

SYBASE®

Developer's Guide

## **Unwired Accelerator**

8.0

DOCUMENT ID: DC00183-01-0800-01

LAST REVISED: July 2006

Copyright © 2000-2006 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase, SYBASE (logo), ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Advantage Database Server, Afaria, Answers Anywhere, Applied Meta, Applied Metacomputing, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Translator, APT-Library, ASEP, Avaki, Avaki (Arrow Design), Avaki Data Grid, AvantGo, Backup Server, BayCam, Beyond Connected, Bit-Wise, BizTracker, Certified PowerBuilder Developer, Certified SYBASE Professional, Certified SYBASE Professional Logo, ClearConnect, Client-Library, Client Services, CodeBank, Column Design, ComponentPack, Connection Manager, Convoy/DM, Copernicus, CSP, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DataWindow .NET, DB-Library, dbQueue, Dejima, Dejima Direct, Developers Workbench, DirectConnect Anywhere, DirectConnect, Distribution Director, Dynamic Mobility Model, e-ADK, E-Anywhere, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, EII Plus, Electronic Case Management, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise Portal (logo), Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, eremote, Everything Works Better When Everything Works Together, EWA, ExtendedAssist, Extended Systems, ExtendedView, Financial Fusion, Financial Fusion (and design), Financial Fusion Server, Formula One, Fusion Powered e-Finance, Fusion Powered Financial Destinations, Fusion Powered STP, Gateway Manager, GeoPoint, GlobalFIX, iAnywhere, iAnywhere Solutions, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InstaHelp, Intelligent Self-Care, InternetBuilder, iremote, irLite, iScript, Jaguar CTS, jConnect for JDBC, KnowledgeBase, Legion, Logical Memory Manager, M2M Anywhere, Mach Desktop, Mail Anywhere Studio, Mainframe Connect, Maintenance Express, Manage Anywhere Studio, MAP, M-Business Anywhere, M-Business Channel, M-Business Network, M-Business Suite, MDI Access Server, MDI Database Gateway, media.splash, Message Anywhere Server, MetaWorks, MethodSet, mFolio, Mirror Activator, ML Query, MobiCATS, MobileQ, MySupport, Net-Gateway, Net-Library, New Era of Networks, Next Generation Learning, Next Generation Learning Studio, O DEVICE, OASIS, OASIS logo, ObjectConnect, ObjectCycle, OmniConnect, OmniQ, OmniSQL Access Module, OmniSQL Toolkit, OneBridge, Open Biz, Open Business Interchange, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, Partnerships that Work, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, Pharma Anywhere, PhysicalArchitect, Pocket PowerBuilder, PocketBuilder, Power++, Power Through Knowledge, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, Powering the New Economy, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Pylon, Pylon Anywhere, Pylon Application Server, Pylon Conduit, Pylon PIM Server, Pylon Pro, QAnywhere, Rapport, Relational Beans, RemoteWare, RepConnector, Report Workbench, Report-Execute, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Resource Manager, RFID Anywhere, RW-DisplayLib, RW-Library, SAFE, SAFE/PRO, Sales Anywhere, Search Anywhere, SDF, Search Anywhere, Secure SQL Server, Secure SQL Toolset, Security Guardian, ShareSpool, ShareLink, SKILS, smart.partners, smart.parts, smart.script, SOA Anywhere Trademark, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, Stage III Engineering, Startup.Com, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Development Framework, Sybase Financial Server, Sybase Gateways, Sybase IQ, Sybase Learning Connection, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase Synergy Program, Sybase Virtual Server Architecture, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SybFlex, SybMD, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, The Enterprise Client/Server Company, The Extensible Software Platform, The Future Is Wide Open, The Learning Connection, The Model For Client/Server Solutions, The Online Information Center, The Power of One, TotalFix, TradeForce, Transact-SQL, Translation Toolkit, Turning Imagination Into Reality, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Viafone, Viewer, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server, XcelleNet, XP Server, XTNDAccess and XTNDConnect are trademarks of Sybase, Inc. or its subsidiaries. 05/06

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

# Contents

<b>About This Book</b> .....	<b>xi</b>	
<b>CHAPTER 1</b>	<b>Introduction</b> .....	<b>1</b>
	Overview .....	1
	Unwired Accelerator features .....	3
	End-user features .....	3
	Developer features .....	5
<b>CHAPTER 2</b>	<b>Getting Started with Unwired Accelerator</b> .....	<b>7</b>
	Overview .....	7
	Mobile Web Studio terminology .....	8
	Guidelines for developing applications .....	10
	Workflow for creating applications .....	11
<b>CHAPTER 3</b>	<b>Developing Applications</b> .....	<b>13</b>
	Overview .....	13
	Mobile Web Studio tools .....	14
	Application Manager .....	14
	Application Builder .....	15
	Creating Web applications .....	18
	Guidelines for choosing capture options .....	19
	Navigation .....	19
	Capture strategies .....	21
	Creating a new Web application .....	25
	Creating XML applications .....	26
	Creating database applications .....	28
	Database connection cache options .....	30
	Creating JSP applications .....	33
	Using a Web application .....	33
	Using a JSP template .....	36
	Using a remote URL .....	37
	Using Web service applications .....	38
	Examples .....	38

- Using Unwired Accelerator Web services ..... 38
- Creating Web service applications ..... 41
- Creating new UDDI registry entries ..... 47
- Creating HTML-based applications ..... 48
- Creating document applications ..... 49
- Adding new content types ..... 51
- Creating file applications ..... 53
- Creating Domino applications ..... 56
- Setting up connection pools for Domino ..... 57
- Setting up Lotus Notes and Domino on one machine  
    (WebSession) ..... 58
- Creating Domino elements ..... 59
- Understanding UA-Domino agent data flow ..... 61
- Sample code for Domino agents ..... 64
- Creating SAP applications ..... 70
- Creating SAP elements ..... 71
- SAP login ..... 73
- Reordering columns ..... 75
- Saving applications ..... 76
- Content tab ..... 77
- Mobile tab ..... 79
- Roles tab ..... 80
- Presentation tab ..... 80
- List/Detail tab ..... 82
- Data Validation tab ..... 83
- Administration tab ..... 84
- Answers Anywhere ..... 84
- Previewing applications ..... 86
- Approving applications ..... 87
- Viewing and editing applications ..... 87
- Properties ..... 88
- Parameters ..... 89
- Labels ..... 94
- Data types ..... 95
- Finding and replacing applications ..... 96
- Performing an application search ..... 96
- Finding and replacing applications ..... 97

