter to specify the horizontal position of the graphic. Valid values are LEFT (default), CENTER, and RIGHT. |
| position | (Optional) Specified in millimeters, this is the distance from the left edge of the paper. Used with the Alignment parameter to specify the horizontal position of the graphic. The default is 0. |

Example

```
<TAG:Picture name="\Program Files\Siebel Handheld\templates\logo.bmp" alignment="Right" Position="88">
```

Syntax (absolute mode-only)

To use this tag, use the following syntax:

```
<TAG:Picture name="FULL PATH FILE NAME" left="LEFT VALUE" right="RIGHT VALUE" top="TOP VALUE" bottom="BOTTOM VALUE" position="POSITION">
```

| Parameter | Description |
|-------------|--|
| TAG:Picture | (Required) Picture tag indicator. |
| left | (Optional, absolute mode-only tag) Specified in millimeters, this is the distance of the left side of the rectangle from the left margin. If omitted, the current head position is used, and the RIGHT parameter is used to specify the width. |
| top | (Optional, absolute mode-only tag) Specified in millimeters, this is the distance of the top side of the rectangle from the top margin. If omitted, the current head position is used, and the BOTTOM parameter is used to specify the height. |
| right | (Required, absolute mode-only tag) Specified in millimeters, this is the position of the right side of the rectangle on the X axis from the left margin. |
| bottom | (Required, absolute mode-only tag) Specified in millimeters, this is the position of the bottom side of the rectangle on the Y axis from the top margin. |

Example (absolute mode-only)

```
<TAG:Picture name="\Program Files\Siebel Handheld\templates\logo.bmp" left="20" top="40" right="80" bottom="80">
```

```
<TAG:Picture name="\Program Files\Siebel Handheld\templates\logo.bmp" right="80" bottom="80">
```

Static Text Tag

You can add static text anywhere in your template by using the current Format settings. The text goes across the entire page width. If the text exceeds the width of the page, it wraps to the next line.

Example

This quote is valid for 60 days from the date on this quotation.

Print Formatting Tags

The following tags provide flexibility in formatting print templates and are used to specify the format of the print output:

- [“Background Tag” on page 341](#)—Prints an image to specified coordinates, much like a watermark.
- [“Comment Tag” on page 342](#)—Adds comments that do not appear in the final document.
- [“Divider Tag” on page 343](#)—Adds visual lines to visually separate different parts of the document.
- [“DrawRect Tag” on page 344](#)—Draws a rectangle at specified coordinates.
- [“EndOfPage Tag” on page 345](#)—Advances to a new page.
- [“Footer Tag” on page 346](#)—Adds information such as page numbers, date, and time.
- [“Format Tag” on page 346](#)—Specifies characteristics such as font, boldface, italics, and underline.
- [“Frame Tag” on page 347](#)—Draws a surrounding frame around the whole page for all pages in the print template.
- [“Header Tag” on page 348](#)—Adds information such as page numbers, date, and time.
- [“Landscape Tag” on page 349](#)—Sets landscape paper orientation for the subsequent print tags.
- [“List Tag” on page 349](#)—Prints the selected columns with the specified width.
- [“LineTo Tag” on page 351](#)—Draws a line of a specified line thickness from the current printer head position to the specified coordinate.
- [“Margin Tag” on page 352](#)—Sets the margin for a specific template.
- [“PageBreak Tag” on page 353](#)—Specifies a break in the text, forcing the text that follows to appear on the next page.
- [“Portrait Tag” on page 353](#)—Sets portrait paper orientation for the subsequent print tags.
- [“Title Tag” on page 354](#)—Specifies a name for the template that appears in the print job list.

Background Tag

The Background tag prints an image at specified coordinates for all pages in the print template. Supported file formats are BMP and JPG only.

Usage

Use the Background tag to print logos or images in the header, footer, or main printable area of the print template. When the Background tag is used, the image is printed like a watermark logo, and any other object included in the print template is printed over it.

It is recommended that you declare the Background tag immediately after the Absolute, Margin, Header, or Footer tags.

Syntax

To use this tag, use the following syntax:

```
<TAG:Background name="FILE NAME" left="LEFT" top="TOP" right="RIGHT" bottom="BOTTOM">
```

| Parameter | Description |
|----------------|--|
| TAG:Background | (Required) Required tag indicator. |
| name | (Required) Specifies the full path and name of the graphics file. |
| left | (Optional) Specifies, in millimeters, the position of the left side of the image on the X axis from the left side of the paper. This setting defaults to zero if omitted. |
| top | Specifies, in millimeters, the position of the top side of the image on the Y axis from the top of the paper. This setting defaults to zero if omitted. |
| right | Specifies, in millimeters, the position of the right side of the image on the X axis from the right side of the paper. The image is shrunk or expanded to fit within the specified size. |
| bottom | Specifies, in millimeters, the position of the bottom side of the paper on the Y axis from the bottom of the paper. The image is shrunk or expanded to fit within the specified size. |

Example

The following example prints a 50 x 30 mm image at 30 mm from the left edge and 50 mm from the top edge of the page. The header or footer or both can overlap the image if the image is placed on the header or footer area:

```
<TAG:Absolute>
<TAG:Margin left="30" Top="50" right="80" bottom="100">
<Tag:Header left="<page>" center="<variable>numrows" right="<date>">
<Tag:Footer left="receipt" center="<variable>numrows" right="<page>">
<Tag:Background name="\logo.gif" left="30" top="50" right="80" bottom="100">
```

Comment Tag

The Comment tag adds code comments to the print template.

Usage

Use the Comment tag to add explanatory notes or documentation to explain and maintain the print template. Comment tags are ignored during printing and do not appear in the printed document.

Syntax

To use this tag, use the following syntax:

```
<REM:COMMENT>
```

| Parameter | Description |
|-----------|-----------------------------------|
| REM | (Required) Comment tag indicator. |
| COMMENT | (Optional) Comment text. |

Example

```
<REM: Quotation Print Template. This template includes four applets. >
```

Divider Tag

The Divider tag renders a horizontal line across the width of the page.

Usage

Use the Divider tag to visually separate the different sections of your document. The Divider tag can be used between other tags; therefore, you can add a line between two applets. However, you cannot embed a Divider within an applet.

Syntax

To use this tag, use the following syntax:

```
<TAG:Divider weight="WEIGHT" startat="START" endat="END">\
```

| Parameter | Description |
|-------------|---|
| TAG:Divider | (Required) Divider tag indicator. |
| weight | (Optional) Specifies line thickness. Valid values are 1–5. The default value is 1, which produces a line 0.2 mm wide. Increasing the value by 1 increases the width by 0.2 mm. A 5 produces a line that is 1 mm wide. |
| startat | (Optional) Percentage that specifies the starting point of the line. The default is 0%, the farthest left point. |
| end | (Optional) Percentage that specifies the end point of the line. The default is 100%, the farthest right point. |

Example

```
<TAG:Divider weight="2" startat="20" endat="80">
```

DrawRect Tag

The DrawRect tag draws a rectangle at the specified coordinates. This is an absolute mode-only tag.

Usage

Use the DrawRect tag to draw a rectangle anywhere in the printable area of your page. This tag can be used with the Applet tag to draw a rectangle around a form applet.

Syntax

To use this tag, use the following syntax:

```
<TAG:DrawRect left="LEFT" right="RIGHT" top="TOP" bottom="BOTTOM" linewidth="THICKNESS">
```

| Parameter | Description |
|--------------|---|
| TAG:DrawRect | (Required) Applet tag indicator. |
| left | (Optional) Specifies the distance, in millimeters, of the left side of the rectangle from the left margin. If omitted, the current head position is used and the RIGHT parameter is used to specify the width. |
| top | (Optional) Specifies the distance, in millimeters, of the top side of the rectangle from the top margin. If omitted, the current head position is used and the Bottom parameter is used to specify the height. |
| right | (Required) Position, in millimeters, of the right side of the rectangle on the X axis from the left margin. |
| bottom | (Required) Position, in millimeters, of the bottom side of the rectangle on the Y axis from the top margin. |
| linewidth | (Required) The thickness, in millimeters, of the line. |

Example

The following example draws a thin 100 x 90 mm rectangle at 30 mm from the top and 20.5 mm from the left edge of the printable area:

```
<TAG:DrawRect left="20.5" top="30" right="120.5" bottom="120" linewidth="0.2">
```


EndOfLine Tag

The EndOfLine tag specifies the end of a row and advances to the next line.

Usage

Use the EndOfLine tag to create rows of data in a table. Used with the Cell tag.

Syntax

To use this tag, use the following syntax:

```
<TAG:EndOfLine>
```

| Parameter | Description |
|---------------|---------------------------------------|
| TAG:EndOfLine | (Required) End of line tag indicator. |

Example

```
<TAG:Cell startat="0" endat="50" name="TOTAL:" align="RIGHT">
<TAG:Cell startat="50" endat="100" variable="vvv1total" align="RIGHT">
<TAG:EndOfLine>
```

EndOfPage Tag

The EndOfPage tag advances to a new page.

Usage

Use the EndOfPage tag to create a page break and start a new page within the print template.

Syntax

To use this tag, use the following syntax:

```
<TAG:EndOfPage>
```

| Parameter | Description |
|---------------|---------------------------------------|
| TAG:EndOfPage | (Required) End of page tag indicator. |

Example

```
<TAG:Cell startat="0" endat="50" name="TOTAL:" align="RIGHT">
<TAG:Cell startat="50" endat="100" variable="vvv1total" align="RIGHT">
<TAG:EndOfPage>
```

Footer Tag

The Footer tag adds footer information such as page numbers, the current date, and the current time.

Usage

You might, for example, always want the text *Thank you for your business!* to appear on the bottom of your receipts. If so, you can include this text in a Footer tag.

The Footer tag can be specified anywhere in the template file; however, it will always appear at the very bottom of your document. Only one Footer tag is applied to the document. If you have multiple Footer tags in the file, only the last Footer tag is used; any other Footer tags are ignored.

If you include a Footer tag in your template, then you must set the BottomMargin in setup.ini to a minimum of 20.

The footer has three sections, Left, Center, and Right, which are used to position the information in the footer.

The text in the footer is formatted in 10 point, Helvetica font. If Helvetica is not available on your system, the default font is used. You cannot customize text formatting of the Footer tag. The Format tag does not affect the formatting of the footer.

Syntax

To use this tag, use the following syntax:

```
<TAG:Footer left="DATA" center="DATA" right="DATA">
```

| Parameter | Description | | | | | | | | |
|------------------------|---|----------|-------------|--------|-------------|--------|----------------------|--------|----------------------|
| TAG:Footer | (Required) Footer tag indicator. | | | | | | | | |
| left, center, or right | (Optional) Static text or one of the following data tags: | | | | | | | | |
| | <table border="1"> <thead> <tr> <th>Data tag</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><page></td> <td>Page number</td> </tr> <tr> <td><date></td> <td>Date of the printing</td> </tr> <tr> <td><time></td> <td>Time of the printing</td> </tr> </tbody> </table> | Data tag | Description | <page> | Page number | <date> | Date of the printing | <time> | Time of the printing |
| Data tag | Description | | | | | | | | |
| <page> | Page number | | | | | | | | |
| <date> | Date of the printing | | | | | | | | |
| <time> | Time of the printing | | | | | | | | |

Example

```
<TAG:Footer Left="<date>" Right="<page>">
```

Format Tag

The Format tag specifies the characteristics of the text—for example, the typeface or font, whether the text is in boldface, italics, or underlined, and the text alignment.

Usage

Use the Format tag to specify the typeface and font size of the text in your template. The Format tag applies to the text that immediately follows it, or up until the next Format tag appears.

When an applet is rendered, the field alignment specification takes precedence over the align="TEXT ALIGNMENT" parameter of the Format tag.

Syntax

To use this tag, use the following syntax:

```
<TAG:Format face="FONT FACE" size="FONT SIZE" bold="BOLD" italic="ITALIC"
underline="UNDERLINE" align="TEXT ALIGNMENT">
```

| Parameter | Description |
|------------|---|
| TAG:Format | (Required) Format tag indicator. |
| face | (Required) Specifies the typeface of the text. Any typeface installed on the handheld device can be specified. There is no default for this parameter. |
| size | (Required) Specifies the text size. Any installed font size for an installed typeface on the handheld device can be specified. There is no default for this parameter. |
| bold | (Optional) Specifies whether the text is in bold. Valid values are TRUE or FALSE (default). If TRUE, text is set to bold. |
| italic | (Optional) Specifies whether the text is in italic. Valid values are TRUE or FALSE (default). If TRUE, text is set to italic. |
| underline | (Optional) Specifies whether the text is underlined. Valid values are TRUE or FALSE (default). If TRUE, text is set to underline. |
| align | (Optional) Specifies the text alignment. Valid values are LEFT (default), RIGHT, and CENTER. |

Example

```
<TAG:Format face="Helvetica" size="10">
```

Frame Tag

The Frame tag draws a frame (rectangular box) around all pages in the template.

Usage

Use the Frame tag to draw a rectangular box around each page in your template. The frame is drawn at the edge of the page, just inside the paper margins.

Place the Frame tag after the Margin, Header, or Footer tags.

Syntax

To use this tag, use the following syntax:

```
<TAG:Frame pageframe="BOOLEAN VALUE" linewidth="THICKNESS">
```

| Parameter | Description |
|-----------|--|
| TAG:Frame | (Required) Required tag indicator. |
| pageframe | (Required) Specifies whether to draw a frame around the page. Valid values are TRUE or FALSE. If set to TRUE, a frame is drawn around each page. |
| linewidth | (Required) Specifies the thickness of the line in millimeters. If omitted, the default value is 0.3 millimeters. |

Example

```
<TAG:Frame pageframe="TRUE" linewidth="0.5">
```

Header Tag

The Header tag adds header information such as page numbers, the current date, and the current time to the template.

Usage

You might always want the current date and time to appear on your invoices. If so, you can include this information in a Header tag.

The Header tag can be specified anywhere in the template file; however, it always appears at the very top of your document. Only one Header tag is applied to the document. If you have multiple Header tags in the file, only the last Header tag is used; all other Header tags are ignored.

If you include a Header tag in your template, then you must set TopMargin in setup.ini to a minimum of 20.

The Header tag has three sections (Left, Center, and Right) which are used to position the information in the header. The Left parameter aligns the text with the left margin, and the Right parameter right aligns the text with the right margin. The Center parameter centers the text in the header.

The text in the header is formatted in 10 point, Helvetica font. If Helvetica is not available on your system, the default font is used. You cannot customize text formatting of the Header tag. The Format tag does not affect the format of the header.

Syntax

To use this tag, use the following syntax:

```
<TAG:Header left="DATA" center="DATA" right="DATA">
```

| Parameter | Description | | | | | | | | |
|------------------------|---|----------|-------------|--------|-------------|--------|----------------------|--------|----------------------|
| TAG:Header | (Required) Header tag indicator. | | | | | | | | |
| left, center, or right | (Optional) Static text or one of the following data tags: | | | | | | | | |
| | <table border="1"> <thead> <tr> <th>Data tag</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><page></td> <td>Page number</td> </tr> <tr> <td><date></td> <td>Date of the printing</td> </tr> <tr> <td><time></td> <td>Time of the printing</td> </tr> </tbody> </table> | Data tag | Description | <page> | Page number | <date> | Date of the printing | <time> | Time of the printing |
| Data tag | Description | | | | | | | | |
| <page> | Page number | | | | | | | | |
| <date> | Date of the printing | | | | | | | | |
| <time> | Time of the printing | | | | | | | | |

Example

```
<TAG:Header center="Company Confidential">
```

Landscape Tag

The Landscape tag sets the page orientation (or printout mode) to landscape for the subsequent page or print tags.

Usage

The Landscape tag must be placed after the Margin, Header, or EndOfpage tags. If the Landscape tag is specified in the template, users cannot override the orientation even if they change the paper orientation setting in the printer settings dialog box.

Syntax

To use this tag, use the following syntax:

```
<TAG:Landscape>
```

| Parameter | Description |
|---------------|-------------------------------------|
| TAG:Landscape | (Required) Landscape tag indicator. |

List Tag

The List tag prints the selected columns in a specified column width.

Usage

Use the List tag to print out a list of records containing specified fields.