- CHAPTER 4      Advanced Application Creation ..... 101**
- Overview ..... 101
- Server-side click-across ..... 102
- Creating click-across applications ..... 103
- Building click-across applications ..... 103
- Building composite applications ..... 114

Shared context .....	115
Building secure applications.....	127
Setting up Unwired Accelerator to create secure applications	128
Creating secure applications .....	129
Working with data-capable elements .....	130
Data-capable post-processing options .....	132
Using grid rules .....	135
“include” options .....	136
Using grid rules .....	136
Creating chart applications.....	141
Creating application input fields .....	143
Linking applications.....	144
Complex transactions .....	144
Creating update links (with conflict resolution) .....	148
Validating data.....	150
ContextID linkage across applications .....	151
Login application .....	151
Linking parameters.....	152
Handling JSESSIONID.....	154
@OP cookie tags .....	154
Adding JSP-related capabilities .....	155
Generalizing parameters with @OP tags.....	157
Using @OP tags in URLs .....	157
Using @OPCGIPARAM to define a default value .....	158
Custom capture options .....	159
Custom Web capture option .....	160
Crystal Reports Web capture option .....	163
Custom XSL Web capture option .....	165
Business Objects capture strategy.....	168
Capturing Business Objects .....	168
Understanding Business Object JSP templates.....	170
Understanding Page Content Processor JSP .....	173
Remedy capture strategy .....	176
Enabling the Split window .....	182
Enabling Answers Anywhere (natural language search) .....	183
Answers Anywhere Agent Network .....	185
Language syntax .....	186
Resolving ambiguity .....	190
Using SMS messaging .....	194
CellularModemController (CMC) .....	195
Security with e-mail and SMS based request handling .....	195
Enabling applications and pages to support multiple languages .	196
Web browsers and UTF-8 .....	198

<b>CHAPTER 5</b>	<b>Personalizing Application Content .....</b>	<b>199</b>
	Overview .....	200
	Adapter types .....	200
	Creating new personalization adapter keys .....	201
	Building applications with personalization adapters .....	202
	Using the user name/password adapter .....	202
	Using the database table adapter .....	202
	Entering user-specific values in Mobile Web Studio .....	203
	Importing and exporting personalization keys .....	204
<b>CHAPTER 6</b>	<b>Building Templates .....</b>	<b>205</b>
	Overview .....	205
	Template Manager .....	205
	Template Upload window .....	206
	Template types .....	207
	Device Manager .....	212
	Creating and editing templates .....	213
	Changing the default template .....	215
	Customizing user XML element templates .....	216
	Enabling different templates for the same element .....	219
	Customizing online device templates .....	220
<b>CHAPTER 7</b>	<b>Using Unwired Accelerator Alerts .....</b>	<b>225</b>
	Overview .....	225
	Alerts .....	227
	Requirements and restrictions .....	227
	Creating an SMS alert using Mobile Web Studio .....	232
	Creating an e-mail alert using Mobile Web Studio and Portal Interface .....	236
	Using agents .....	241
	Changing an agent's status and running an agent once .....	242
	Schedule tab .....	243
	Rules tab .....	243
	Actions tab .....	249
	Error Management tab .....	253
	Saving a new agent .....	254
	Handling "permission denied" error messages .....	254
	Viewing or changing agent properties .....	255
	Using adapters .....	255
	Creating a file-system adapter .....	256
	Creating an e-mail adapter .....	257
	Creating a database adapter .....	258
	Editing an adapter or viewing the log .....	259

	Using agent servers .....	259
	Editing an agent server .....	261
	Viewing transaction logs .....	261
	Limiting the log size .....	261
<b>CHAPTER 8</b>	<b>Building Catalogs .....</b>	<b>263</b>
	Overview .....	263
	Catalog Manager .....	264
	Catalog Builder .....	265
	Creating and editing catalogs .....	266
	Updating Portal Interface with changes .....	270
<b>CHAPTER 9</b>	<b>Building Pages .....</b>	<b>271</b>
	Overview .....	271
	Page Manager .....	271
	Page Builder .....	272
	Choosing the page type .....	273
	Selecting a page layout .....	274
	Creating and editing pages .....	277
	Updating Portal Interface with changes .....	280
<b>CHAPTER 10</b>	<b>Building Page Groups .....</b>	<b>281</b>
	Overview .....	281
	Page Group Manager .....	282
	Page Group Builder .....	282
	Choosing a navigation style .....	283
	Navigation styles .....	284
	Sample navstyles.xml file .....	287
	Sample *View.xml file .....	288
	Page groups, page numbers, and <IFRAME> .....	289
	Creating and editing page groups .....	289
	Updating Portal Interface with changes .....	293
<b>CHAPTER 11</b>	<b>Deploying Mobile Applications .....</b>	<b>295</b>
	Overview .....	295
	Deploying to BlackBerry and Symbian/J2ME clients .....	296
	Using the BlackBerry user interface .....	299
	Using the Symbian/J2ME user interface .....	302
	Deploying to online devices .....	304
	PDA user interface .....	305
	Deploying to Portal Interface .....	306

<b>CHAPTER 12</b>	<b>Importing and Exporting Portal Objects.....</b>	<b>307</b>
	Overview .....	307
	Importing files into Mobile Web Studio.....	308
	Exporting objects from Mobile Web Studio .....	311
	Rules and restrictions.....	313
	Importing and exporting page groups.....	313
	Exporting composite applications.....	314
	Exporting agents .....	315
	Roles and security integration .....	315
	Restrictions.....	316
	Exporting/importing parameters .....	316
<b>CHAPTER 13</b>	<b>Changing the Portal’s Look and Feel .....</b>	<b>319</b>
	Introduction .....	319
	Editing styles.xml .....	320
	Editing css.css .....	321
	Editing messages.xml .....	322
	Editing global.properties.xml .....	323
<b>APPENDIX A</b>	<b>Troubleshooting .....</b>	<b>325</b>
	Overview .....	325
	SMS user not specified .....	325
	Agent log fills up the database .....	325
	Auto indexer does not work.....	326
	Chart is not updating to show new data .....	326
	Co-brands error .....	326
<b>APPENDIX B</b>	<b>.NET Container Client.....</b>	<b>327</b>
	Overview .....	327
	Installing the .NET container client.....	327
	Understanding .NET client requirements .....	328
	Installing the .NET software .....	330
	Starting and stopping the .NET client.....	335
	Setting up .NET container client accounts .....	336
	Customizing the .NET container client .....	337
	Customization overview .....	338
	Understanding the Strings.xml file.....	341
	Messages in Strings.xml .....	344
	Strings in Strings.xml.....	346
	Images in Strings.xml .....	348
	Understanding the device template files .....	349

<b>Glossary .....</b>	<b>353</b>
<b>Index .....</b>	<b>359</b>



# About This Book

## **Audience**

This book is for anyone developing portals, portlets, applications, agents, templates and so forth for Sybase® Unwired Accelerator.

## **How to use this book**

This guide describes how to develop and deploy applications, portals, and their associated components (pages, page groups, catalogs, and so on).

- Chapter 1, “Introduction,” introduces Unwired Accelerator and explains how to get started using Mobile Web Studio.
- Chapter 2, “Getting Started with Unwired Accelerator,” describes the basic concepts and terminology you need to understand to develop applications.
- Chapter 3, “Developing Applications,” describes how to build and manage a variety of applications.
- Chapter 4, “Advanced Application Creation,” has advanced topics for application development, including creating messaging and portal applications.
- Chapter 5, “Personalizing Application Content,” describes how to create input fields for application values and how users can personalize these values to fit the content they want to see.
- Chapter 6, “Building Templates,” explains how to build and manage templates. Every application is placed on either a default Hyper Text Markup Language (HTML) template, or a user-defined HTML template.
- Chapter 7, “Using Unwired Accelerator Alerts,” describes how to use alerts to schedule or externally trigger automatic processing of an application’s content.
- Chapter 8, “Building Catalogs,” explains how to build and manage catalogs that contain applications. Portal Interface has one default display catalog from which users can select and add applications to their user pages.
- Chapter 9, “Building Pages,” describes how to build predefined pages, from which portal users can select.

- 
- Chapter 10, “Building Page Groups,” explains how to add pages to page groups.
  - Chapter 11, “Deploying Mobile Applications,” explains how to deploy mobile applications to mobile devices.
  - Chapter 12, “Importing and Exporting Portal Objects,” explains how to copy Mobile Web Studio objects from one Mobile Web Studio installation to another and how to update Portal Interface with updated Mobile Web Studio pages, catalogs, and applications.
  - Chapter 13, “Changing the Portal’s Look and Feel,” describes the portal’s primary configuration files and explains how to alter the look and feel of your portal.
  - Appendix A, “Troubleshooting,” provides possible solutions to common Unwired Accelerator problems.
  - Appendix B, “.NET Container Client” describes the .NET container client included with Unwired Accelerator.

#### **Related documents**

**Unwired Accelerator documentation** The following Unwired Accelerator documents are available on the Getting Started with Unwired Accelerator CD:

- The Unwired Accelerator installation guide for your platform explains how to install the Unwired Accelerator software.
- The Unwired Accelerator release bulletin for your platform contains last-minute information not documented elsewhere.
- The *Unwired Accelerator Quick Start Guide* shows how to deploy a database application to mobile devices.
- The *Mobile Application Development Tutorial* provides tutorials that help you get started using Mobile Web Studio to develop and deploy mobile applications.

**Unwired Accelerator online documentation** The Unwired Accelerator documentation set includes:

- The *Unwired Accelerator Administration Guide* provides administration topics for Unwired Accelerator and its configuration.
- The *Portal Interface User’s Guide* describes the Portal Interface user interface and how to use Portal Interface to build and manage your enterprise’s portal.

**jConnect™ for JDBC™ documents** Unwired Accelerator 8.0 includes the jConnect for JDBC driver to allow JDBC access to Sybase database servers and gateways. The *Programmer's Reference jConnect for JDBC* is included on the SyBooks CD.

**Adaptive Server® Anywhere documents** Unwired Accelerator 8.0 includes the ASA database to store system information including security authentication and authorization information. The ASA document set is included on the SyBooks CD.

### Other sources of information

Use the Sybase Getting Started CD, the SyBooks CD, and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

To access the Sybase Product Manuals Web site, go to Product Manuals at <http://www.sybase.com/support/manuals/>.

### Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

#### ❖ Finding the latest information on product certifications

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.

- 
- 2 Select Products from the navigation bar on the left.
  - 3 Select a product name from the product list and click Go.
  - 4 Select the Certification Report filter, specify a time frame, and click Go.
  - 5 Click a Certification Report title to display the report.

❖ **Creating a personalized view of the Sybase Web site (including support pages)**

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Click MySybase and create a MySybase profile.

**Sybase EBFs and software maintenance**

❖ **Finding the latest information on EBFs and software maintenance**

- 1 Point your Web browser to the Sybase Support Page at <http://www.sybase.com/support>.
- 2 Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
- 3 Select a product.
- 4 Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the “Technical Support Contact” role to your MySybase profile.

- 5 Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

**Conventions**

The formatting conventions used in this manual are:

Formatting example	To indicate
commands and methods	When used in descriptive text, this font indicates keywords such as: <ul style="list-style-type: none"> <li>• Command names</li> <li>• C++ and Java method or class names</li> </ul>
<i>variable, package, or component</i>	Italic font indicates: <ul style="list-style-type: none"> <li>• Program variables, such as <i>myCounter</i></li> <li>• Parts of input text that must be substituted, for example: <pre>Server.log</pre> </li> <li>• File names</li> </ul>
File   Save	Menu names and menu items are displayed in plain text. The vertical bar shows you how to navigate menu selections. For example, File   Save indicates “select Save from the File menu.”
package 1	Monospace font indicates: <ul style="list-style-type: none"> <li>• Information that you enter on a command line, or as program text</li> <li>• Example program fragments</li> <li>• Example output fragments</li> </ul>

**Variables**

The variable `%SYBASE%` is used in this manual to represent the Sybase installation directory.

**If you need help**

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.



# Introduction

This guide provides instructions for the mobile application developer using Unwired Accelerator.

<b>Topic</b>	<b>Page</b>
Overview	1
Unwired Accelerator features	3

## Overview

Unwired Accelerator (UA) provides a way to easily connect people, departments, and information across your organization. You can use Unwired Accelerator's content management capabilities to strengthen your company's identity by integrating your company's brand, logos, and data to present a personalized view of the available information and applications.