Syntax

To use this tag, use the following syntax:

```
<TAG:List appletname="APPLET" columns="COLUMN1|COLUMN2|...|COLUMNx"
columnwidths="Col1Width|Col2Width|...|ColxWidth" header="HEADER"
headerdivider="HEADERDIVIDER" recorderdivider="RECORDDIVIDER" query="QUERY" sql="SQL"
sort="SORT" maxlines="MAXLINES" startat="START" wrap="WRAP" grid="GRID"
rowsprinted="ROWSPRINTED" offset="OFFSET" width="WIDTH" linewidth="LINEWIDTH">
```

| Parameter | Description |
|-----------------|--|
| TAG:List | (Required) List tag indicator. |
| appletname | (Required) Applet name as defined in Siebel Tools. |
| columns | (Required) List of field names in applet to be printed in the specified order, separated by a vertical bar (). |
| columnwidth | (Optional) Relative weight of each column width. For example, 1 2 means that the first column is half the width of the second column. |
| header | (Optional) Specifies whether to suppress or show headers. Valid values are TRUE (default) or FALSE. If FALSE, captions in form applets are suppressed. This parameter applies only to form applets. |
| headerdivider | (Optional, relative mode-only tag) Specifies whether to print a divider below the heading in a list applet. Valid values are TRUE or FALSE (default). If TRUE, a divider is printed below the column heading in list applets. This parameter applies only to list applets. |
| recorderdivider | (Optional, relative mode-only tag) Specifies whether to print a horizontal divider between records in a list applet. Valid values are TRUE or FALSE (default). If TRUE, a horizontal divider is printed between records in list applets. This parameter applies only to list applets. |
| query | (Optional) Specifies query specification. |
| sql | (Optional) Where clause for the SQL statement; sql="\status"=\='good'" selects all records where status field is set to good. NOTE: Enclose the field name in double quotes and use backslash followed by double quotes to include a double quote within the parameter, then enclose the value in single quotes. |
| sort | (Optional) Specifies sort specification. |
| maxlines | (Optional) The maximum number of records that are printed for this applet. This parameter applies only to list applets. |

| Parameter | Description |
|-------------|--|
| startat | (Optional) The point at which the specified record in the row set starts printing. |
| wrap | (Optional) Specifies whether word wrap is enabled for text field values. Valid values are TRUE or FALSE (default). If TRUE, word wrap is applied to text field values. This setting does not apply to numeric fields. |
| grid | (Optional, absolute mode-only) Specifies whether to print a grid between fields, rows and columns. Valid values are TRUE or FALSE (default). If TRUE, a grid is printed between fields, rows, and columns. |
| rowsprinted | Specifies the number of rows to print for list applets. This parameter does not apply to form applets. |
| offset | (Optional) Specifies the distance, in millimeters, to the right of the left margin. This can be a number ("40") or a variable that contains a number value ("_X" or "posx"). This is an absolute-mode only parameter. |
| width | (Optional, absolute mode-only) Specifies the width of the applet in millimeters. |
| linewidth | (Optional, absolute mode-only) Specifies the line grid thickness in millimeters. |

Example

The following example prints out the records (30 x 70 mm respectively) from the CS HH Product Bucket List Applet containing the Status and Product Name fields:

```
<TAG:List appletname="CS HH Product Bucket List Applet" columns="Status|Product Name"
columnwidths="30|70" maxlines="2" line="False" wrap="TRUE" rowsprinted="numrows"
grid="TRUE" linewidth="0.5" offset="10" width="75">
```

LineTo Tag

The LineTo tag draws a line of a specified thickness from the current printer head position to a specified coordinate.

Usage

Use the LineTo tag along with the GetX, GetY, SetX, SetY, and Offset tags to draw lines anywhere in the printable area of your template.

Syntax

To use this tag, use the following syntax:

<TAG:LineTo x="END X" y="END Y" linewidth="THICKNESS">

| Parameter | Description |
|------------|--|
| TAG:LineTo | (Required) Applet tag indicator. |
| x | (Required) Position, in millimeters, on the X axis where the line ends from the left margin. |
| y | (Required) Position, in millimeters, on the Y axis where the line ends from the top margin. |
| linewidth | (Required) Thickness, in millimeters, of the line. |

Example

The following example draws a vertical line of 1 mm thickness from (100,100) to (100,140), and a thin horizontal line from (100,140) to (120,140):

```
<TAG:SetX x="100">
<TAG:SetY y="100">
<TAG:LineTo x="100" y="140" linewidth="1">
<TAG:LineTo x="120" y="140" linewidth="0.2">
```

Margin Tag

The Margin tag sets all margin values for all pages in the print template.

Usage

Use the Margin tag to set all four page margin values. Use at the top of a page if you want to alter margins.

As an absolute mode-only tag, the Margin tag must be declared immediately after the Absolute tag. Any string or image printed on the page before a Margin tag is discarded.

Syntax

To use this tag, use the following syntax:

```
<TAG:Margin left="LEFT" top="TOP" right="RIGHT" bottom="BOTTOM">
```

| Parameter | Description |
|------------|--|
| TAG:Margin | (Required) Applet tag indicator. |
| left | (Required) Specifies the position, in millimeters, of the left side of the printable area from the left edge of the paper. |
| top | (Required) Specifies the position, in millimeters, of the top side of the printable area from the top edge of the paper. |

| Parameter | Description |
|-----------|--|
| right | (Required) Specifies the position, in millimeters, of the right side of the printable area from the right edge of the paper. |
| bottom | (Required) Specifies the position, in millimeters, of the bottom of the printable area from the bottom edge of the paper. |

Example

The following example sets the margins at 30 mm from the left and right paper edge, and 50 mm from the top and bottom paper edge:

```
<TAG:Title title="Inventory Report">
<TAG:Absolute>
<TAG:Margin left="30" top="50" right="30" bottom="50">
```

PageBreak Tag

The PageBreak tag forces text that follows the tag to appear on the next page.

Usage

Use the PageBreak tag for multiple page documents or forms.

Syntax

To use this tag, use the following syntax:

```
<TAG:PageBreak>
```

| Parameter | Description |
|---------------|--------------------------------------|
| TAG:PageBreak | (Required) Page break tag indicator. |

Example

```
<TAG:PageBreak>
```

Portrait Tag

The Portrait tag sets the page orientation (or printout mode) to portrait for the subsequent page or print tags.

Usage

The Portrait tag must be placed after the Margin, Header, or Endofpage tags. If the Portrait tag is specified in the template, users cannot override the orientation even if they change the paper orientation setting in the printer settings dialog box.

Syntax

To use this tag, use the following syntax:

```
<TAG:Portrait>
```

| Parameter | Description |
|--------------|------------------------------------|
| TAG:Portrait | (Required) Portrait tag indicator. |

Title Tag

The Title tag is the name used to identify the template.

Usage

The Title tag is a required tag in your print template file.

When you print your document, the Title tag appears in the print job list, but does not appear on the document that is printed. Use static text to add a document title to your template.

Syntax

To use this tag, use the following syntax:

```
<TAG:Title title="TITLE">
```

| Parameter | Description |
|-----------|---|
| TAG:Title | (Required) Title tag indicator. |
| title | (Required) Title text can be between 1 to 255 characters. |

Example

```
<TAG:Title title="Quotation">
```

Get and Set Tags

You can add static text to the print template and also set variables. The following print tags allow you to calculate additional information and add it to your reports:

- ["GetCount Tag" on page 355](#)—Calculates the number of records.
- ["GetDate Tag" on page 356](#)—Gets the current date.

- [“GetDirect Tag” on page 356](#)—Returns the business component fields for any specified business component.
- [“GetField Tag” on page 357](#)—Gets the value of a field.
- [“GetRegistry Tag” on page 358](#)—Returns registry values.
- [“GetTime Tag” on page 359](#)—Gets the current time.
- [“GetTotal Tag” on page 359](#)—Calculates the total for a column of data.
- [“SetVariable Tag” on page 360](#)—Assigns a value to a variable.

All of this information is stored in variables that you specify. You can present this information in a report using the Cell and EndofLine tags to create tables of information. For more information about how variable parameters can be used to manipulate data output, see [“Variable Parameter Tags” on page 369](#). For an example of how variables can be used to present information in print templates, see [“Using Variables in Print Templates” on page 369](#).

GetCount Tag

The GetCount tag calculates the number of records in the selected rows and assigns this value to a variable.

Usage

Use the GetCount tag to calculate the number of records and print this value in your report.

Syntax

To use this tag, use the following syntax:

```
<TAG:GetCount appletname="APPLET NAME" query="QUERY STRING" variable="VARIABLE NAME">
```

| Parameter | Description |
|--------------|--|
| TAG:GetCount | (Required) Required tag indicator. |
| appletname | (Required) Name of the applet. |
| query | (Optional) Search specification for the rows to be included in the row set. If not specified, the current search specification is used. The syntax for the query string is: Field Name Query Statement. You can specify one or more query strings separated by a vertical bar (). For example, Query="Name LIKE 'Sieb*'" "Location LIKE 'H*'" |
| variable | (Required) Name of variable where the value is stored. |

Example

```
<TAG:GetCount appletname="CS HH Product Bucket List Applet" query="Status|Good" variable="StatusGood">
```

GetDate Tag

The GetDate tag obtains the current date, offsets the date by the number of days specified in the dayoffset parameter, and stores the result in a variable.

Usage

Use the GetDate tag to print a specific date in your report.

Syntax

To use this tag, use the following syntax:

```
<TAG:GetDate dayoffset="NUMBER OF DAYS" variable="VARIABLE NAME" longformat="DATE FORMAT">
```

| Parameter | Description |
|-------------|---|
| TAG:GetDate | (Required) Required tag indicator. |
| dayoffset | (Optional) Number of days by which the current date is offset. For example, –100 subtracts 100 days from the current date; 100 adds 100 days to the current date. The default is 0. |
| variable | (Required) Name of the variable where the value is stored. |
| longformat | (Optional) TRUE prints the date in long date format (for example, Monday, February 11, 2002). FALSE prints the short date format (for example, 2/21/02). The default is FALSE. |

Example

```
<TAG:GetDate longformat="TRUE" variable="LongDate">
```

GetDirect Tag

The GetDirect tag returns the data in any business component of the Siebel Handheld application configuration, and stores it in the specified variable.

Usage

Use GetDirect to obtain the field data in any business component of your Siebel Handheld application configuration. GetDirect works the same way as GetFieldData, except that it specifies the business object name and business component name for the appropriate table. GetDirect is similar to the GetField tag in that the specified

field value of the first record is stored in the specified variable.

Syntax

To use this tag, use the following syntax:

```
<TAG: GetDirect businessobject="BUSINESSOBJECT" businesscomponent="BUSINESSCOMPONENT"
fieldname=<"FIELD" variable="VARIABLE" query="QUERY" sort="SORT" sql="SQL">
```

| Parameter | Description |
|-------------------|--|
| TAG: GetDirect | (Required) Required tag indicator. |
| businessobject | (Required) Business object name. |
| businesscomponent | (Optional) Business component name. |
| fieldname | (Required) Field name. |
| variable | (Required) Variable to store the data. |
| query | (Optional) Query specification. |
| sort | (Optional) Sort specification. |
| sql | (Optional) Where clause for SQL statement; sql="\`status\`='good'" selects all records where status field is set to good. NOTE: Enclose the field name in double quotes and use backslash followed by double quotes to include a double quote within the parameter, then enclose the value in single quotes. |

Example

The following example returns the Name field of the first record from the Account business component which is part of the Account business object, and puts it into the variable called getdirectresult1:

```
<TAG: GetDirect businessobject="Account" businesscomponent="Account" fieldname="Name"
variable="getdirectresult1" sql="var_equal">
```

GetField Tag

The GetField tag returns the value of the first record (in the first row of selected rows) in a specified column and assigns the result to a variable.

Usage

Use the GetField tag to print a value without developing a print applet. Design the query and sort strings so that the desired record is the first record in the selected rows (or column).

GetField only allows you to obtain the value in the first record.

Syntax

To use this tag, use the following syntax:

```
<TAG:GetField appletName="APPLET NAME" fieldname="FIELD NAME" query="QUERY STRING"
sort="SORT STRING" variable="VARIABLE NAME">
```

| Parameter | Description |
|--------------|--|
| TAG:GetField | (Required) Required tag indicator. |
| appletname | (Required) Name of the applet. |
| fieldname | (Required) Name of the field or column. |
| query | (Optional) Search specification for the rows to be included in the total. If not specified, the current search specification is used. |
| sort | (Optional) Sort specification. If not specified, the current sort specification is used. The syntax is: Field Name [ASC][DESC]. The default is ASC (Ascending). For example: Sort="Account DESC, Date". In the example, the Account field is sorted in descending order, then the date field is sorted in ascending order. |
| variable | (Required) Name of variable where the value is stored. |

Example

```
<TAG:GetField appletname="CS HH Product Bucket List Applet" fieldname="Quantity"
variable="ProdBucketListField">
```

GetRegistry Tag

The GetRegistry tag obtains a value in the registry and assigns it to a variable.

Usage

Use GetRegistry to print the contents of the registry key in your report.

Syntax

To use this tag, use the following syntax:

```
<TAG:GetRegistry name="REGISTRY NAME" variable="VARIABLE NAME">
```

| Parameter | Description |
|-----------------|--|
| TAG:GetRegistry | (Required) Required tag indicator. |
| name | (Required) Value in the registry. |
| variable | (Required) Name of the variable where the value is stored. |

Example

```
<TAG:GetRegistry name="InstallDir" variable="Registry">
```

GetTime Tag

The GetTime tag obtains the current time, offsets the time by the number of seconds specified by the Secondoffset parameter, and assigns the result to a variable.

Usage

Use the GetTime tag to print a specific time in your report.

Syntax

To use this tag, use the following syntax:

```
<TAG:GetTime secondoffset="NUMBER OF SECONDS" variable="VARIABLE NAME">
```

| Parameter | Description |
|--------------|--|
| TAG:GetTime | (Required) Required tag indicator. |
| secondoffset | (Optional) Number of seconds by which the current time is offset. For example, -100 subtracts 100 seconds from the current time; 100 adds 100 seconds to the current time. The default is 0. |
| variable | (Required) Name of variable where the value is stored. |

Example

```
<TAG:GetTime secondoffset="3600" variable="Time">
```

GetTotal Tag

The GetTotal tag calculates the total for a specified column (or for all selected rows) and assigns the value to a variable.

Usage

Use the GetTotal tag to calculate a column total and display the total in your report.

Syntax

To use this tag, use the following syntax:

```
<TAG:GetTotal appletname="APPLET NAME" fieldName="FIELD NAME" query="QUERY STRING"
variable="VARIABLE NAME">
```

| Parameter | Description |
|--------------|--|
| TAG:GetTotal | (Required) Required tag indicator. |
| appletname | (Required) Name of the applet. |
| fieldname | (Required) Name of the field or column. Specify a field or column with numeric data. If you specify a field with non-numeric data, a 0 is assigned to the variable. |
| query | (Optional) Search specification for the rows to be included in the total. If not specified, the current search specification is used. The syntax for the query string is: Field Name Query Statement. You can specify one or more query strings separated by a vertical bar (). For example, Query="Name LIKE 'Sieb*'" "Location LIKE 'H*'" |
| variable | (Required) Name of variable where the value is stored. |

Example

```
<TAG:GetTotal appletname="CS HH Product Bucket List Applet" fieldName="Quantity"
variable="ProdBucketListTotal">
```

SetVariable Tag

The SetVariable tag sets a value to a variable.

Usage

Use the SetVariable tag to assign a value to a variable. You can then print the value by specifying the variable in the Cell tag.

Syntax

To use this tag, use the following syntax:

```
<TAG:SetVariable name="VALUE" variable="VARIABLE NAME">
```

| Parameter | Description |
|-----------------|--|
| TAG:SetVariable | (Required) Required tag indicator. |
| name | (Required) A value. |
| variable | (Required) Name of variable where the value is stored. |

Example

```
<TAG:SetVariable name="100" variable="TotalQuantity">
```

Control Positioning Tags

The following tags can be used to control positioning of printer heads. These tags can be used in conjunction with the Applet (SetY only), Cell, Picture, and LineTo tags to position objects in the printable area.