For mobile applications to be successful, they must be able to:

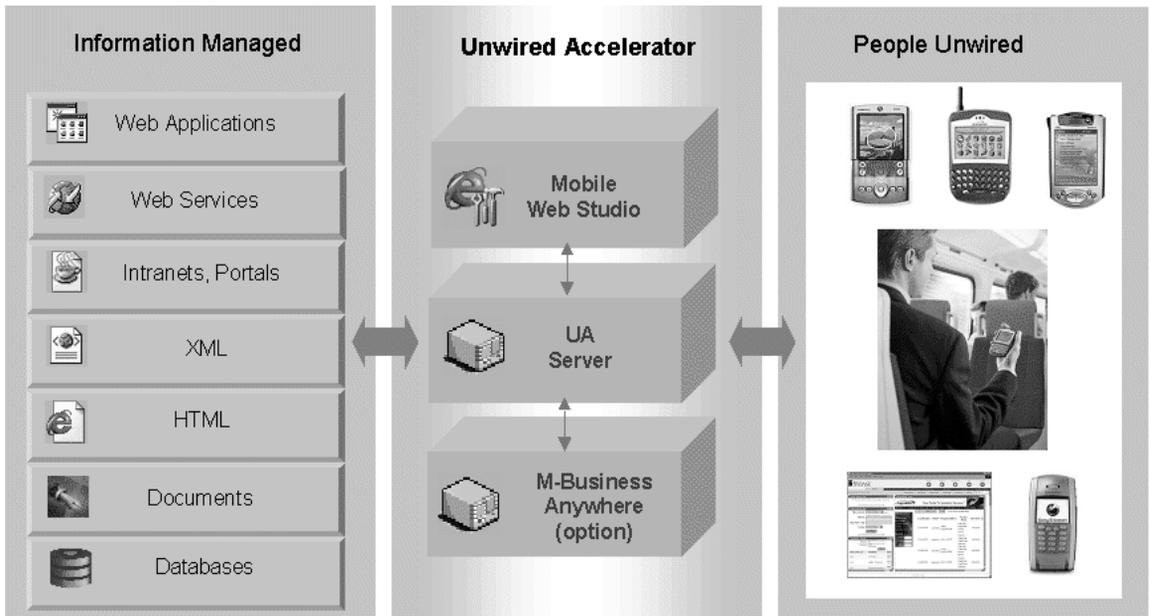
- Run in both connected and disconnected modes
- Run in multiple mobile environments
- Be quickly and easily deployed using Internet technology
- Be quickly and easily developed

The Sybase Unwired Accelerator solution includes several proven technologies: the Mobile Web Studio, a powerful visual development tool; the Mobile Portal Server, a scalable mobile server; Content Capture, Sybase patented technology for repurposing Web applications and content without programming, and M-Business Anywhere, an optional component for delivering content in an occasionally connected model. With these integrated products, you can create secure, enterprise-wide, mobile applications.

Mobile applications provide:

- Better access to customer and product data
- Reduction in error rate
- Reduction of paperwork
- Improved productivity by reducing the need for telephone calls to obtain information
- Better customer service
- Automated data capture and activity tracking

**Figure 1-1: Architectural overview**



**Mobile Web Studio**

Mobile Web Studio is a Web-based rapid development tool that uses patented technology for creating powerful and interactive mobile Web applications or for mobilizing existing Web applications or data sources like SAP, Enterprise Resource Planning (ERP), PeopleSoft ERP, databases, Extensible Markup Language (XML), Web Services, HTML, JavaServer Pages (JSPs), and Active Server Pages (ASPs). Mobile Web Studio is a visual development tool with point-and-click functionality.

Mobile Portal Server	Mobile Portal Server delivers Web applications and content in a connected or disconnected environment, coordinating the communication between M-Business Anywhere and legacy back-end systems in an occasionally connected environment. Based on a scalable and distributed architecture, the Mobile Portal Server runs in the J2EE compliant Web containers, Tomcat and EAServer.
M-Business Anywhere (optional)	M-Business Anywhere is a reliable, scalable, and secure platform for delivering Web-based applications and content to mobile devices in an occasionally connected model.
Content Capture	Content Capture is a Sybase-patented technology for repurposing Web-based applications and content using an object-oriented scripting language. The Sybase capture engine analyzes Web content and stores the feature definition, or signature. At runtime, the Content Capture engine uses pattern matching to locate the same signature. Therefore, even if content has moved or changed, the engine is still likely to find the information.

## Unwired Accelerator features

Unwired Accelerator accelerates the mobilization of enterprise Web applications, online enterprise applications, and data sources for continuous access, which means you can access the same information whether you are connected, disconnected, or occasionally connected to the network.

Unwired Accelerator provides rapid development tools to mobilize existing Web applications, or to create new mobile applications based on your enterprise's data sources without programming.

Unwired Accelerator is the best strategy for getting the most value possible out of your current information infrastructure by getting timely information on the correct device to the most valuable individuals in your enterprise. Mobilization extends the information contained in these systems to new channels, with anytime, anywhere access. Unwired Accelerator leverages your existing infrastructure to create new mobile business opportunities.

## End-user features

The use of mobile application provides many benefits to the end user, such as:

- Better access to customer and product data through online, or always-available, connected access to their applications and data from these devices:
  - PocketPC Internet Explorer running Windows CE 2003
  - Motorola MPx200 (Windows Mobile 2003 OS)
  - HandSpring Treo 600, 650 (PalmOne 5.x OS)
  - Sony Ericsson P900 (Symbian OS)
  - RIM BlackBerry (including Proximus and Vodafone)
- Offline, or occasionally connected, access to their applications and data from these devices:
  - PocketPC
  - Palm
  - Symbian/J2ME
  - RIM BlackBerry (including Proximus and Vodafone)
  - M-Business Anywhere client
- Reduction in error rate and paperwork, which improves customer service.
- Improved productivity by reducing the need for telephone calls and traveling to the office to obtain information.
- Automated data capture and activity tracking.
- Alerts – schedule or externally trigger application content for processing, including writing content to an e-mail message, SMS, the database, or disk. See Chapter 7, “Using Unwired Accelerator Alerts.”
- Online enterprise application mobilization – allows users to access, update, and create new transactions in their SAP, Crystal Reports, and Domino Business Objects systems while they are away from the office.
- Insert and update data from the mobile device.
- Automatic and manual synchronization of the datasource from the UA server and from the mobile device.
- Change the look and feel of the UA interface by device type (for example, language, graphics, colors, and so forth).
- Change the look and feel of mobile application by device type.

## Developer features

Unwired Accelerator makes mobile application development easy with these features:

**Table 1-1: Developer features**

Feature	Function
<b>Mobilization</b>	
Point-and-click interface	Easily create new Web-based and online enterprise mobile applications without programming.
JSP implementation	Implement JSP code snippets to customize mobile applications before, during, or after creation
Application preview	Preview new applications, and preview applications by device display orientation and size.
Application deployment	Deploy a single application to multiple devices, and for online and offline modes.
Transaction management	See Chapter 4, “Advanced Application Creation.”
Conflict resolution	See Chapter 4, “Advanced Application Creation.”
<b>Integration</b>	
Linked applications	Create linked applications for click-across access, and linked parameters for insert, update, or delete at any level.
Create composite applications	Combine multiple services to build a composite application that consists of functionality drawn from a variety of sources within a service-oriented architecture.
Define workflow and communication	Automatically route events or work items from one user or program to another.
<b>Look and feel</b>	
Build and customize templates	Propagate your company’s look and feel across UA while allowing flexibility within standards.
Customize mobile applications	Modify mobile applications for one or more mobile devices, to best present the application on the device.
Define navigation styles	Specify how users’ pages display based on the operating system and browser used.
.NET client API	Develop your own .NET client using the .NET client API and configuration files.
<b>Personalization</b>	
Adapters	Define adapters for personalizing content for each user using authentication. Chapter 5, “Personalizing Application Content.”
Alerts	Schedule or externally trigger automatic processing of an application’s content. See Chapter 7, “Using Unwired Accelerator Alerts.”

<b>Feature</b>	<b>Function</b>
Page layout	Control the layout of your pages with a variety of column widths and the ability to customize individual cell widths. See Chapter 9, “Building Pages.”
Create pages	Create and manage pages to use as containers for mobile applications.
Edit pages	Add, modify, or delete objects from pages.
<b>Connectivity</b>	
Offline access	Content caching
Synchronization	Online – automatically synchronize changes without user intervention, when connected. Offline – synchronize updates, changes and new transactions. Push synchronization from UA and from mobile devices. Manage push synchronization from Mobile Web Studio.
<b>Life cycle management</b>	
Versioning	Allows you to track application changes.

See the *Unwired Accelerator Administration Guide* for information about administrative and security features.

# Getting Started with Unwired Accelerator

This chapter describes the basic concepts and terminology you need to understand to develop applications.

<b>Topic</b>	<b>Page</b>
Overview	7
Mobile Web Studio terminology	8
Guidelines for developing applications	10
Workflow for creating applications	11

## Overview

Some examples of typical business activities than can be mobilized are:

- Sales people can access client information while onsite and respond immediately to client requests, and create new client records.
- Clients can reorder products in real time.
- Service teams in the field can be immediately notified of critical customer situations.
- Customers can conduct financial transactions from any mobile device; access airline schedules and travel updates, set up accounts, or browse an online catalog.
- Executives can be notified of critical changes to the business and can take immediate action.

## Mobile Web Studio terminology

This section discusses some basic concepts and terms you should be familiar with before you start using Unwired Accelerator to develop applications. When you log in to Mobile Web Studio, you can choose the Mobile Web Studio objects that you want to work with and have permission to access.

**Table 2-1: Mobile Web Studio functions**

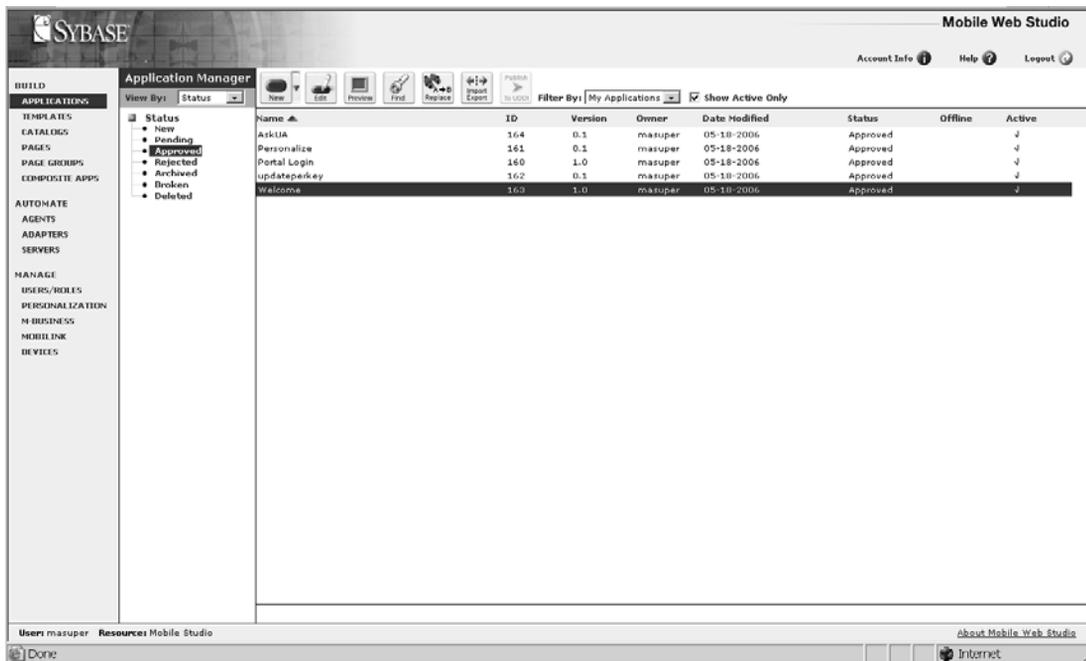
<b>Mobile Web Studio</b>	
<b>Build</b>	
Mobile applications	Create applications that you can deploy to the different types of mobile devices.
Templates	Templates define the organization of applications that contain more than one element by where applications are located, background HTML code, and so on. You can apply the templates you create to applications. You can assign a different template for each device type for a given application. See Chapter 6, “Building Templates.”
Catalogs	Catalogs allow you to create user-defined hierarchies of content for display in an application. See Chapter 8, “Building Catalogs”.
Pages and page groups	Use page groups as containers for mobile applications deployed to a mobile device.
Composite applications	Composite applications allow you to create virtual Web applications using several existing applications. See “Building composite applications” on page 114.
<b>Automate</b>	
Agents (alerts)	<ul style="list-style-type: none"> <li>• E-mail/SMS</li> <li>• Database</li> <li>• File system</li> <li>• Push synchronization</li> </ul> <p>Create, edit, manage, start, stop, and view logs for agents that automatically process application content. See Chapter 7, “Using Unwired Accelerator Alerts.”</p>
Adapters	Create, edit, manage, and view logs for adapters that write application content to e-mails, databases, or file systems. See Chapter 7, “Using Unwired Accelerator Alerts.”
Servers	Create, manage, and view logs for servers on which agents run. See Chapter 7, “Using Unwired Accelerator Alerts.”
<b>Manage</b>	
Resources	Create and manage portal resources, or cobrands.
Users	Provides single sign-on LDAP, user name, and password authentication
Roles	Control access to UA resources by granting roles for security to a single user or group of users
Personalization	Configure application parameter input values to be filled in from adapters that extract values from other sources.
M-Business Anywhere Server	Provide M-Business server functions management for channels, group applications, and users.
MobiLink	View a list of applications that have been deployed to the MobiLink database.