- ["GetX Tag" on page 361](#)—Returns the position of the printer head along the X axis.
- ["GetY Tag" on page 362](#)—Returns the position of the printer head along the Y axis.
- ["Offset Tag" on page 362](#)—Adds a number to a number value stored in a variable.
- ["SetX Tag" on page 363](#)—Sets the position of the printer head along the X axis.
- ["SetY Tag" on page 364](#)—Sets the position of the printer head along the Y axis.

GetX Tag

The GetX tag returns the printer head position, measured from the left margin (in millimeters) along the X axis, and assigns it to the input variable.

Usage

GetX is an absolute mode-only tag which can be used along with the SetX, SetY, and GetY tags to control the position of the printer head. Use the GetX tag in conjunction with the Cell, Picture, and Lineto tags to position objects in the printable area.

Syntax

To use this tag, use the following syntax:

```
<TAG:GetX variable="VARIABLE NAME">
```

| Parameter | Description |
|-----------|---|
| TAG:GetX | (Required) Applet tag indicator. |
| variable | (Required) Name of the variable to which the X axis value of the current printer head position is stored. |

Example

The following example returns the current X axis position of the printer head and stores it in the posx variable:

```
<TAG:GetX variable="posx">
```

GetY Tag

The GetY tag returns the printer head position, measured from the top margin (in millimeters) along the Y axis, and assigns it to the input variable.

Usage

GetY is an absolute mode-only tag that can be used with the SetX, SetY, and GetX tags to control the position of the printer head. Use the GetY tag in conjunction with the Cell, Picture, and Lineto tags to position objects in the printable area.

Syntax

To use this tag, use the following syntax:

```
<TAG:GetY variable="VARIABLE NAME">
```

| Parameter | Description |
|-----------|---|
| TAG:GetY | (Required) Applet tag indicator. |
| variable | (Required) Name of the variable to which the Y axis value of the current printer head position is stored. |

Example

The following example returns the current Y axis position of the printer head and stores it in the posy variable:

```
<TAG:GetY variable="posy">
```

Offset Tag

The Offset tag adds a value to a number value in a variable.

Usage

Use the Offset tag to perform (positive or negative) offset variable calculations. Offset can be used with SetX and SetY to position the printer head. Special variable names `_X` and `_Y` are reserved variables which indicate the current x and y coordinates of the printer head position. These variable names can be used to offset current printer head coordinates by a specified number value.

Syntax

To use this tag, use the following syntax:

<TAG:Offset variable="VARIABLE NAME" offset="NUMBER VALUE">

| Parameter | Description |
|------------|--|
| TAG:Offset | (Required) Applet tag indicator. |
| variable | (Required) Name of the variable to be offset. _X and _Y can be used to offset the current printer head position. |
| offset | (Required) Number value with which to offset the variable. This can be a positive or negative value. |

Example

The following examples yield the same result. The examples place the printer head position at (+40,-40) from the current head position (_X, _Y). If the current printer head is (100,50), then this means that the new printer head position is (140,10):

```
<TAG:GetX variable="posx">
<TAG:GetY variable="posy">

<TAG:Offset variable="posx" offset="40">
<TAG:Offset variable="posy" offset="-40">

<TAG:SetX x="<variable>posx">
<TAG:SetY y="<variable>posy">

<TAG:Offset variable="_X" offset="40">
<TAG:Offset variable="_Y" offset="-40">
```

SetX Tag

The SetX tag sets the printer head position along the X axis.

Usage

SetX is an absolute mode-only tag that can be used with the GetX, GetY, and SetY tags to control the position of the printer head. Use the SetX tag in conjunction with the Cell, Applet, and Lineto tags to position objects in the printable area.

SetX cannot be used to position static text because full-line text is meant for printing text across the full width of the page. It is recommended that you use the Cell tag to position text in the exact location.

Syntax

To use this tag, use the following syntax:

```
<TAG:SetX x="X">
```

| Parameter | Description |
|-----------|--|
| TAG:SetX | (Required) Applet tag indicator. |
| x | (Required) String value or variable if used with GetX tag specifying where to position the printer head based on its distance to the right, in millimeter from the left margin of the paper. |

Example

The following sets the printer head position to 100 millimeters from the left margin:

```
<TAG:SetX x="100">
```

If using SetX with the GetX and Offset tags, the following example sets the printer head position 40 millimeters to the right of the current printer head position. This means that if the current printer head position is 100 millimeters from the left margin, then the new printer head position is 140 millimeters from the left margin:

```
<TAG:GetX variable="posx">
<TAG:Offset variable="posx" offset="40">
<TAG:SetX x="<variable>posx">
```

SetY Tag

The SetY tag sets the printer head position along the Y axis.

Usage

SetY is an absolute mode-only tag which can be used along with the GetX, GetY, and SetX tags to control the position of the printer head. Use the SetY tag in conjunction with the Cell, Applet, and Lineto tags to position objects in the printable area.

Syntax

To use this tag, use the following syntax:

```
<TAG:SetY y="Y">
```

| Parameter | Description |
|-----------|--|
| TAG:SetY | (Required) Applet tag indicator. |
| y | (Required) String value or variable (if used with GetY tag) specifying where to position the printer head based on its distance to the bottom (in millimeters) from the top margin of the paper. |

Example

The following sets the printer head position to 40 mm from the top margin:

```
<TAG:SetY y="40">
```

If using SetY with the GetY and Offset tags, the following example sets the printer head position 40 mm below the current printer head position. This means that if the current printer head position is 100 mm from the top margin, then the new printer head position is 140 mm from the top margin:

```
<TAG:GetY variable="posy">
<TAG:Offset variable="posy" offset="40">]
<TAG:SetY y="<variable>posy">
```

Data Manipulation and Logic Tags

The following tags are used to manipulate data, perform logical calculations and queries on print data output:

- [“Concat Tag” on page 365](#)—Appends or prepends a string to a string already stored in a variable
- [“Conditional Tags” on page 366](#)—Enable the conditional output of print data
 - NOTE:** Double quote characters—Can be used in values, provided they are preceded by a leading backslash.
- [“sql \(WHERE\) Tag” on page 367](#)—Used in various tags to perform a simple query on the underlying print data
- [“Variable Parameter Tags” on page 369](#)—Are accepted for Applet, Header, and Footer tags

Concat Tag

The Concat tag appends or prepends a value to a string stored in a variable.

Usage

Use the Concat tag to combine string variables or simple string texts. The resulting (concatenated) string is stored in the original variable specified.

Syntax

To use this tag, use the following syntax:

```
<TAG:CONCAT variable= "VARIABLE NAME" prepend="PREPEND" append="APPEND">
```

| Parameter | Description |
|------------|--|
| TAG:Concat | (Required) Applet tag indicator. |
| variable | (Required) Name of the variable to be appended or prepended. The resulting string is stored in this variable |

| Parameter | Description |
|-----------|--|
| prepend | (Optional) Specifies whether to add a simple text string or variable in front of an existing stored variable. The variable must be specified in the form " <code><variable>name of variable</code> ". Prepend is required if Append is not specified. |
| append | (Optional) Specifies whether to add a simple text string or variable to the end of an existing stored variable. The variable must be specified in the form " <code><variable>name of variable</code> ". Append is required if Prepend is not specified. |

Example

The following example sets the variables `var_equal` and `var_criteria`, then appends `var_criteria` and prepends a simple text string to `var_equal`. The concatenated string is stored in `var_equal`:

```
<TAG:SetVariable variable="var_equal" name="">
<TAG:SetVariable variable="var_criteria" name=" 'Good' ">
<TAG:Concat variable="var_equal" prepend="\status\" " append="<variable>var_criteria">
```

Conditional Tags

The conditional tag compares a variable against a criterion according to the operator, and decides to execute either an IF or an ELSE clause.

Usage

Use conditional tags to develop simple if-then logic to control the output of your print data. It is recommended that if-then-else tags are kept simple to ensure that print output is successful.

Syntax

To use this tag, use the following syntax:

```
<TAG:IF variable="VARIABLE NAME" operator="OPERATOR" criteria="CRITERIA" numeric=
"NUMERIC">
  <TAG: ...>
  <TAG: ...>
<TAG:ELSE>
  <TAG: ...>
  <TAG: ...>
```

NOTE: If the comparison is true, the tags immediately following the `<TAG:IF...>` tag are executed; if the comparison is false, the tags immediately following the `<TAG:ELSE...>` tag are executed.

| Parameter | Description |
|-----------|----------------------------------|
| TAG: IF | (Required) Applet tag indicator. |

| Parameter | Description |
|-----------|--|
| TAG: ELSE | (Required) Applet tag indicator |
| variable | (Required) Name of the variable being compared. |
| criteria | String value to which the variable will be compared. |
| operator | (Required) This can be one of the following: <ul style="list-style-type: none"> ■ EQ signifies Equal to ■ GT signifies Greater than ■ GE signifies Greater than or equal to ■ LT signifies Less than ■ LE signifies Less than or equal to |
| numeric | Specifies whether the condition is evaluated as a numeric comparison or a string comparison. Valid values are TRUE (numeric comparison) or FALSE (string comparison). For a string comparison, use the vertical bar () as an OR operator. |

Example

The following example evaluates whether the variable numrows is equal to 15, 17, or 19:

- If true, then a string similar to the following is printed: Printed [numrows] rows that are odd
- If false, then a string similar to the following is printed: Printed [numrows] rows that are even

```
<TAG: I F vari abl e="numrows" oper ator="EQ" cri teri a="15|17|19">
    <TAG: Concat vari abl e="numrows" append=" rows. " prepend="Pri nted ">
    <TAG: Concat vari abl e="numrows" append=" that are odd. ">
<TAG: ELSE>
    <TAG: Concat vari abl e="numrows" append=" rows. " prepend="Pri nted ">
    <TAG: Concat vari abl e="numrows" append=" that are even. ">
<TAG: ENDI F>
```

sql (WHERE) Tag

The sql (WHERE) tag is placed as a parameter, and so can be a value stored in a string variable.

Usage

Use the sql (WHERE) tag, for example, as follows:

```
sql ="\"Status\"=' Defecti ve' " and \"Quanti ty\">500"
```

in the Applet, GetTotal, GetField, GetDirect, and GetCount tags.

Syntax

To use this tag, use the following syntax:

```
SQL="\Status\":"'Defective'"
```

Example 1

```
<TAG: Applet name="CS HH Product Bucket List Applet" sql="\Status\":"' Defecti ve' "
maxlines="2" startat="10" recorddivider="FALSE" wrap="TRUE" grid="TRUE" offset="0"
width="75" linenewidth="0.4">

<TAG: SETVARIABLE variable="var_equal " name="">

<TAG: SETVARIABLE variable="var_fieldname" name="\Status\"">

<TAG: SETVARIABLE variable="var_criteria" name="' Good' ">

<TAG: Concat variable="var_equal " prepend="<variable>var_fieldname" >
append="<variable>var_criteria">

<TAG: SETY y="0">

<TAG: Applet name="CS HH Product Bucket List Applet" sql="<variable>var_equal "
maxlines="2" startat="10" recorddivider="FALSE" wrap="TRUE" grid="TRUE" offset="80"
width="75">
```

Example 2

```
<TAG: Getfield appletname="CS HH Orders Print Applet" fieldname="Account" variable=
"var_name">

<TAG: SETVARIABLE variable="var_where" name="">

<TAG: SETVARIABLE variable="var_fieldname" name="\Name\"">

<TAG: CONCAT variable="var_where" prepend="<variable>var_fieldname" append=
"<variable>var10">

<TAG: CONCAT variable="var_where" append="">

<TAG: GETDIRECT sql="<variable>var_equal " businessobject="Account"
businesscomponent="Account" fieldname="Name" variable="var15">

<TAG: CELL name="<variable>var15" left="10" top="80" right="70" bottom="100"
box="TRUE" linenewidth="0.4" align="RIGHT">
```


Variable Parameter Tags

Most parameters support variables. For example, a user can store a field value and use the field value as one of the parameters in a subsequent tag. Combining this feature with the Concat and Conditional tags and the SQL (WHERE) tag, you can construct complex query statements and print out the results.

Example 1

After the following tag sequence, var_equal contains "\"Status\"='Defective'":

```
<TAG: SETVARIABLE variable="var_equal" name="">
<TAG: SETVARIABLE variable="var_fieldname" name="\\"Status\\"">
<TAG: SETVARIABLE variable="var_criteria" name="' Good' ">
<TAG: CONCAT variable="var_equal" prepend="<variable>var_fieldname" append=
"<variable>var_criteria">
```

NOTE: To include double quotes in a parameter, include a backslash before them.

Example 2

The following example includes a variable in the header tag:

```
<TAG: HEADER left="<page>" center="<variable>numrows" right="<date>">
```

Using Variables in Print Templates

The following example shows how to use variables in a print template.

```
<TAG: Title Title="Inventory Report">
<REM: *****Variable Assignment Examples*****>

<TAG: SetVariable name="THIS IS VARIABLE" variable="VarStr">
<TAG: GetRegistry name="InstallDir" variable="VarReg">
<TAG: GetDate dayoffset="10" variable="VarDate">
<TAG: GetDate longformat="TRUE" variable="VarLongDate">
<TAG: GetTime secondoffset="-3600" variable="VarTime">

<TAG: GetTotal applename="CS HH Product Bucket List Applet" fieldname="Quantity"
variable="VarTotal">

<TAG: GetTotal applename="CS HH Product Bucket List Applet" fieldname="Quantity"
query="Status|Good" variable="VarGoodTotal">
```

```
<TAG: GetTotal appletname="CS HH Product Bucket List Applet" fieldname="Quantity"
query="Status|Defective" variable="VarDefectiveTotal">
```

```
<TAG: GetField appletname="CS HH Product Bucket List Applet" fieldname="Quantity"
variable="VarQuantityField">
```

```
<TAG: GetCount appletname="CS HH Product Bucket List Applet" query="Status|Good"
variable="VarGoodCount">
```

```
<TAG: GetCount appletname="CS HH Product Bucket List Applet" query="Status|Defective"
variable="VarDefectiveCount">
```

```
<REM: *****Variable Assignment Examples End*****>
```

```
<REM: Inventory Print Template. This template includes 3 applets. The ENU template>
```

```
<TAG: Format Face="Helvetica" Size="16" Align="Center">
```

[Distributor Name

Street Address

City, Country Postal Code

Phone Number Fax Number]

```
<TAG: Format Face="Helvetica" Size="20" Bold="TRUE" Align="Center">
```

Inventory Report

```
<TAG: Format Face="Helvetica" Size="8" >
```

```
<TAG: Applet Name="CS HH Inv Loc Print Applet" Caption="TRUE" Wrap="TRUE">
```

```
<TAG: Applet Name="CS HH Product Bucket List Applet" Line="FALSE"
Headerdivider="TRUE" Recorddivider="FALSE" Wrap="TRUE">
```

```
<REM: *****Variable Usage Example*****>
```

```
<TAG: Divider Weight="3" startat="20" endat="80">
```

```
<TAG: Cell startat="0" endat="50" name="TOTAL: " Align="RIGHT">
```

```
<TAG: Cell startat="50" endat="100" variable="VarTotal " Align="RIGHT">
```

```
<TAG: EndOfLine>
```

```
<TAG: Cell startat="0" endat="50" name="TOTAL Defective: " Align="RIGHT">
```

```
<TAG: Cell startat="50" endat="100" variable="VarDefectiveTotal " Align="RIGHT">
```

```
<TAG: EndOfLine>
```

```
<TAG: Cell startat="0" endat="50" name="TOTAL Good: " Align="RIGHT">
```

```
<TAG: Cell startat="50" endat="100" variable="VarGoodTotal " Align="RIGHT">
```

```
<TAG: EndOfLine>
```

```

<TAG: Cell startat="0" endat="50" name="Defective Count: " Align="RIGHT">
<TAG: Cell startat="50" endat="100" variable="VarDefectiveCount" Align="RIGHT">
<TAG: EndOfLine>
<TAG: Cell startat="0" endat="50" name="Good Count: " Align="RIGHT">
<TAG: Cell startat="50" endat="100" variable="VarGoodCount" Align="RIGHT">
<TAG: EndOfLine>
<TAG: Cell startat="50" endat="100" variable="VarReg">
<TAG: EndOfLine>
<TAG: Cell startat="50" endat="100" variable="VarStr">
<TAG: EndOfLine>
<TAG: Cell startat="0" endat="30" variable="VarDate">
<TAG: Cell startat="30" endat="70" variable="VarLongDate">
<TAG: Cell startat="70" endat="100" variable="VarTime">
<TAG: EndOfLine>
<REM: *****Variable Usage Example End*****>
<TAG: Divider Weight="2">
<TAG: Format Face="Helvetica" Size="12" Align="LEFT">
Warehouse Signature
<TAG: Divider Weight="2">
<TAG: Format Face="Helvetica" Size="12" Align="RIGHT">
Rep Signature

```


F

Print Configuration Settings

This appendix describes the print configuration settings that can be set up for your Siebel Handheld application, and includes the following topic:

- [“Overview of Print Configuration Settings for Siebel Handheld Applications” on page 373](#)
- [“Print Configuration Settings” on page 373](#)

Overview of Print Configuration Settings for Siebel Handheld Applications

The administrator sets up the default printer settings in the setup.ini file located in the setup directory. There are a number of parameters to specify printing on Siebel Handheld applications. The parameters specify the following:

- Printer attributes (printer model, port, baud rate, and printer handshake).
- Printing attributes (portrait or landscape orientation, print density and quality, use of compression techniques, and advancing continuous-feed paper).
- Dimensions of the paper (U.S. or European standard paper sizes or custom-size paper).
- Margin settings of the pages (left, right, top, and bottom margins).

These parameters are specified in the [Printing] section of the setup.ini file located in the setup directory. The file contains default settings for all the printing parameters. Review the settings in the file to verify that these settings will work for your printing environment. If necessary, edit the settings as required. You can use any text editor to edit the file.

Printing parameters are listed below in alphabetical order. Parameters are set up using the following syntax:

```
PARAMETER_NAME = VALUE
```

For example:

```
PrinterType = 1
```

Print Configuration Settings

The print configuration settings that can be set up for your Siebel Handheld application are:

- [“MarginBottom” on page 374](#)
- [“MarginLeft” on page 374](#)
- [“MarginRight” on page 374](#)

- ["MarginTop" on page 375](#)
- ["PaperHeight" on page 375](#)
- ["PaperSize" on page 375](#)
- ["PaperWidth" on page 375](#)
- ["PrinterBaudrate" on page 376](#)
- ["PrinterCompressed" on page 376](#)
- ["PrinterDensity" on page 376](#)
- ["PrinterDither" on page 377](#)
- ["PrinterDraftMode" on page 377](#)
- ["PrinterFormFeed" on page 377](#)
- ["PrinterHandshake" on page 378](#)
- ["PrinterOrientation" on page 378](#)
- ["PrinterPort" on page 379](#)
- ["PrinterType" on page 379](#)
- ["PrinterUserSelect" on page 380](#)
- ["Default Printing Settings" on page 381](#)

MarginBottom

Use the MarginBottom parameter to specify the dimensions of the bottom margin of your page. Specify the margin in millimeters using a whole number; decimal fractions are not valid. The default is 5 millimeters.

The data specified in the Footer tag of the print template is printed in the margin specified by MarginBottom. Therefore, if the print template includes a Footer tag, you must specify a wide enough bottom margin to accommodate the footer. MarginBottom must be set to a minimum of 20 millimeters. For more information about MarginBottom, see ["Footer Tag" on page 346](#).

MarginLeft

Use the MarginLeft parameter to specify the width of the left margin of your page. Specify the margin in millimeters using a whole number; decimal fractions are not valid. The default is 5 millimeters.

MarginRight

Use the MarginRight parameter to specify the width of the right margin of your page. Specify the margin in millimeters using a whole number; decimal fractions are not valid. The default is 5 millimeters.

MarginTop

Use the MarginTop parameter to specify the dimensions of the top margin of your page. Specify the margin in millimeters using a whole number; decimal fractions are not valid. The default is 5 millimeters.

The data specified in the Header tag of the print template is printed in the margin specified by MarginTop. Therefore, if the print template includes a Header tag, you must specify a wide enough top margin to accommodate the footer. MarginTop must be set to a minimum of 20 millimeters. For more information about MarginTop, see [“Header Tag” on page 348](#).

PaperHeight

If the paper that your printer uses is not one of the supported standard sizes, then set PaperSize to 5 (custom paper dimensions). Then, specify the height of the paper, in millimeters, using the PaperHeight parameter. You must use a whole number; decimal fractions are not valid. The default is 280 millimeters. If you specify a value of 1–4 for PaperSize, the PaperHeight parameter is ignored.

Use the PaperWidth parameter to specify the width of the paper.

PaperSize

Specify one of the standard paper sizes, or specify a custom paper size. [Table 38](#) describes the valid paper dimension values that you can specify for PaperSize. If you specify a custom paper size (value 5 in [Table 38](#)), you must also specify the PaperWidth and PaperHeight parameters.

Table 38. Paper Dimensions

| Value | Description |
|-------------|----------------------------|
| 1 | Letter size (8.5" x 11.5") |
| 2 | A4 (210 mm x 297 mm) |
| 3 | B5 (176 mm x 250 mm) |
| 4 | Legal size (8.5" x 14") |
| 5 (default) | Custom paper dimensions |

PaperWidth

If the paper that your printer is printing to is not one of the supported standard sizes, then set PaperSize to 5 (custom paper dimensions). Then, you must specify the width measurement of the paper, in millimeters, using the PaperWidth parameter. You must use a whole number; decimal fractions are not valid. The default is 5 millimeters. If you specify a value of 1–4 for PaperSize, the PaperWidth parameter is ignored.

Use the PaperHeight parameter to specify the length of the paper.

PrinterBaudrate

Use PrinterBaudrate to specify the speed of the data transmission for the printer. [Table 39](#) describes the valid values you can specify for PrinterBaudRate. The PrinterBaudrate is applicable only when the PrinterPort is set to serial port.

Table 39. Printer Baud Rate

| Value | Description |
|-------------|---|
| 0 | Always use 0 for LPT or printer ports, or for COM ports that communicate with the printer at 4800 baud. |
| 1 (default) | 9600 baud |
| 2 | 19200 baud |
| 3 | 38400 baud |
| 4 | 57600 baud |
| 5 | 115200 baud |

PrinterCompressed

Use PrinterCompressed to specify the data compression mode. [Table 40](#) describes the valid values you can specify for PrinterCompressed. For the printers supported in this release, always set PrinterCompressed to 1.

Table 40. Data Compression

| Value | Description |
|-------------|--------------------------------|
| 0 | No compression techniques used |
| 1 (default) | Use any known compression |

PrinterDensity

Use the PrinterDensity parameter to get a lighter or darker output from the printer. Start with the default setting, 2, which corresponds to the printer manufacturer's default. As you get feedback from the field, you might need to adjust this setting.

The valid values are 0–4 where 0 is the lightest print density, 4 is the heaviest print density, and 2 is average print density.

PrinterDither

Use to specify the method by which different colors are represented. [Table 41](#) describes the valid values you can specify for PrinterDither. For the printers supported in this release, PrinterDither must always be set to 0.

Table 41. Print Dithering

| Value | Description |
|-------------|---------------------|
| 0 (default) | Use color diffusion |
| 1 | Use dithering |

PrinterDraftMode

Use PrinterDraftMode to specify the quality of the printing. [Table 42](#) describes the valid values you can specify for PrinterDraftMode. For the printers supported in this release, PrinterDraftMode must always be set to 0.

Table 42. Print Quality

| Value | Description |
|-------------|---|
| 0 (default) | Use the highest quality printing |
| 1 | Use a lower quality printing if available |

PrinterFormFeed

Use PrinterFormFeed to specify how paper advances to the next page. [Table 43](#) describes the valid values you can specify for PrinterFormFeed.

If using sheets of paper, set PrinterFormFeed to 0.

Specify 1 or 2 for continuous-feed paper, as follows:

- Set PrinterFormFeed to 1 if you do not need to conserve paper or if you want pages of uniform length. The printer scrolls to the length specified by the PageHeight parameter and trims the page there.
- Set PrinterFormFeed to 2 to conserve paper or if it does not matter if your pages are of varying length. In this case, the printer scrolls 25 millimeters beyond the last printed line and trims the page there. The scroll distance, 25 mm, is not customizable.

For both values, 1 and 2, the page length must never exceed PageHeight.

Table 43. Printer Form Feed

| Value | Description |
|-------------|---|
| 0 (default) | Printer sends a form feed command to the printer at the end of each page. Use this setting for sheet paper. |
| 1 | Printer scrolls the page to the length specified by the PaperHeight parameter. |
| 2 | Printer scrolls the page 25 millimeters past the last printed line of the page. |

PrinterHandshake

Use PrinterHandshake to define the serial port handshake. [Table 44](#) describes the valid values you can specify for PrinterHandshake. The PrinterHandshake parameter applies only when the printer port is a Com1, Com2, or any COM port. If PrinterPort is set to Com1 or Com2, then set PrinterHandshake to 1.

Table 44. Printer Handshake

| Value | Description |
|-------------|-------------------------------|
| 0 | Uses Xon or Xoff |
| 1 (default) | Uses hardware control lines |
| 2 | No handshaking of serial port |

PrinterOrientation

Specify whether the paper is to be oriented vertically (portrait) or horizontally (landscape). [Table 45](#) describes the valid values you can specify for PrinterOrientation. Not all printers can print in landscape orientation. Therefore, you will need to check the specifications for your printer.

Table 45. Paper Orientation

| Value | Description |
|-------------|------------------------|
| 1 (default) | Portrait (vertical) |
| 2 | Landscape (horizontal) |

PrinterPort

Specify the port to which the printer is connected with the PrinterPort parameter. [Table 46](#) describes the valid values you can specify for PrinterPort. If PrinterPort is an LPT port or a network path (value 2 or 9 in [Table 46](#)), then the PrinterBaudrate must be set to 0.

NOTE: The communication port on the printer must be compatible with the host PDA. For IrDA, the serial communication setting must be set correctly to allow reliable printing.

Table 46. Printer Port

| Value | Description |
|-------------|---|
| 0 | Com1: serial port |
| 1 | Com2: serial port |
| 2 | LPT: parallel port |
| 3 (default) | IR Infrared port |
| 4 | Com3: serial port |
| 5 | Com4: serial port |
| 6 | Com5: serial port |
| 7 | Com6: serial port |
| 8 | Print to OUTPUT.PRN |
| 9 | Print to network address |
| 10 | Bluetooth BlueCard solution by Wireless Solutions |
| 11 | Socket PNC Bluetooth card support |
| 12 | Bluetooth BlueCard solution by Anycom |
| 13 | Com7: serial port |
| 14 | Com8: serial port |

PrinterType

Specify the type of printer using the PrinterType parameter. [Table 47](#) lists the printers supported by the Printer CE print driver used by Siebel applications.

Table 47. Supported Printers

| Value | Description |
|-------|-------------------------------------|
| 0 | Canon BubbleJet compatible printers |
| 1 | Citizen PD-04 |

Table 47. Supported Printers

| Value | Description |
|-------|--|
| 2 | Citizen PN60i |
| 3 | Epson ESC/P 2 compatible printers |
| 4 | Epson Stylus compatible printers |
| 5 | HP PCL compatible - includes DeskJet and LaserJet |
| 6 | Pentax PocketJet 200 |
| 7 | Pentax PocketJet II |
| 8 | Seiko DPU-3445 |
| 9 | Generic Epson compatible 180DPI 24-pin printers (LQ/ESC) |
| 10 | Generic Epson compatible 360DPI 24-pin printers (LQ/ESC) |
| 11 | Generic Epson compatible 203DPI 24-pin printers (LQ/ESC) |
| 12 | Extech 2" |
| 13 | Extech 3" |
| 14 | O'Neil MicroFlash |
| 15 | DymoCoStar label printers |
| 16 | Seiko LabelWriter printers |
| 17 | Extech 4" |
| 18 | SiPix A6 printer |
| 20 | Zebra RW 220 |
| 102 | Generic 40 column Dot Matrix Printer |
| 108 | Generic 80 column Dot Matrix Printer |

PrinterUserSelect

Use PrinterUserSelect to enable a printer dialog box for Siebel Handheld applications, whereby users are prompted via a dialog box to specify the settings for their printer each time they initiate a print job. For more information about the printer settings dialog box, see ["Configuring a Printer Settings Dialog Box for the Siebel Handheld Application"](#) on page 86.

Table 48 describes the valid values you can specify for PrinterUserSelect.

Table 48. Printer User Select

| Value | Description |
|-------|--|
| 1 | Displays a printer dialog box for each copy of each form if multi-form, multi-copy printing is used. |
| 2 | Displays a printer dialog box only once when the first form starts printing. The same settings are used for all copies if multi-form, multi-copy printing is used. |

Default Printing Settings

The following are the printing parameters from the setup.ini file. The values shown are the default settings.

```
[Pri nti ng]
Pri nterType= 14
Pri nterPort= 3
Pri nterBaudrate= 1
Pri nterHandshake= 1
Pri nterOri entati on= 1
Pri nterDensi ty= 2
Pri nterCompressed= 1
Pri nterDi ther= 0
Pri nterDraftMode= 0
Pri nterFormFeed= 0
PaperSi ze= 5
PaperWi dth= 115
PaperHei ght= 280
Margi nLeft= 5
Margi nRi ght= 5
Margi nTop= 5
Margi nBottom= 5
```




COM Extensibility

This appendix covers the extension of Component Object Model (COM) support for Siebel Handheld applications. The audience for this appendix includes Siebel application developers and Siebel integration developers. The following topics are covered in this appendix:

- [About COM Extensibility on page 383](#)
- [Process of Implementing a SCEC on page 390](#)

About COM Extensibility

COM support for Siebel Handheld applications allows customers to programmatically extend the functionality of the Handheld application. By developing a COM object according to a predefined Siebel specification, users can customize responses to Siebel events. Action request methods allow a customized COM object to interact with the Siebel Handheld application.

Overview of the SCEC Framework

This topic describes the major components of the Siebel Handheld CE (SHCE) COM Extension Framework.

An SHCE COM Extension Component (SCEC) is part of the Siebel Handheld application architecture. One or more SCECs can subscribe to event notifications on Siebel objects, including applets, business components, business services, and the application. A SCEC can perform action requests on the Siebel Handheld application to manipulate Siebel objects. SCECs implement an interface derived from `ISHCEEEvent`, which is defined by the COM Extension Framework.

Siebel events are defined as actionable requests triggered by application logic. Externally defined program modules, implemented by the customer, will handle these events. This feature allows customer-defined business logic to extend the application to handle third-party logic and validation. For example, `Buscomp::PreWriteRecord` will be called before any record is written to the Siebel Server database, thus allowing you to perform early, specialized validation on a particular field.

Siebel action requests are defined as APIs, implemented by the Siebel Handheld architecture, which allow external customer modules to perform standard operations on the Siebel Handheld application. All action request interfaces are derived from the base `ISHCEActionReq` interface. For example, `BusComp::SetFieldValue` can be called by a SCEC in order to set a specific value for a Business Component.

Both Siebel objects and SCECs exist within the Siebel process boundary in each layer of the Siebel architecture. For example, COM components can extend the applet functionality in the user interface layer, while on the object manager layer, extension objects can implement specialized logic by interacting with business components, business objects, and services.

The Siebel object and the SCEC maintain references to each other and interact through synchronous method calls. The Siebel object blocks while the SCEC implements custom behavior for the event to which it is subscribed. After the SCEC handles an event, it returns control to the Siebel object. The COM component uses a returned value to specify whether the Siebel object can continue processing the event normally, cancel operations, or return an error.

Receiving Event Notifications

To intercept Siebel object events, a customer must create a SCEC that implements one of the exported SHCE COM Event Interfaces (SCEI) derived from ISHCEEEvent. Although both the Siebel object and the SCEC are within the Siebel Handheld application process boundary, the SCEC is the COM server, and the Siebel Handheld application is the COM client consumer. All SCECs are free-threaded, in-process (.dll) COM servers.

A SCEC implements one of the event interfaces derived from the generic event interface (ISHCEEEvent). The event interfaces are listed in [Table 49](#).