Devices	Create a custom look-and-feel BlackBerry client for a particular resource identifier (RID).
---------	---

<b>Application Builder</b>
----------------------------

Content Capture	<ul style="list-style-type: none"> <li>• Web accessible sources – Web sites and Web applications</li> <li>• XML sources</li> <li>• Databases</li> <li>• JSP or ASP applications</li> <li>• Web Services (or advanced Web Services plus WSDL file objects)</li> <li>• HTML sources</li> <li>• Documents – PDF, Word, Excel, etc.</li> <li>• Domino – build applications from Domino Business Objects</li> <li>• SAP – build applications from any SAP/R3 Remote Function Call</li> </ul> <p>See Chapter 3, “Developing Applications” for detailed information.</p>
Save	Save an application, save an application to another name, and publish to UDDI.
Edit properties	Edit the properties of an application any time. See “Saving applications” on page 76 and “Viewing and editing applications” on page 87.
Configure parameters	Identify or create application parameters, such as CGI scripts for Web elements or custom @OP parameters for XML and database applications. See “Parameters” on page 89.
Labels	Create custom field labels for data-capable elements. See “Labels” on page 94.
Data Type	Define special data types, such as integer, string, E-mail, phone, address, calendar, task, contact, or add a new data type. See “Data types” on page 95.
Templates	Portal, BlackBerry Online, Nokia Online, Palm Online, PPC Online, Smartphone Online, WAP-HTML, WAP-WML, PDA Online
Preview	Preview application playback, XML, and edit screen, and preview by device type. See “Previewing applications” on page 86.

Figure 2-1: Mobile Web Studio



## Guidelines for developing applications

Applications are the primary objects you work with in Mobile Web Studio. Applications are dynamic, reusable components that capture and deliver information to end users of portals or wireless devices. Applications consist of one or more content elements arranged within a template. These elements can display Web content, XML feeds, database query results, JavaServer Pages, or HTML. The type of elements you use when creating applications depends on the results you wish to achieve.

This section provides some tips and guidelines for developing applications.

- When creating mobile applications, be aware of the memory and display limitations of the mobile device.
- Mobilize only what you need. Identify the key information the mobilized user needs. For example, choose tabular data.

- Use the List/Detail functionality to filter rules to repurpose data for the mobile device.
- Use images only where necessary inside grids.
- Use linked application and server side click-across features to achieve application flow.
- Use the appropriate templates for the various device types you expect the users to have.
- Identify the application source. Unwired Accelerator provides specialized tools for creating elements, such as data, Web, XML, and JSP.

## Workflow for creating applications

This section describes the basic workflow for creating, saving, and approving applications using Mobile Web Studio. These procedures are a general outline, and each task is covered more thoroughly in subsequent chapters. Basic tasks include:

- 1 Create an element – create the element using the Application Builder. You can select from these elements:
  - Web Element
  - XML Element
  - Database Element
  - JSP Element
  - Web Service Element (includes advanced Web Service, using WSDL file objects)
  - HTML Element
  - Document Element
  - File Element
  - Domino Element
  - SAP Element
- 2 Create the application – use the Application Builder wizard to create the application using the element, and customize how the application looks and operates.

Optionally you can preview how the application will appear on various device types, and make modifications to the look and feel for particular devices.

- 3 Save the application – give the application a name, access privilege, and additional post-processing configuration details.
- 4 Approve the application – approve the application to make it available for use.
- 5 Deploy the application – deploy the application for online or offline use. For online use, create a composite application, otherwise deploy or export the application.

Once an application is approved, it is available for use, and you can deploy it to mobile devices, such as BlackBerry and PDA devices, as well as Portal Interface. See Chapter 11, “Deploying Mobile Applications.”

# Developing Applications

This chapter explains the tools you use to create and manage the various types of applications. This chapter also describes the different types of applications you can create and how to create them.

<b>Topic</b>	<b>Page</b>
Overview	13
Mobile Web Studio tools	14
Creating Web applications	18
Creating XML applications	26
Creating database applications	28
Creating JSP applications	33
Using Web service applications	38
Creating HTML-based applications	48
Creating document applications	49
Creating file applications	53
Creating Domino applications	56
Creating SAP applications	70
Saving applications	76
Previewing applications	86
Approving applications	87
Viewing and editing applications	87
Finding and replacing applications	97

## Overview

Mobile Web Studio allows you to build a variety of applications, templates, pages, and catalogs without having to learn any application interfaces. Mobile Web Studio also provides a collaborative environment for developers and administrators by offering versioning, a search filter, a detail view, and roles based access.

Mobile Web Studio enables the automated real-time deployment of individual applications to multiple portals and mobile devices. UA allows you to build once and deploy to multiple mobile operating systems.

This chapter assumes you have:

- Installed Mobile Web Studio using information and procedures in the *Unwired Accelerator Installation Guide*
- Network access
- Internet Explorer installed as your browser
- Configured access to datasources, such as Remedy, Domino, Crystal Reports, and so forth, using information and procedures in the *Unwired Accelerator Administration Guide*

## Mobile Web Studio tools

This section describes the tools and menus you use to develop applications.

### Application Manager

Use the Application Manager to view, edit, and deploy applications. The Application Manager supports two basic types of applications:

- Aggregated application – use the Application Builder to select multiple elements, modify the template selection and other application properties.
- Secure application – content captured from a secure Web site. Use the Application Builder to create and edit secure applications.

Application management functions appear in windows that you launch from the Application Manager tool bar.