Table 49. SCEI Interfaces

| SHCE COM Event Interface (SCEI) | Description |
|---------------------------------|---|
| ISHCEEEvent | Contains generic interfaces common to all SCEIs. All event interfaces are derived from ISHCEEEvent. |
| ISHCEAppletEvent | Used for applet events. |
| ISHCEApplicationEvent | Used for application and global events. |
| ISHCEBusCompEvent | Used for business component events. |
| ISHCEServiceEvent | Used for service events. |

For example, to receive notifications that a business component (such as Account) is committing a record, a SCEC implements the event interface corresponding to the business component events (ISHCEBusCompEvent interface), and adds the specialized handling code in the method (PreWriteRecord).

User properties of the Siebel object in Siebel Tools indicate the events for which a SCEC will be notified. For more information about Tools configuration, see [Configuring SCEC User Properties in Siebel Tools on page 402](#). When a Siebel event, to which a SCEC is subscribed, occurs, the Siebel object will instantiate the SCEC and notify it of the event by invoking one of the SCEI interface methods.

A SCEC can manipulate the program flow by returning one of the HRESULT codes listed in [Table 50](#).

Table 50. SCEI Return Codes

| SHCE COM Event Interface (SCEI) Return Codes | Description |
|--|---|
| EL_EVENT_EMPTY | No event procedure was found. Log a warning and perform normal processing. |
| EL_EVENT_CONTINUE | The event procedure ran successfully. The Siebel application should continue to perform normal processing. |
| EL_EVENT_COMPLETE | The event procedure performed the operation and ran successfully. The Siebel application should return without performing any additional processing (without performing normal processing). |
| EL_EVENT_ERROR | The event procedure generated an error. Return with error. |
| EL_EVENT_CANCEL | The event procedure was cancelled. Return without error. |

COM extensibility supports the following Siebel events:

| | | |
|-----------------------------|--------------------------|----------------------------|
| Applet_GotFocus | Application_PreSync | BusComp_PreNewRecord |
| Applet_InvokeMethod | Application_Start | BusComp_PreQuery |
| Applet_Load | BusComp_Associate | BusComp_PreSetFieldValue |
| Applet_PreInvokeMethod | BusComp_ChangeRecord | BusComp_PreWriteRecord |
| Applet_TaskButtonPressed | BusComp_DeleteRecord | BusComp_Query |
| Application_Close | BusComp_InvokeMethod | BusComp_SetFieldValue |
| Application_InvokeMethod | BusComp_NewRecord | BusComp_WriteRecord |
| Application_Navigate | BusComp_PreAssociate | Service_InvokeMethod |
| Application_PreClose | BusComp_PreDeleteRecord | Service_PreCanInvokeMethod |
| Application_PreInvokeMethod | BusComp_PreGetFieldValue | Service_PreInvokeMethod |
| Application_PreNavigate | BusComp_PreInvokeMethod | |

Invoking Siebel Methods (Action Requests)

A SCEC uses action request interface references to interact with Siebel objects. Immediately after creating a SCEC, Oracle's Siebel Handheld applications provides the action request interface pointers of the root Siebel objects to the SCEC and invokes the appropriate SCEC event method. A SCEC can use the root interface references to access and manipulate all other Siebel objects. When a SCEC intercepts a Siebel event, it can customize business logic by invoking action requests through the SHCE COM Action Request Interfaces (SCARI). Action request interface references to other objects are available from the root interfaces listed in [Table 51](#).

Table 51. SCARI Interfaces

| SHCE COM Action Request Interface (SCARI) | Description |
|---|---|
| ISHCEActionReq | Generic interface from which all other callback interfaces are derived. |
| ISHCEApplet | Used for applet methods. |
| ISHCEApplication | Used for application and global methods. |
| ISHCEBusComp | Used for business component methods. |
| ISHCEBusObject | Used for business object methods. |
| ISHCEPropertySet | Property sets used to pass input arguments to other object methods. |
| ISHCEService | Used for service events. |

COM Extensibility supports the Siebel Action Requests listed in [Table 52](#).

Table 52. SCARI Interfaces and Action Requests

| SHCE COM Action Request Interface (SCARI) | Action Request |
|---|------------------|
| Applet Action Requests: ISHCEApplet | BusComp |
| | BusObject |
| | DrillDown |
| | GetChildProperty |
| | InvokeMethod |
| | Name |
| | SetChildProperty |

Table 52. SCARI Interfaces and Action Requests

| SHCE COM Action Request Interface (SCARI) | Action Request |
|--|--------------------|
| Application Action Requests: ISHCEApplication | ActiveApplet |
| | ActiveBusComp |
| | ActiveBusObject |
| | GetBusObject |
| | GetLastErrCode |
| | GetLastErrText |
| | GetLocalString |
| | GetProperty |
| | GetService |
| | GetUserProperty |
| | GotoView |
| | InvokeMethod |
| | NewPropertySet |
| | PlaySound |
| | SetActiveApplet |
| | SetErrorCode |
| | SiebelMessageBox |
| Trace | |
| Business Component Action Requests: ISHCEBusComp | Associate |
| | BusObject |
| | CheckActiveRow |
| | ClearToQuery |
| | DeleteRecord |
| | EnableFullFetch |
| | ExecuteQuery |
| | FirstRecord |
| | GetAssocBusComp |
| | GetFieldValue |
| | GetNamedSearch |
| | GetPicklistBusComp |

Table 52. SCARI Interfaces and Action Requests

| SHCE COM Action Request Interface (SCARI) | Action Request |
|--|----------------|
| Business Component Action Requests: ISHCEBusComp (continued) | GetSearchExpr |
| | GetSearchSpec |
| | GetViewMode |
| | InvokeMethod |
| | IsFullFetch |
| | LastRecord |
| | Name |
| | NewRecord |
| | NextRecord |
| | ParentBusComp |
| | Pick |
| | PreviousRecord |
| | SetFieldValue |
| | SetNamedSearch |
| | SetSearchExpr |
| | SetSearchSpec |
| | SetSortSpec |
| | SetViewMode |
| UndoRecord | |
| WriteRecord | |
| Business Object Action Requests: ISHCEBusObject | GetBusComp |
| | Name |

Table 52. SCARI Interfaces and Action Requests

| SHCE COM Action Request Interface (SCARI) | Action Request |
|---|----------------|
| Property Set Action Requests: ISHCEPropertySet | AddChild |
| | Copy |
| | GetChild |
| | GetCount |
| | EnumProperty |
| | GetProperty |
| | GetType |
| | GetValue |
| | InsertChildAt |
| | RemoveChild |
| | Reset |
| | SetProperty |
| | SetValue |
| Business Service Action Requests: ISHCEService | GetPropertySet |
| | InvokeMethod |
| | Name |

Obtaining and Using Action Request Interface References

The action request interface corresponding to the application object (ISHCEApplication) allows access to all other ISHCEActionReq derived interfaces. A pointer to this interface is stored in `m_pApp`, which is a member of the `CSiebelSCECBase` class. For example, `ISHCEApplication::ActiveApplet` returns a reference to the `ISHCEApplet` interface of the active applet. Similarly, a reference to the Contact business component can be obtained by invoking `ISHCEBusObject::GetBusComp` on the business object interface reference returned by `ISHCEApplication::GetBusObject`.

A SCEC is also passed a reference to the corresponding action request object of the Siebel object that created it. For example, a SCEC implementing `ISHCEAppletEvent` will be able to access the `ISHCEApplet` interface reference of the current applet. A pointer to this interface is stored in `m_pRootObj`, which is a member of the `CSiebelSCECBase` class.

A SCEC can use this interface to obtain references to other action request interfaces. For example, if created from a service, a SCEC could access the service's property set using the `ISHCEService::GetPropertySet` method. Once the SCEC handles an event and returns control to the Siebel object that created it, the action request interfaces are invalidated and can no longer be used by the SCEC. Only action requests invoked on the same thread as event notification (the SCEC method invoked by the Siebel object) are supported. When the SCEC has finished handling the event, it releases all action request interface pointers.

Process of Implementing a SCEC

To create a SHCE COM Extension Component (SCEC), perform the following tasks:

- [Obtaining the SCEC Support Files on page 390](#)
- [Generating and Configuring a SCEC on page 391](#)
- [Modifying SCEC Files on page 392](#)
- [Deploying SCEC Files to a PDA Device on page 401](#)
- [Configuring SCEC User Properties in Siebel Tools on page 402](#)
- [Configuring SCEC Registry Settings on page 403](#)

Obtaining the SCEC Support Files

This task is a step in [Process of Implementing a SCEC on page 390](#).

Contact your Oracle Expert Services or Account Support representative to obtain the following SCEC support files:

| | | | |
|---------------------------------|---------------------------------|-----------------------------|------------------------------------|
| <code>actionrequest.h</code> | <code>application_ar_i.c</code> | <code>busobj_ar.h</code> | <code>service_ar.idl</code> |
| <code>actionrequest.idl</code> | <code>application_ar_p.c</code> | <code>busobj_ar.idl</code> | <code>service_ar_i.c</code> |
| <code>actionrequest_i.c</code> | <code>application_ev.h</code> | <code>busobj_ar_i.c</code> | <code>service_ar_p.c</code> |
| <code>actionrequest_p.c</code> | <code>application_ev.idl</code> | <code>busobj_ar_p.c</code> | <code>service_ev.h</code> |
| <code>applet_ar.h</code> | <code>application_ev_i.c</code> | <code>dlldata.c</code> | <code>service_ev.idl</code> |
| <code>applet_ar.idl</code> | <code>application_ev_p.c</code> | <code>event.h</code> | <code>service_ev_i.c</code> |
| <code>applet_ar_i.c</code> | <code>buscomp_ar.h</code> | <code>event.idl</code> | <code>service_ev_p.c</code> |
| <code>applet_ar_p.c</code> | <code>buscomp_ar.idl</code> | <code>event_i.c</code> | <code>shcecom.h</code> |
| <code>applet_ev.h</code> | <code>buscomp_ar_i.c</code> | <code>event_p.c</code> | <code>siebeliscecmacro.h</code> |
| <code>applet_ev.idl</code> | <code>buscomp_ar_p.c</code> | <code>propset_ar.h</code> | <code>t_siebeliscec.h</code> |
| <code>applet_ev_i.c</code> | <code>buscomp_ev.h</code> | <code>propset_ar.idl</code> | <code>t_siebeliscecapp.h</code> |
| <code>applet_ev_p.c</code> | <code>buscomp_ev.idl</code> | <code>propset_ar_i.c</code> | <code>t_siebeliscecapplet.h</code> |
| <code>application_ar.h</code> | <code>buscomp_ev_i.c</code> | <code>propset_ar_p.c</code> | <code>t_siebeliscecbc.h</code> |
| <code>application_ar.idl</code> | <code>buscomp_ev_p.c</code> | <code>service_ar.h</code> | <code>t_siebeliscecsrv.h</code> |

Generating and Configuring a SCEC

This task is a step in [Process of Implementing a SCEC on page 390](#).

To generate and modify a SCEC using the ATL Wizard in Visual Studio 2005

- 1 Create a new project in Visual Studio 2005 with the following settings:
 - Project type = Visual C++ > Smart Device
 - Template = ATL Smart Device Project
 - Name = Name of the SCEC (for example, SampleSCEC)
- 2 In the ATL Smart Device Project Wizard, click Next.
- 3 Click the left arrow button to remove the Pocket PC 2003 SDK from the list of Selected SDKs.
- 4 Click the right arrow button to add the Windows Mobile 5.0 Pocket PC SDK to the list of Selected SDKs. Click Next.
- 5 In the Application Settings screen, choose Dynamic-link library for the Server type. Do not check any other options. Click Finish.

This step generates SCEC files, such as SampleSCEC.cpp and SampleSCEC.idl.

- 6 Switch from the Solution Explorer to the Class view, right-click the SampleSCEC project, and select Add, then Class.
- 7 From the Visual C++ Category, select Smart Device. From Visual Studio installed templates, select ATL Simple Object. Click Add.
- 8 In the Welcome to the ATL Simple Object Wizard screen, enter a name for the SCEC in the Short Name field (for example, SampleSCEC). This will automatically populate the other fields.

NOTE: If the project name is the same as the Short Name (for example, SampleSCEC), you must slightly modify the names of the .h and .cpp files, because by default they will be SampleSCEC.h and SampleSCEC.cpp, respectively. The ATL wizard will generate an error if you leave these names unchanged and click Next, because SampleSCEC.h is generated by MIDL and therefore cannot be used, and SampleSCEC.cpp has already been generated by the ATL wizard. For example, you can modify the names to be SampleSCECObj.h and SampleSCECObj.cpp.

Click Next.

- 9 In the Options screen, select the following options:
 - Threading Model = Free
 - Interface = Custom

Click Finish.

- 10 Navigate to the Solution Explorer, right-click the generated .idl file (for example, SampleSCEC.idl), and select Properties.
- 11 Navigate to Configuration Properties, then General and change the value of the Tool property from MIDL Tool to Custom Build Tool. Click Apply.

- 12** Navigate to Configuration Properties, Custom Build Step, then General. In the Command Line text box, add the location of the Siebel (COMMIDL) .idl files in the following format:

```
mi dl /nologo /Oicf /I <action requests location> /I <events location> /h
"<Proj Name>.h" /iid "<Proj Name>_i.c" "<Proj Name>.idl"
```

For example:

```
mi dl /nologo /Oicf /I ..\..\com\interfaces\actionrequests /I
..\..\com\interfaces\events /h "SampleSCEC.h" /iid "SampleSCEC_i.c"
"SampleSCEC.idl"
```

- 13** Select Outputs and add the following files to the Outputs text box:

- .\<>ProjName>.tlb
- .\<>ProjName>.h
- .\<>ProjName>_i.c

For example:

- .\SampleSCEC.tlb
- .\SampleSCEC.h
- .\SampleSCEC_i.c

Click OK.

- 14** Right-click the Project name and select Properties.

- 15** Navigate to Configuration Properties, C/C++, then General and modify the settings to include the generated Siebel COMMIDL idl files, t_siebelsec.h, and shcescec.h by specifying the appropriate directories in the Additional Include Directories text box.

For example:

```
..\..\com\comimpl;..\..\com\scec;..\..\com\interfaces\mi dl
```

Click OK.

Modifying SCEC Files

This task is a step in [Process of Implementing a SCEC on page 390](#).

To modify SCEC files

- 1** Open and modify the .cpp file generated by the ATL wizard (for example, SampleSCEC.cpp) by including one or more appropriate _i.c files.

For example:

```
#include "application_ev_i.c"
#include "buscomp_ev_i.c"
```



```
#include "service_ev_i.c"
#include "applet_ev_i.c"
```

- 2 Open the stdafx.cpp file, and add the following code above `#include "stdafx.h"`:

```
#define _CE_ALLOW_SINGLE_THREADED_OBJECTS_IN_MTA
```

- 3 Modify the .idl files (for example, SampleSCEC.idl) as follows:

- a Remove the definition of the interface generated by the ATL wizard. For example, remove the following code:

```
import "oaidl.idl";
import "ocidl.idl";
import "atlinterface.idl";
```

```
[
object,
uuid (D26E065A-3365-4A41-ACDE-486E98C204A8),
helpstring ("ISampleSCEC Interface"),
pointer_default (unique);
]
Interface ISSampleSCEC : IUnknown{
};
```

- b Import the appropriate Siebel .idl files for the event to be implemented.

For example, if you are implementing a business component SCEC, add the following:

```
import "application_ar.idl";
import "application_ev.idl";
import "buscomp_ev.idl";
```

- c Replace the automatically generated interface name (for example, ISampleSCEC) with the appropriate ISHCEXXXEvent interface. For example, if you are implementing a business component SCEC, change the interface name to ISHCEBusCompEvent. The final lines of code resemble the following:

```
class SampleSCEC
{
    [default] interface ISHCEBusCompEvent;
};
```

```
};
```

4 Modify the SampleSCECObj.h file as follows:

- a Add statements to include SampleSCEC.h and any Siebel templates for the implemented objects. For example:

```
#i ncl ude "Sampl eSCEC. h"
#i ncl ude "t_si ebel scecbc. h"
```

- b Add any events that you intend to implement to the class definition. Use the appropriate template according to the event. Possible events and templates are listed in the following table:

| Event | Template |
|---------------------------|------------------------|
| Application events | CSiebelSCECAppBase |
| Business Component events | CSiebelSCECBusCompBase |
| Applet events | CSiebelSCECAppletBase |
| Service events | CSiebelSCECServiceBase |

For example, to replace the ATL-generated class definition for CSampleSCEC, replace the following code:

```
cl ass ATL_NO_VTABLE CSampl eSCEC :
    publ i c CComObj ectRootEx<CComMul ti ThreadModel >,
    publ i c CComCoCl ass<CSampl eSCEC, &CLSI D_Sampl eSCEC>,
    publ i c ISampl eSCEC
```

with this definition:

```
cl ass ATL_NO_VTABLE CSampl eSCEC :
    publ i c CSi ebel SCECBusCompBase
        <CSampl eSCEC, &CLSI D_Sampl eSCEC,
        &LI BI D_Sampl eSCECLI b, IDR_SAMPLESCEC>
```

where:

- The first parameter (CSampleSCEC) indicates the class name.
 - The second and third parameters hold class and library Ids, respectively, and are defined in the <ProjName>_i.c file.
 - The fourth parameter (IDR_xxx) is defined in the resourceppc.h file.
- c Replace the ATL-generated interface name (for example, ISampleSCEC) with the appropriate ISHCEXXXEvent interface. For example:

```

BEGIN_COM_MAP(CSampleSCEC)
    COM_INTERFACE_ENTRY(ISHCEBusCompEvent)
END_COM_MAP()

```

- d** Remove the following:

```

DECLARE_PROTECT_FINAL_CONSTRUCT()

HRESULT FinalConstruct()
{
    return S_OK;
}

void FinalRelease()
{
}

```

- e** Add the appropriate STDMETHODS that you want to implement. For example, to add `InvokeMethod`, add the following code:

```
STDMETHOD (InvokeMethod) (BSTR pMethod, BSTR* args, int nArgs);
```

- 5** Implement the SCEC in the `.cpp` file (for example, `SampleSCECObj.cpp`). To implement `CSampleSCEC::InvokeMethod`, add the following code:

```

STDMETHODIMP CSampleSCEC::InvokeMethod(BSTR pMethod, BSTR* args, int nArgs)
{
    // Your implementation here
}

```

The following extended example first implements a helper method (`URLEncode`), then queries the selected account for an address, and displays it on a map.

```

#include "stdafx.h"
#include "SampleSCECObj.h"

// CSampleSCEC
// METHOD NAME
// URLEncode
// DESCRIPTION

```

```

// This function is used to encode a URL.
CComBSTR URLEncode (LPCTSTR pStr)
{
    CComBSTR pURL = _T("");

    if (!pStr)
    {
        return pURL;
    }

    TCHAR pBuffer[4];

    memset (pBuffer, 0x00, sizeof(TCHAR) * 4);

    for ( ; *pStr != _T('\0'); pStr++)
    {
        switch (*pStr)
        {
            case 0x0009 : case 0x000A : case 0x000D :
            {
                // put a 0, since these are single digit chars
                _stprintf (pBuffer, _T("%0%x"), *pStr);
                pURL.Append (pBuffer);
                break;
            }

            case 0x0020 :
            {
                // encode space as a plus sign
                pURL.Append (_T("+"));
            }
        }
    }
}

```

```

        break;
    }

    case 0x0021 : case 0x0022 : case 0x0023 : case 0x0024 :
    case 0x0025 : case 0x0026 : case 0x0027 : case 0x0028 :
    case 0x0029 : case 0x002B : case 0x002C : case 0x002F :
    case 0x003A : case 0x003B : case 0x003C : case 0x003D :
    case 0x003E : case 0x003F : case 0x0040 : case 0x005B :
    case 0x005C : case 0x005D : case 0x005E : case 0x0060 :
    case 0x007B : case 0x007C : case 0x007D : case 0x007E :
    {
        // encode the special characters
        _stprintf (pBuffer, _T("%%x"), *pStr);
        pURL.Append (pBuffer);
        break;
    }

    default:
    {
        // no encoding is needed
        TCHAR pChar[1];
        _stprintf (pChar, _T("%c"), *pStr);
        pURL.Append (pChar);
        break;
    }
}

return pURL;
}

```

```

STDMETHODIMP CSampleSCEC::InvokeMethod(BSTR pMethod, BSTR* args, int nArgs)
{
    CComPtr<ISHCEBusComp> pBusComp;
    CComPtr<ISHCEBusObject> pBusObj;

    START_EVENT_ARG (CSampleSCEC::InvokeMethod);

    // Check if the Map Address button was clicked
    if (pMethod && _tcscmp (pMethod, _T("EventMethod")) == 0)
    {
        /// Make sure there is at least 1 account record present
        if (m_pRootObj->CheckActiveRow () == S_OK)
        {
            // get the primary address id
            CComBSTR bstrPrimaryAddressId;
            CALL_INT (m_pRootObj, GetFieldValue (FALSE, CComBSTR (_T("Primary Address
            Id")), &bstrPrimaryAddressId));

            // now get the appropriate business object and business component
            CALL_INT (m_pApp, GetBusObject (CComBSTR (_T("Account")), &pBusObj));
            CALL_INT (pBusObj, GetBusComp (CComBSTR (_T("CUT Address")), &pBusComp));

            // query for the record:
            BOOL bBool = FALSE;
            CALL_INT (pBusComp, ClearToQuery ());

            CComBSTR bstrSearchSpec = CComBSTR (_T("[Id] = '"));
            bstrSearchSpec += bstrPrimaryAddressId;
            bstrSearchSpec += CComBSTR (_T(""));
        }
    }
}

```

```

CALL_INT (pBusComp, SetSearchExpr (bstrSearchSpec));
CALL_INT (pBusComp, ExecuteQuery ());
if (pBusComp->FirstRecord (&bBool) == S_OK && bBool)
{
    // get the required info about the address

    // get Street Address
    CComBSTR bstrStreetAddress;
    CALL_INT (pBusComp, GetFieldValue (FALSE, CComBSTR (_T("Street Address")),
    &bstrStreetAddress));

    // get Street Address 2
    CComBSTR bstrStreetAddress2;
    CALL_INT (pBusComp, GetFieldValue (FALSE, CComBSTR (_T("Street Address
2")), &bstrStreetAddress2));

    // get City
    CComBSTR bstrCity;
    CALL_INT (pBusComp, GetFieldValue (FALSE, CComBSTR (_T("City")),
    &bstrCity));

    // get State
    CComBSTR bstrState;
    CALL_INT (pBusComp, GetFieldValue (FALSE, CComBSTR (_T("State")),
    &bstrState));

    // get Zip Code
    CComBSTR bstrZipCode;
    CALL_INT (pBusComp, GetFieldValue (FALSE, CComBSTR (_T("Postal Code")),
    &bstrZipCode));

```

```

        // get Country
        CComBSTR bstrCountry;

        CALL_INT (pBusComp, GetFieldValue (FALSE, CComBSTR (_T("Country")),
&bstrCountry));

        // construct and encode the URL
        CComBSTR bstrURL = CComBSTR (_T("www.mapquest.com/pda/
maps.adp?address="));

        if (_tcscmp (bstrStreetAddress2, _T("")) != 0)
        {
            bstrStreetAddress += CComBSTR (_T(" "));
            bstrStreetAddress += bstrStreetAddress2;
        }

        bstrURL += URLEncode (bstrStreetAddress);

        bstrURL += CComBSTR (_T("&city="));
        bstrURL += URLEncode (bstrCity);

        bstrURL += CComBSTR (_T("&state="));
        bstrURL += URLEncode (bstrState);

        bstrURL += CComBSTR (_T("&zipcode="));
        bstrURL += URLEncode (bstrZipCode);

        bstrURL += CComBSTR (_T("&country="));
        bstrURL += URLEncode (bstrCountry);

        // map the address
        PROCESS_INFORMATION inf;

```



```

        CreateProcess (_T("\\Windows\\iexpl ore. exe"), bstrURL, NULL, NULL, FALSE,
0, NULL, NULL, NULL, &inf);
    }
    else
    {
        // Log message that there is no address to map
        CComBSTR bstrInfoMessage =
            CComBSTR (_T("CSampleSCEC::InvokeMethod : The currently selected
account record has no address to map. "));

        CALL_INT (m_pApp, Trace (bstrInfoMessage, 0));
    }
}
}

II_abort:
    FINISH_EVENT;
}

```

Deploying SCEC Files to a PDA Device

This task is a step in [Process of Implementing a SCEC on page 390](#). You can either deploy the SCEC files to the PDA using Visual Studio, or deploy them manually.

To Deploy SCEC files to a PDA Device using Visual Studio

CAUTION: Do not choose the `\Windows` directory as the deployment location on the PDA.

- 1 To specify the deployment location, right-click the project and select Properties.
- 2 Navigate to Configuration Properties, Deployment and specify a Remote Directory, such as `\program files\siebel handheld\scec`.
- 3 Choose Build, then Deploy. This copies the `.dll` to the specified directory on the PDA and automatically registers the `.dll`.

To Manually Deploy SCEC files to a PDA device

CAUTION: Do not choose the `\Windows` directory as the deployment location on the PDA.

- 1 Copy the SCEC .dll to a directory on the PDA.
- 2 Register the .dll by running regsvrce.exe (located in \windows on the PDA). Specify the path of the .dll path, such as \program files\siebel handheld\scec\sampl escec. dll.

Configuring SCEC User Properties in Siebel Tools

This task is a step in [Process of Implementing a SCEC on page 390](#).

Once the SCEC has been created, locate the corresponding Siebel object in Oracle’s Siebel Tools and add a user property to specify the class ID (CLISD), the Siebel method to subscribe to, and any additional information.

Format the user properties as follows:

Property Name: SHCE COM Event <x>

Property Value: "<MethodName>", "<CLISD>", "<extra info>"

NOTE: You must use quotes around the concatenated values.

where:

- <x> indicates number starting with 1. For example, SHCE COM Event 1.
- <MethodName> indicates the name of the method, such as BusComp_PreWriteRecord.
- <CLISD> indicates the Class ID of the SCEC. The Class ID can be found in the SampleSCEC.idl file by searching for “library SampleSCECLib” and locating the uuid string. For example, uuid(85342298-0FD5-4F58-B454-07B1F6671A8C).
- <extra info> indicates optional extra information. Some events allow the specification of additional information, such as the method or field name for which events should be dispatched. The event will be dispatched only if the current method or field name match. If no additional arguments are specified, the event is always dispatched. For example, if you formatted the property value as “BusComp_InvokeMethod”, “12373DB6-4CEA-11D2-A3DC-0008C78A05F9”, “FooBar Method”, the event is dispatched when the FooBar Method is invoked. For all other methods, no event is dispatched.

The following Siebel events support the <extra info> information:

| | | |
|-----------------------------|--------------------------|----------------------------|
| Applet_InvokeMethod | Application_PreNavigate | Service_InvokeMethod |
| Applet_PreInvokeMethod | BusComp_InvokeMethod | Service_PreCanInvokeMethod |
| Applet_TaskButtonPressed | BusComp_PreGetFieldValue | Service_PreInvokeMethod |
| Application_InvokeMethod | BusComp_PreInvokeMethod | |
| Application_Navigate | BusComp_PreSetFieldValue | |
| Application_PreInvokeMethod | BusComp_SetFieldValue | |

For example:

Name: SHCE COM Event 1

Value: "Buscomp_PreWriteRecord", "37073DB6-4CEA-11D2-A3DC-0008C78A05F9"

Configuring SCEC Registry Settings

This task is a step in [Process of Implementing a SCEC on page 390](#).

Configure the following Windows Registry PDA settings for each SCEC:

- **SHCE COM Disable.** This registry setting is not present by default. Add this setting and set it to Y to disable COM Extensibility.
- **SCEC Dir.** This registry setting specifies the location of the directory of the SCEC. All *.dll files in this directory are preloaded. This registry setting is not present by default; you must add this setting.

For example:

`\program files\siebel handheld\scec`

CAUTION: Specify a directory containing only SCEC .dll files. Using `\Windows` causes the application to attempt to preload all .dll files.

Index

A

About BatchSync for Siebel Handheld

Applications 124

Absolute print tag 334

Account Business Component

modifying 60

Account Call view

adding account calls from 255

account calls, about 241

account calls, creating

Account Call view, adding from 255

Accounts view, adding from 253

creating, about 253

account calls, recording

about 255

completed account call, submitting 256

entering attendee information 256

products detailed information, entering for
account call 255

Accounts view

adding account calls from 253

accounts, managing

account addresses, adding 236

account, adding 236

account, described 233, 234, 235

accounts and professionals, indicating
affiliations 238

accounts, specifying best times to call 237

indicating affiliations between contacts and
accounts 231

notes, adding about accounts 238

sales data, viewing 239

active inventory period 263

Activities view

general activities, adding from 257

activities, creating general activities

about 257

general activities, adding from Activities
view 257

general activities, adding using
Calendar 258

administration screens 97

All Contacts button, using 251

applet classes

CSSFrame 301

CSSFrameAcctCall 301

CSSFrameBase 307

CSSFrameCECaIAddModify 307

CSSFrameCEGridDay 301

CSSFrameCEGridMonth 301

CSSFrameCEGridWeek 302

CSSFrameCEHome 302

CSSFrameFindAccount 303

CSSFrameFindContact 302

CSSFrameList 307

CSSFrameListBase 307

CSSFrameListSampleItem 303

CSSFrameLSGridDay 304

CSSFrameMultiplePick 305

CSSFramePopupAccount 305

CSSFramePopupContact 306

CSSFramePopupCurrency 307

CSSFrameProfCall 306

applet classes, list of 301

applets

default behavior, overriding 33

form applets, designing 84

list applets, designing 85

print applets, designing 84

Print buttons, configuring 85

printing, configuring 85

screen allocation 33

togglng between 207

URLs in applet fields 208

application memory, increasing 220

application-level menu 206

architecture, five layers described 26

asterisk (*), and required fields 37

attendees

attendee calls, about creating 257

attendee information, entering for an account
call 256

audit trail

obtaining audit trail for synchronization
sessions 137

Auto Call button

about 241

My Accounts view and Account Calls,
about 253

using, about 244

auto pop-up list, about using 41

Auto Update, about 241

automatic backup, after

synchronization 213

auxiliary configuration file

- setting key values 76

B

- Back button, about** 208

- Background print tag** 341

backing up data

- about 212
- automatic backup after synchronization 213
- editing database parameters 180
- external media 180
- restoring data 213
- restoring from 181
- troubleshooting 275
- user database backups, enabling 180

barcode scanning

- views, enabling for 119

barcode, administering

- barcode scanning, enabling views for 119
- scanning, enabling views for 119

- BarcodeNewRecord user property** 113

barcodes, administering

- about 112
- barcode definition, creating 114, 122
- barcode scanning, enabling 123
- enabling views for barcode scanning 119
- enhancing the behavior of barcode scan 119
- setting new record behavior for scans 113

business component classes

- CSSBCAccountSIS 297
- CSSBCAutoProduct 300
- CSSBCBase 300
- CSSBCHHPharmaCall 297
- CSSBCHHSampleItem 298
- CSSBCUser 300
- CSSBusComp 300

- business component classes, list of** 297

business components

- list of supported components 291
- synchronization conflicts, obtaining information 136

business object declarations

- setting PDQ examples 162

business object filters

- stand-alone deployments 162

- buttons, about and behavior** 44

C**Calendar**

- buttons, configuring 49
- color codes, configuring 47
- screen, using to add a contact call 247
- settings, user properties 47

- view, using to add general activities 258

call management procedures

- account calls, about 253
- account calls, about recording 255
- account calls, adding from Account Call view 255
- account calls, adding from Accounts view 253
- all contacts, showing 251
- completed account call, submitting 256
- completed contact call, submitting 251
- contact calls, adding using Calendar screen 247
- contact calls, adding using Contact Calls view 246
- contact calls, adding using Contacts view 244
- decision issues discussed during call, recording information about 249
- entering attendee information for account call 256
- list of procedures 243
- products detailed information, entering for account call 255
- promotional items dropped, recording information about 249
- recording contact calls, about 247
- recording information about products detailed 247
- samples dropped, recording information about 248
- Signature Capture view, changing contacts 251
- signature, capturing 250
- Smart Call template, applying 247

calls, planning and recording

- about 241
- Auto Call, about 241
- Auto Update, about 241
- contact and account call, about 241
- planning and recording calls 243
- planning in Siebel Pharma application and recording calls in Handheld application 242
- Smart Call, about 241