#### ❖ **Launching the Application Manager**

- 1 To launch the Application Manager, log in to Mobile Web Studio.
- 2 Select Build | Applications from the Mobile Web Studio left pane.

Application management functions include (depending on the status and application you have selected):

- **New** – launches the Application Builder, where you can create new applications.
- **Edit** – launches the Application Builder where you can edit existing applications and their properties and perform a variety of other functions. See “Viewing and editing applications” on page 87.
- **Preview** – preview any application in a separate browser window.
- **Find** – search for specific applications or view all active approved applications. See “Performing an application search” on page 96.
- **Replace** – search for broken applications and replace them. See “Finding and replacing applications” on page 97.
- **Import/Export** – export applications, composite applications, page groups, pages, agents, and catalogs (to Portal Interface only) to an archive file (with an XML format and extension) in *SYBASE\upload*.  
  
Import files from another source into Mobile Web Studio. You can view XML content, and specify how to handle the import.  
  
See Chapter 12, “Importing and Exporting Portal Objects.”
- **Publish to UDDI** – publish a Web services application to a public registry. See “Using Web service applications” on page 38.

## Application Builder

Use the Application Builder to create or edit applications. To launch the Application Builder, click **New** on the Application Manager toolbar to display the Application Builder.

Once you create your application, the left pane of the Application Builder window displays the **Element list**, which includes the names and position of each element in the application. See “Element list” on page 17.

The right pane of the window displays the **Application Viewer**, which shows the layout of the application you created. See “Application viewer” on page 18.

## Application Builder toolbar

The Application Builder buttons allow you to edit and view application properties and parameters, create and edit labels and templates, and preview application content. Table 3-2 on page 17 describes these buttons.

**Figure 3-1: Application Builder toolbar**



**Table 3-1: Application Builder toolbar options**

Option	Description
Save	<p>Choose from:</p> <ul style="list-style-type: none"> <li>• Save – saves the application under the same name.</li> <li>• Save As – saves a new application or saves an existing application under a new name.</li> <li>• Publish to UDDI – publishes a Web services application to a public registry.</li> </ul> <p>See “Saving applications” on page 76.</p>
Properties	View and edit the properties that were specified when the application was created. Refer to the section in this chapter on each element type.
Params	View and configure application parameters. See “CGI parameters” on page 89.
Labels	Launch the edit window, which allows you to specify application labels. See “Labels” on page 94.
Data Type	Further define the data types for the application. This enables you to add specialized data fields for integers, strings, e-mail, phone, address, calender, task, or contact, and to define new data data types. See “Data types” on page 95 for more information.
Template	<p>Select or create a different template. See Chapter 6, “Building Templates” for more information.</p> <hr/> <p><b>Warning!</b> Creating new templates when you create secure applications could cause a JavaScript error.</p> <hr/>
Preview	<p>Preview the application as follows:</p> <ul style="list-style-type: none"> <li>• Playback – display the HTML preview.</li> <li>• XML – display only data-capable elements.</li> <li>• Edit Screen – display the window that users see if they select Edit Application from their portal environment.</li> </ul> <hr/> <p><b>Note</b> If you choose HTML preview for an XML application, the application does not play back correctly.</p> <hr/> <p>Additionally, you can preview how the application appears on various mobile device types. See “Previewing applications” on page 86.</p>

## Element list

The Element list displays the names and position for each element within an application. Selecting an option from the position drop-down assigns the corresponding element to a template cell.

**Note** A template contains HTML code that defines the position and layout of elements within an application. See Chapter 6, “Building Templates.”

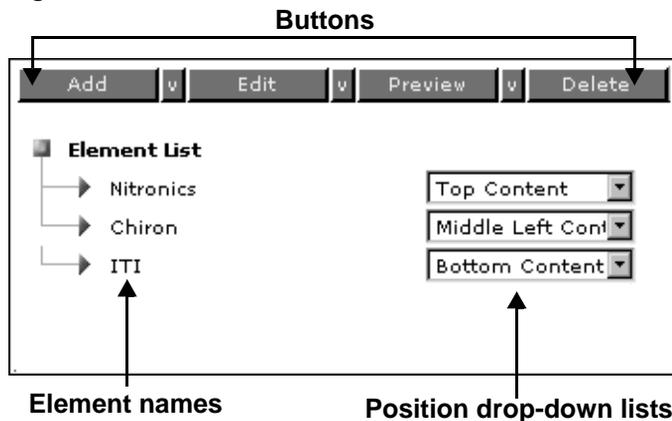
Table 3-2 describes the functionality of the Element list buttons.

**Table 3-2: Element list buttons**

Button	Description
Add	Click the down arrow beside the Add button to add an element of type: Web, XML, Database, JSP, Web Service, HTML, Document, File, Domino, or SAP. Create applications with multiple elements from multiple sources.
Edit	Edit the element definition for XML, database, JSP, and document elements.
Preview	Launch a preview in a separate window: <ul style="list-style-type: none"> <li>HTML – displays the HTML output of the selected element.</li> <li>XML – displays the XML output of the selected element.</li> </ul>
Delete	Remove the selected element.

Figure 3-2 shows the Element List in the Application Builder.

**Figure 3-2: Element list**

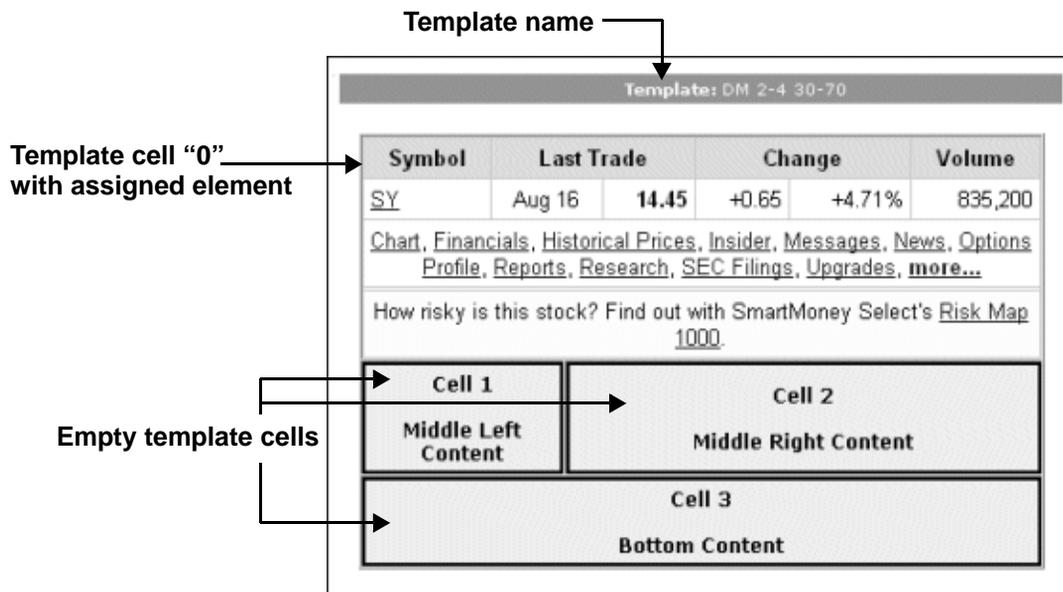


The position drop-down list lets you specify the area of the application—the template cell—where the element should display.