- calls, preparing receipts** 157

- Cell tag** 337

- changing contact or address of existing calls** 253

Class Definitions

- creating 54

client

- installation, troubleshooting 275
- performance, troubleshooting 284

- Client Wins conflict resolution** 197
- columns, working with** 209
- COM Extensibility, about** 383
- Comment print tag** 342
- CompactFlash card**
 - backing up to 180
 - Siebel Handheld application, setting installation on 182
- Companion Synchronization**
 - architecture 22
 - checklist, using to troubleshoot 276
 - Handheld application, using to install 202
 - log files, using for troubleshooting 280
 - patches, deploying using 186, 190
- completed contact call, submitting** 251
- component level settings, configuring** 111
- Concat print tag** 365
- Conditional print tags** 366
- configuring**
 - Siebel Handheld Client 23
 - Web Client and Handheld Client, differences in configuring 26
- configuring calendar buttons** 49
- configuring calendar settings** 46
- configuring the goto view button** 48
- Conflict Administration screen** 99
- conflicts**
 - See synchronization, conflict handling and recovery
- contact calls**
 - account calls, about 253
 - account calls, adding from Account Call view 255
 - account calls, adding from Accounts view 253
 - all contacts, showing 251
 - Calendar screen, using to add contact calls 247
 - completed contact call, submitting 251
 - Contact view, using to add contact calls 244
 - decision issues discussed during call, recording information about 249
 - promotional items dropped, recording information about 249
 - recording contact calls, about 247
 - recording information about products detailed 247
 - samples dropped, recording information about 248
 - Signature Capture view, changing contacts 251
 - signature, capturing 250
 - Smart Call template, applying 247
 - view 246

- contacts**
 - contact calls, about 241
 - MedEd events, inviting contacts to 272
 - showing contacts all regardless of affiliation 256
 - view, using to add contact calls 244
- contacts, managing**
 - contact addresses, adding 229
 - contact information, adding 228
 - contact, specifying best time to call 230
 - contacts and accounts, indicating affiliations 231
 - contacts, adding 228
 - contacts, described 226, 227
 - license information, adding and editing 230
 - notes, adding private or shared notes about a contact 231
 - prescription data, viewing 232
- creating a new Sample Drop based on an existing contact call for a new contact** 252
- creating a new Sample Drop contact call based on an existing call** 253

D

- data**
 - automatic backup after synchronization 213
 - backing up, about and restoring 212
 - columns, working with 209
 - Direct Server Synchronization, using with Siebel Handheld Application Client 217
 - Direct Server Synchronization, using with Siebel Handheld Client 217
 - entering, about 209
 - exporting 211
 - find, about using 210
 - queries, about 210
 - restoring data from a backup 213
 - sales data, viewing 239
 - Siebel Handheld application synchronization, using with Siebel Handheld Client 214
 - synchronizing data, about 214
- data filtering**
 - business object declaration examples 162
 - business object filters, general and default 161
 - developing data filters, steps 159
 - popup visibility, overriding 165
 - stand-alone deployments 162
 - visibility, designating 164
- database**

- backup, restoring from 181
 - database backup parameters, editing 180
 - external media, backing up to 180
 - user database backups, about enabling 180
 - database backups**
 - about and restoring 212
 - automatic backup after synchronization 213
 - restoring data from a backup 213
 - troubleshooting 275
 - DEA number, validating** 157
 - decision issues discussed during call**
 - about recording information 249
 - default synchronization filters, about** 161
 - deploying**
 - attachments, deploying with Companion Synchronization 190
 - attachments, deploying with Direct Server Synchronization 188
 - CompactFlash card, setting Handheld application installation on 182
 - DSS components, using Siebel Server installer 169
 - external media, installing Handheld application from 183
 - multiple synchronization servers 174
 - optimizing server process management 173
 - patches, deploying with Companion Synchronization 186
 - patches, deploying with Direct Server Synchronization 184
 - patches, distributing 183
 - patches, downloading 192
 - print templates, installing 181
 - server installation, server topology overview 168
 - server installation, support of Siebel Handheld Client 167
 - server logging levels, configuring 171
 - setup.ini file, editing 175
 - Siebel Handheld application, process of installing 175
 - Siebel Handheld Client, issues and validation 80
 - Siebel Handheld Direct Server Synchronization, server configuration 171
 - Siebel Handheld Direct Server Synchronization, server installation 169
 - stand-alone installer, creating 183
 - synchronization performance and scalability 173
 - synchronization security 174
 - SyncURL, changing on handheld device 182
 - upgrade kits, creating to remotely deploy patch files 188
 - user database backups, enabling 180
 - Design Data Filters** 221
 - diagnostics, enabling SQLTrace** 282
 - Direct Server Synchronization**
 - architecture 22
 - checklist, using to troubleshoot 277
 - deploying, about 168
 - deployments, stand-alone deployments 162
 - described 21
 - Handheld application, using to install 202
 - log files, using for troubleshooting 279
 - network performance, troubleshooting 285
 - optimizing server process management 173
 - patches, deploying using 184, 188
 - server performance, troubleshooting 284
 - Siebel Handheld Client, using with 217
 - Direct Server Synchronization by proxy** 22
 - Disburse on Signature Capture, system parameter** 143, 262
 - disclaimer text, administering** 144
 - distributor names, configuring** 144
 - Divider print tag** 343
 - documents, defining** 84
 - Downloading a Database Extract** 126
 - downloading patches** 192
 - DrawRect print tag** 344
 - drilldowns, about** 46
 - drilling down, about** 208
 - DSS components**
 - using Siebel Server installer 169
 - dynamic drilldown** 51
 - from the daily calendar view to Med Ed Event 51
 - dynamic drilldown, about** 208
 - dynamic hyperlink, about** 46
- E**
- ellipsis, and expandable fields** 208
 - Enabling BatchSync for Siebel Handheld Applications** 124
 - end user error messages**
 - See troubleshooting
 - EndOfLine print tag** 345
 - EndOfPage print tag** 345
 - enhanced find, configuring best times** 54
 - error messages. See troubleshooting errors**
 - error data, accessing 199
 - error details, obtaining from synchronization session 136
 - general synchronization errors 219, 220

- synchronization transaction conflicts 219
- exchanging samples**
 - creating a samples transfer 268
- expense reports**
 - creating 273
 - detailed line items, adding 273
- exporting data** 211
- extended insert processing**
 - about and synchronization 198
 - processing settings, recommended 199
- extended pick processing**
 - about and synchronization 195
 - enabling 196
 - processing settings, recommended 197
- external media**
 - backing up to 180
 - Siebel Handheld application, installing from 183
 - stand-alone installer, creating 183

F

- find**
 - about using 210
 - enhancements, configuring 53
- Footer print tag** 346
- form applets, configuring**
 - about 39
 - designing, about 84
 - group boxes, about 40
 - labels, about 39
 - layout sequence 39
 - radio buttons, about 40
- form UI applets, configuring**
 - about 40
- Format print tag** 346
- Frame print tag** 347
- Full Sync** 124

G

- general activities**
 - See activities, creating general activities
- general synchronization filters, about** 161
- GetCount print tag** 355
- GetDate print tag** 356
- GetDirect print tag** 356
- GetField print tag** 357
- GetRegistry print tag** 358
- GetTime print tag** 359
- GetTotal print tag** 359
- GetX print tag** 361
- GetY print tag** 362
- Goto View**
 - configuring button 48

- GotoView method, using** 33
- group boxes, about configuring** 40
- Guidelines** 221
- guidelines**
 - Siebel Handheld Client, configuration guidelines 28
- Guidelines for Designing Data Filters** 221
- Guidelines for Reducing the Synchronization Time** 220
- Guidelines on Improving the Configuration of the Siebel Handheld Application** 223

H

- Handheld application interface**
 - See interface
- Handheld application synchronization**
 - improving performance, about 220
 - See synchronizing
- handheld device**
 - list of log files 281
 - logging synchronization errors 281
 - working with different specifications 35
- Header print tag** 348
- History button, about** 208
- Home Page applet, using** 44
- hot keys**
 - about locked keyboard hot keys 250
- hyperlinks, about** 46

I

- inactive inventory period** 263
- initial master inventory**
 - establishing, about 263
 - initial count adjustments, creating and submitting 265
 - products, adding to inventory and submitting initial count 264
- installation**
 - Companion Synchronization, using to install Handheld application 202
 - Direct Server Synchronization, using to install Handheld application 202
 - reinstalling the Handheld application 201
 - software on handheld devices, about 202
 - uninstalling the Handheld application 201
 - upgrading from a previous release 201
- installing**
 - attachments, deploying with Companion Synchronization 190
 - attachments, deploying with Direct Server Synchronization 188
 - CompactFlash card, setting Handheld

- application installation on 182
 - Direct Server Synchronization checklist, using 277
 - external media, installing Handheld application from 183
 - patches, deploying with Companion Synchronization 186
 - patches, deploying with Direct Server Synchronization 184
 - patches, distributing 183
 - patches, downloading 192
 - resynchronization problems 278
 - Siebel Handheld application, editing setup.ini file 175
 - Siebel Handheld application, process of installing 175
 - stand-alone installer, creating 183
 - synchronization, about problems caused by 276
 - SyncURL, changing on handheld device 182
 - the client, troubleshooting 275
 - upgrade kits, creating to remotely deploy patch files 188
 - user database backups, enabling 180
 - installing the Handheld application** 201
 - interface**
 - application-level menu 206
 - components, about and diagram 205
 - expandable fields 208
 - More Info view, navigation level 217
 - queries drop-down list 207
 - Screens drop-down list, about 206
 - Screens drop-down list, navigation level 207
 - Show drop-down list, about 206
 - Show drop-down list, navigation level 207
 - Siebel toolbar, about 207
 - status bar 207
 - toggling between applets 207
 - URLs in applet fields 208
 - inventory**
 - initial count adjustments, creating and submitting 265
 - initial inventory count, about establishing 263
 - inventory counts, adjusting samples 270
 - inventory periods, configuring user properties for closing 148
 - inventory periods, types of 263
 - locations, setting up 112
 - products, adding to inventory and submitting initial count 264
- K**
- keyboard hot keys, about locked** 250
- L**
- labels, about configuring** 39
 - Landscape print tag** 349
 - license information**
 - adding and editing 230
 - LineTo print tag** 351
 - list applets, configuring**
 - about 37, 38
 - designing, about 85
 - recommended strategy 37
 - List print tag** 349
 - log files**
 - Companion Synchronization logs, using for troubleshooting 280
 - Direct Server Synchronization logs, using for troubleshooting 279
 - handheld device logs, using for troubleshooting 281
 - Logging into Siebel Pharma Handheld** 226
- M**
- Margin print tag** 352
 - MarginBottom parameter** 374
 - MarginLeft parameter** 374
 - MarginRight parameter** 374
 - MarginTop parameter** 375
 - MedEd events, setting**
 - about 271
 - contacts, inviting to 272
 - creating a MedEd event 271
 - media**
 - backing up to external media 180
 - Siebel Handheld application, installing from external media 183
 - stand-alone installer, creating 183
 - medical education**
 - See MedEd events, setting
 - memory, increasing** 220
 - menu bar, about** 45
 - Microsoft ActiveSync, about** 22
 - Mobile Web Client**
 - note, about receiving and installing upgrade kits 188
 - modifying applets** 62
 - More Info view, navigation level** 217
 - multi-form, multi-quantity printing, configuring** 87
 - multiple synchronization servers** 174

N**Name field drop-down list**

samples do not appear 248

naming conventions, about 30**network performance, about** 285**Non UI fields**

extract during synchronization 62

notes

adding about accounts 238

adding private or shared notes about a contact 231

O**objections**

decision issues discussed during call 249

objects

caution, do not modify 308

Offset print tag 362**P****PageBreak print tag** 353**PaperHeight parameter** 375**PaperSize parameter** 375**PaperWidth parameter** 375**Parameters for Scheduling Handheld**

Database Extract Jobs 127

parent form applet, about 33**patches**

Companion Synchronization, deploying using 186, 190

Direct Server Synchronization, deploying using 184, 188

distributing 183

downloading to the handheld device 192

upgrade kits, creating to remotely deploy patch files 188

PDQs, and Queries drop-down list 207**Performance** 220, 221, 222, 223**performance**

application memory, increasing 220

client performance, about 284

improving by identifying user activities 28

improving performance, about 220

network performance, about 285

server performance, about 284

SQLTrace and synchronization 282

pick applets

business object declaration examples 162

creating 55

popup visibility, overriding 165

visibility, designating 164

pick lists, modifying and creating 57**Picture print tag** 339**popup applets**

extracting 74

extracting Contact Find and Account Find 75

pop-up lists, about using auto pop-up lists 41**pop-up text areas in list applets, configuring** 38**popup visibility, overriding** 165**Portrait print tag** 353**predefined queries, and Queries drop-down list** 207**preferences, setting user preferences** 212**prescription data, viewing** 232**print applets, about designing** 84**Print buttons, configuring** 85**print tagging language** 333

See print templates

print templates

Absolute print tag 334

Background print tag 341

Cell tag 337

Comment print tag 342

Concat print tag 365

Conditional print tags 366

creating print templates 86

Divider print tag 343

DrawRect print tag 344

EndOfLine print tag 345

EndOfPage print tag 345

Footer print tag 346

Format print tag 346

Frame print tag 347

GetCount print tag 355

GetDate print tag 356

GetDirect print tag 356

GetField print tag 357

GetRegistry print tag 358

GetTime print tag 359

GetTotal print tag 359

GetX print tag 361

GetY print tag 362

Header print tag 348

Landscape print tag 349

LineTo print tag 351

List print tag 349

Margin print tag 352

Offset print tag 362

PageBreak print tag 353

Picture print tag 339

Portrait print tag 353

print template file example 369

print template file, creating overview 333, 383

Relative print tag 334

- SetVariable print tag 360
 - SetX print tag 363
 - SetY print tag 364
 - sql (WHERE) print tag 367
 - Static Text print tag 341
 - Title print tag 354
 - Variable parameters print tag 369
 - variables, example using in template 369
 - views, associated with 34
 - print templates, installing** 181
 - printer settings dialog box** 86
 - PrinterBaudrate parameter** 376
 - PrinterCompressed parameter** 376
 - PrinterDensity parameter** 376
 - PrinterDither parameter** 377
 - PrinterDraftMode parameter** 377
 - PrinterFormFeed parameter** 377
 - PrinterHandshake parameter** 378
 - PrinterOrientation parameter** 378
 - PrinterPort parameter** 379
 - PrinterType parameter** 379
 - PrinterUserSelect parameter** 380
 - printing**
 - receipts for samples dropped 157
 - printing with the Handheld application** 210
 - printing, configuration settings**
 - default printing settings, list of 381
 - MarginBottom parameter 374
 - MarginLeft parameter 374
 - MarginRight parameter 374
 - MarginTop parameter 375
 - PaperHeight parameter 375
 - PaperSize parameter 375
 - PaperWidth parameter 375
 - parameters and syntax 373
 - PrinterBaudrate parameter 376
 - PrinterCompressed parameter 376
 - PrinterDensity parameter 376
 - PrinterDither parameter 377
 - PrinterDraftMode parameter 377
 - PrinterFormFeed parameter 377
 - PrinterHandshake parameter 378
 - PrinterOrientation parameter 378
 - PrinterPort parameter 379
 - PrinterType parameter 379
 - PrinterUserSelect parameter 380
 - printing, configuring from Handheld application**
 - about 83
 - applet, configuring for printing 85
 - documents, defining 84
 - form applets, designing 84
 - list applets, designing 85
 - multi-form, multi-quantity printing 87
 - print applets, designing 84
 - Print buttons, configuring 85
 - print templates, creating 86
 - printer settings dialog box 86
 - private notes**
 - adding for accounts 238
 - adding for contacts 231
 - Process for Administering Batch Sync for Siebel Handheld Applications** 125
 - products**
 - products detailed information, entering for account call 255
 - recording information about products detailed 247
 - samples transaction, adding to 270
 - professionals**
 - affiliations, indicating between accounts and professionals 238
 - MedEd event, inviting to 272
 - projects**
 - naming conventions, and 30
 - promotional items dropped, recording information about** 249
- Q**
- queries**
 - about using 210
 - drop down list, about 207
- R**
- radio buttons, about configuring** 40
 - receipts, preparing for calls** 157
 - reconciled inventory period** 263
 - record navigation buttons** 209
 - recording account calls**
 - about 255
 - attendee information, entering for account call 256
 - completed account call, submitting 256
 - products detailed information, entering for account call 255
 - recording contact calls**
 - about 247
 - account calls, about 253
 - account calls, adding from Account Call view 255
 - account calls, adding from Accounts view 253
 - all contacts, showing 251
 - completed contact call, submitting 251
 - decision issues discussed during call, recording information about 249
 - promotional items dropped, recording

- information about 249
 - recording information about products
 - details 247
 - samples dropped, recording information 248
 - Signature Capture view, changing
 - contacts 251
 - signature, capturing 250
 - records, navigating a list of**
 - about 208
 - Back button 208
 - record navigation buttons 209
 - Reduce size of RML file** 222
 - Reduce Synchronization Time** 222
 - Reduce the amount of Data Download** 220
 - Reduce the Time of Data Extraction** 222
 - reinstalling the Handheld application** 201
 - Relative print tag** 334
 - removing**
 - business component filter setting 101
 - business object filter section 103
 - reports**
 - associating views with 34
 - expense reports, adding detailed line
 - items 273
 - expense reports, creating 273
 - repository, compiling** 81
 - responsibilities, assigning** 99
 - restoration, troubleshooting** 275
 - restoring data**
 - from a backup 213
 - Siebel Handheld Client, using 212
 - restoring from a backup** 181
 - returning samples**
 - creating a samples transfer 268
 - Review Configuration** 223
 - Reviewing Batch Sync Status** 127
 - running a batch extract** 129
 - running a batch transaction process** 129
 - Rx Data view**
 - using to view prescription data 232
- S**
- sales data, viewing** 239
 - sample records, managing** 143
 - samples**
 - Name field drop-down list, samples do not
 - appear 248
 - recording information about samples
 - dropped 248
 - samples, managing**
 - about 259
 - initial count adjustments, creating and
 - submitting 265
 - initial inventory count, about
 - establishing 263
 - inventory periods, types of 263
 - managing samples scenario 260
 - products, adding to inventory and submitting
 - initial count 264
 - samples management, working with
 - calls 262
 - samples order, creating 267
 - samples transaction, adjusting quantity 271
 - samples transfer, acknowledging receipt 269
 - samples transfer, adjusting 270
 - samples transfer, creating 268
 - samples views, about 266
 - server and Siebel Handheld application
 - samples records, keeping them
 - synchronized 261
 - screens**
 - administrative screens, about 97
 - designing screens for Siebel Handheld,
 - about 32
 - screen allocation 33
 - views, and 287
 - Screens drop-down list**
 - about 206
 - navigation level 207
 - scripting**
 - Siebel Handheld Client, issues and
 - validation 80
 - server**
 - logging levels, configuring 171
 - multiple synchronization servers 174
 - performance, troubleshooting 284
 - server installation**
 - DSS components, using Siebel Server
 - installer 169
 - hardware and network configuration
 - recommended 168
 - optimizing server process management 173
 - server logging levels, configuring 171
 - server topology overview 168
 - Siebel Handheld Client, support of 167
 - Siebel Handheld Direct Server
 - Synchronization, server installation
 - process 169, 171
 - ServiceCE Generic, about** 45
 - Session Administration screen** 98
 - SetVariable print tag** 360
 - SetX print tag** 363
 - SetY print tag** 364
 - shared notes**
 - adding for accounts 238
 - adding for contacts 231
 - SHCE COM Extension Component (SCEC)**

- about 383
- implementing 390
- Show drop-down list**
 - about 206
 - navigation level 207
- Siebel Handheld Application**
 - general synchronization errors 220
- Siebel Handheld application**
 - and device specifications 35
 - attachments, deploying with Companion Synchronization 190
 - attachments, deploying with Direct Server Synchronization 188
 - CompactFlash card, setting installation on 182
 - external media, installing from 183
 - form UI applets, about configuring 40
 - patches, deploying with Companion Synchronization 186
 - patches, deploying with Direct Server Synchronization 184
 - patches, distributing 183
 - patches, downloading 192
 - process of configuring 26
 - process of installing 175
 - projects, about creating and naming conventions 30
 - setup.ini file, editing 175
 - stand-alone installer, creating 183
 - synchronization, general synchronization errors 219
 - SyncURL, changing on handheld device 182
 - upgrade kits, creating to remotely deploy patch files 188
 - user database backups, enabling 180
- Siebel Handheld application synchronization**
 - application memory, increasing 220
 - Direct Server Synchronization, using with Siebel Handheld Client 217
 - general synchronization errors 219
 - Siebel Handheld Client, using 214
 - synchronization transaction conflicts 219
- Siebel Handheld Client**
 - alpha tabs, configuring 38
 - application memory, increasing 220
 - auto-pop-up list, using 41
 - backing up and restoring data, using 212
 - configuration guidelines 28
 - Direct Server Synchronization, using with Siebel Handheld Client 217
 - form applets, about configuring 39
 - general synchronization errors 219
 - hardware and network configuration recommended 168
 - Home page applet, about 44
 - hyperlinks, about 46
 - improving performance, about 220
 - list applets, configuring 37
 - menu bar, about 45
 - multi-value groups, about 44
 - pop-up text areas in list applets, configuring 38
 - printing, configuring 83
 - repository, compiling 81
 - scripting, issues and validation 80
 - server installation, support of Siebel Handheld Client 167
 - server topology overview 168
 - Siebel Web Client, differences in configuring 26
 - status bar, about 46
 - synchronization transaction conflicts 219
 - tabbed applets, configuring 33
 - testing the Handheld application 94
 - toolbars, about 45
 - tree applets, configuring 42
 - unsupported functions 27
 - user functionality, defining 28
 - user interface elements, about configuring 36
- Siebel Handheld Direct Server Synchronization**
 - DSS components, using installer 169
 - optimizing server process management 173
 - server configuration 171
 - server installation 169
 - server logging levels, configuring 171
- Siebel Mobile Web Client**
 - caution, about submitting calls 252
- Siebel Pharma Handheld**
 - applet classes, list of 301
 - business component classes, list of 297
 - business components, list of 291
 - objects (list), do not modify 308
 - screens and views, list of 287
- Siebel Pharma Handheld application, administering**
 - administering process 142
 - DEA number, validating 157
 - disclaimer text, administering 144
 - distributor names, configuring 144
 - inventory period, configuring user properties for closing 148
 - receipts, preparing for calls 157
 - sample records, managing 143
 - sign and submit button, validation logic 150
 - signature audit, performing 145
 - signature capture verification, configuring

- user properties 149
 - signature capture, verifying 146
- Siebel Task UI, developing tasks in Siebel Handheld applications** 77
- Siebel Tasks, for Siebel Handheld applications**
 - creating 78
 - designing 77
 - developing 77
 - example of how to create 79
- Siebel toolbar, about** 207
- Siebel Web Client**
 - Siebel Handheld Client, differences in configuring 26
 - unsupported functions 27
- Sign button, validation logic** 150
- Signature Capture view**
 - changing contacts 251
- signatures**
 - audit, performing 145
 - capture verification, configuring user properties 149
 - capturing for recording contact calls 250
 - checksums, outcomes of 146
 - Signature Capture view, changing contacts 251
 - signature key, about 146
 - user properties 147
 - verifying signature capture, about 146
- Smart Call**
 - about 241
 - contact calls shortcut 244
 - Smart Call template, applying 247
- software, about installing on handheld device** 202
- sql (WHERE) print tag** 367
- SQLTrace, enabling** 282
- .SRF file**
 - compiling repository file 81
- SSL (Secure Sockets Layer)**
 - synchronization security 174
- stand-alone deployments, about** 162
- stand-alone installer, creating** 183
- Static Text print tag** 341
- status bar**
 - application window, appears at bottom of 207
 - toolbar, located above 46
- Submit button, validation logic** 150
- submitting**
 - completed account call 256
- synchronization server, described** 22
- synchronization sessions**
 - error details, obtaining from 136
 - monitoring 135
 - monitoring, about 134
 - synchronization conflicts, obtaining information 136
 - synchronization sessions, obtaining an audit trail 137
 - synchronization transaction errors, responding to 137
- synchronization, conflict handling and recovery**
 - about 195
 - Client Wins conflict resolution 197
 - error data, accessing 199
 - extended insert pick processing settings, recommended 199
 - extended insert pick processing, about 198
 - extended pick processing settings, recommended 197
 - extended pick processing, about 195
 - extended pick processing, enabling 196
- synchronizing**
 - application memory, increasing 220
 - automatic backup after synchronization 213
 - Companion Synchronization 22
 - data, about 214
 - Direct Server Synchronization 22
 - Direct Server Synchronization by means of proxy 22
 - Direct Server Synchronization, using with Siebel Handheld Client 217
 - filters, general and default business object filters 161
 - general synchronization errors 219, 220
 - handheld application synchronization errors, logging 281
 - improving performance, about 220
 - methods and architecture 21
 - multiple synchronization servers 174
 - overview 20
 - performance and scalability, about 173
 - resynchronization problems, troubleshooting 278
 - servers, types of 22
 - Siebel Handheld application synchronization, using with Siebel Handheld Client 214
 - synchronization security 174
 - transaction conflicts 219
 - troubleshooting, about problems caused by 276
- SyncURL, changing on handheld device** 182

T**tabbed applets, configuring**

about 33

tasks

creating Siebel tasks 78
 designing Siebel tasks 77
 developing Siebel tasks 77
 example Siebel task 79

templates

install templates, installing 181
 print templates, creating 86
 print templates, views associated with 34
 Smart Call template, applying 247

testing the Handheld application 94**timeout parameter**

changing on Windows 2000 283

Title print tag 354**tooggling between applets** 207**toolbars**

about 207
 toolbar functions 45

tree applets, about configuring 42**troubleshooting**

client installation, backup, and restoration 275
 client performance, about 284
 Companion Synchronization checklist, using 276
 Companion Synchronization log files 280
 Direct Server Synchronization checklist, using 277
 Direct Server Synchronization log files 279
 general synchronization errors 219
 handheld device logs 281
 network performance, about 285
 resynchronization problems 278
 server performance, about 284
 SQLTrace, enabling 282
 synchronization transaction conflicts 219
 synchronization, about problems caused by 276
 timeout parameter, changing on Windows 2000 283

TxnProcOk 129**TxnRcvOk** 129**U****UI**

See user interface elements, configuring

uninstalling the Handheld application 201**upgrade kits**

creating to remotely deploy patch files 188

upgrading

Handheld application 201

Upload Only 124**Upload Only, Auto Download Later** 124**URLs, in applet fields** 208**User Administration screen** 97**user database backups**

backup, restoring from 181
 database backup parameters, editing 180
 enabling, about 180
 external media, backing up to 180

user functionality, about defining 28**user interface elements, configuring**

about 36
 alpha tabs, about configuring 38
 auto pop-up lists, about using 41
 buttons, about and behavior 44
 form applets, about configuring 39
 form UI applets, about configuring 40
 Home Page applet, about 44
 hyperlinks, about 46
 list applets, about configuring 37
 menu bar, about 45
 pop-up text areas in list applets 38
 status bar, about using 46
 tabbed applets, about configuring 33
 toolbars functions 45
 tree applets, about configuring 42

user preferences, about customizing 212**user properties**

AlphaTab Named Search 39, 309
 Attendee Call ReadOnly 309
 AutoPopupField 310
 AutoSelect 310
 BarcodeNewRecord 113, 310
 BC Read Only Field 310
 BorderStyle 94, 310
 BrowserPath 88, 311
 Cascading Fields 311
 ConfirmAccept 94, 311
 Credit Payment Dec Types 311
 Credit Payment Inc Types 311
 DefaultViewType 33, 311
 Disable Second Applet 312
 DisableAlphaTabGroups 39, 312
 DisableParentBar 33, 313
 Document Type Field 313
 Document Type LOV Type 313
 Document Type1 313
 Document Type2 313
 Electronic Signature Read Only 313
 EncryptKey 94, 313
 Expired Product Message Box 313
 Explorer Applet Size 44, 313
 Field Read Only Field fieldname 314

- FormUI 40, 79, 314
- Fulfilled Order Status 314
- HandheldInsertFailMode 314
- HandheldSyncPickMode 314
- InkColor 94, 315
- InkWidth 94, 315
- Max Auto Generated Priority 315
- Maximum Periods 316
- MaximumFooterLines 94, 316
- MaximumHeaderLines 94, 316
- Multicurrency 316
- Must Detail Products 316
- On Field Update Set n 317
- OnlySubmitByPosition 317
- OrderAccessControl 317
- OverrideDefaultApplet 34, 317
- OverridePopupVisibility 317
- Paper Reference Number Required 318
- Paper Signature Read Only 318
- PrintForms 88, 318
- Propagate Address BC Fields 319
- Propagate Address Fields To Parent BC 320
- Read Only If Status Synchronized 320
- Read-Only Order Status 321
- Sample Disbursed Required 321
- SHCE COM Disable 321
- SHCEDailySlotHeight 321
- SHCEHideStatusBar 47, 321
- SHCEHideToolBar 47, 322
- SHCEMenuBackButton 47, 322
- SHCEMenuForwardButton 47, 322
- SHCEMenuNewButton 47, 322
- ShowData 94, 323
- Signature 94, 324
- Signature Body Text Field
n 323
- Signature Date 323
- Signature Time 323
- SignatureDataField 94, 324
- SignatureDate 94
- SignatureField 324
- SignatureFooterField 94, 324
- SignatureFooterSaveField 94, 324
- SignatureHeaderField 94, 325
- SignatureHeaderSaveField 94, 325
- SignatureHeight 94, 325
- SignatureTime 94
- SignatureWidth 325
- SingleSelect 325
- Skip Conflict Resolution for Duplicate
Records 326
- SRE Professional Types 326
- SRE Reference Number Required 326
- SubCall Component CE 326
- Submit At Handheld Sync 327
- Synchronize Rollup Data 327
- TaskUI 79, 327
- Template Components 327
- Template Fields 328
- TriggerFindRecord 328
- Two Digit Currency Rounding 328
- Update After Recreate Receipt 328
- Update After Submit 329
- Update Inter Table BC Fields 329
- Update Status To Synchronized 329
- Use Document Type Field 328
- UseSystemLocale 47, 330
- Validate DEA Number 330
- Validate Product Priority 330
- validation, in Pharma Professional Call-CE
Business Component 331
- View 34, 331
- ViewType 33, 43, 47, 332
- WritingGuide 94, 332
- users, setting up**
 - barcode definition, creating 122
 - barcode definition, process of creating 122
 - barcode scanning, enabling the SHCE Service
FS Activity Parts Movements
views 123
 - barcode scanning, enabling views for 119
 - barcode settings, administering 112
 - business component filter setting,
adding 101
 - business object filter section, adding 102
 - business object filter section, removing 103
 - component level settings, configuring 111
 - enhancing the behavior of barcode scan 119
 - error details, obtaining from synchronization
sessions 136
 - new barcode definitions, creating 114
 - process tasks 99
 - responsibilities, assigning 99
 - synchronization conflicts, obtaining
information 136
 - synchronization sessions, about
monitoring 134
 - synchronization sessions, monitoring 135
 - synchronization sessions, obtaining an audit
trail 137
 - synchronization transaction errors,
responding to 137
 - user level settings, removing 101
 - views, administering 99
 - views, methods for specifying 100

V

Variable parameters print tag 369

views

- default behavior, overriding 33
- drill-down only 33
- print templates or reports, associated with 34
- Siebel Handheld, about designing views for 32

views, administering

- business component filter setting, adding 101
- business object filter section, adding 102
- business object filter section, removing 103
- component level settings, configuring 111

- tasks associated with 99
- user level settings, removing 101
- views, methods for specifying 100

visibility

- designating in default Business Object declarations 164
- popup visibility, overriding 165

W

Web Server timeout errors

- changing timeout parameter on Windows 2000 283

Windows

- changing timeout parameter on Windows 2000 283