## Application viewer

The application viewer (the right pane of the Application Builder) displays the element layout within the application.

**Figure 3-3: Application viewer**



Template cells are sequentially numbered beginning with “0” (zero). Template cells that do not have elements assigned to them display the cell number and the cell name. The application viewer also displays the name of the template assigned to the application.

## Creating Web applications

Web applications allow you to capture Web content and display it in an application. The captured content is updated dynamically as the source Web content changes.

The Web Element wizard guides you through creating a Web element and provides different capture strategies, which represent the different HTML objects that you can capture from Web pages.

---

**Warning!** The content of many Web sites is protected by the Copyright Law of the United States, related laws contained in Title 17 of the United States Code, and international treaties. Failure to comply with the provisions of such law may result in the imposition of fines and/or damages. Each developer and user is responsible for compliance with applicable law in the use and reproduction of copyrighted content.

---

## Guidelines for choosing capture options

This section includes tips for choosing a capture option.

- Try the default One-click capture first. Enter the URL for the Web content you want to capture and navigate to the desired form.
  - Enter the URL for the Web content you want to capture and navigate to the desired form.
  - The wizard displays various selectable options. A successful capture shows the content you want.
  - If the default One-click capture does not produce the results you want, try a different capture strategy. For example, if you want to capture tabular data on a page, try selecting the Grid format.

## Navigation

The Capture All and Advanced Navigation options are available in Mobile Web Studio only from Application Builder. These options are not selectable from within Portal Interface. Once you select Capture All or Advanced Navigation for an application, an application rendered with these options always plays back in Portal Interface with these options used.

---

**Note** Using Capture All and/or Advanced Navigation can cause a significant performance hit during application playback, as more interaction is required between the UA server and the Web server from which the content is being captured.

---

## Capture all

The Capture All option controls how much navigation information is stored in the CCL. The Capture All option enables the recording of all GET and POST URLs navigated during the capture process in the CCL, therefore, you do not have to go through intermediate URLs before getting to the target page because UA remembers the final URL of the page that you want to capture and play back, along with any intermediate URLs that are generated by HTTP Post requests. These latter URLs typically result from submitting form data. During playback, these URLs are submitted in order and the resulting page content is returned to the client browser.

For some sites, the backend Web servers require the user to navigate through all the GET requests before they can reach a destination page, which can result in not getting to the correct page. In this situation, it may be necessary to include other intermediate HTTP requests (that is, HTTP GET requests) so that the correct client state can be established and the desired page returned. If you suspect this situation is occurring, select the Capture All option to request that during the capture phase, UA should record all significant HTTP requests against the Web server from which the page is being captured.

## Advanced navigation

The Advanced Navigation option passes HTML and JavaScript through the ActiveX component. Problems can occur when you capture and play back content that is partially or completely generated by client-side JavaScript. If you suspect that this is occurring on a site with content you want to capture, select ACX to specify that the capture technology use ActiveX controls to force the rendition of any embedded JavaScript and generate the entire HTML content. When using this option, all JavaScript is removed from the page content returned to the client browser, and the resulting HTML is static.

To capture all GETs and POSTs, and also pass the final HTML or JavaScript through the ActiveX component, select both CapAll and Advanced Navigation.

The Advanced Navigation option allows you to use an embedded browser to perform navigation and capture. This uses ActiveX and Java applet technology, which enables you to perform captures that do not work using the standard navigation/capture approach. ActiveX provides a mechanism for automating Internet Explorer, so that when you navigate to a URL, you can see redirects and other navigational activities that are triggered as a result. This means you can use the browser to perform navigation and monitor the HTTP GETs and POSTs that were generated during the process.

---

**Note** Selecting Advanced Navigation does not automatically result in server-side ActiveX being used at playback. Server-side ActiveX is used only when DOM is also selected.

---

## Capture strategies

The capture strategies listed in the Capture Strategy drop-down of the Web Element wizard represent the different elements on a Web page that Mobile Web Studio can capture. Because of the dynamic nature of Web content, choosing a capture strategy when creating a Web application is sometimes a matter of trial and error. The default One-click format works for most captures. However, if you do not get the expected results, experiment and try a different strategy. For example, if you want to capture tabular data on a page, try using the Grid strategy.

The Document Object Model (DOM) format is useful for pages that are built using a templating mechanism, where content is filled in to various cells of the template. If you want to always get the content of a particular cell from the template, using DOM is effective because the path to the cell of the template always returns the content from the cell, no matter what the content is. See “DOM” on page 24.

These available capture strategies include:

- One-click – allows you to navigate to one page and click through the navigation links on that page to capture the desired content.
- DOM – allows programs and scripts to dynamically access and update the content, structure, and style of documents. See “DOM” on page 24.
- Applet – returns a list of all applets on the page.
- Area – allows you to capture a selection of areas such as table, cell content, or form content, plus similar miscellaneous areas.

- Button – specifies a combination of text and img within <a> and </a> tags. Selecting buttons shows a list of these combinations.
- Grid – defines table content formed into a grid for grid rules processing.
- Image – HTML img element.
- Image Map – defines an image with clickable regions.
- List – specifies sections within the <OL> or <UL> tags. List is used only for Portal Interface portlets, not for mobile device applications.
- Form – specifies that the contained controls take part in a form.
- OP Tag – specifies HTML enclosed inside <optag> sections.
- Visible Text – defines any section of text in any location.
- Page – the default capture mode, represents the page.
- PB Data Win – specifies sections of code inside PBDatawindow HTML output (Power Builder).
- Custom – specifies a custom parser approach that uses a JSP file to extract the desired HTML feature and transform the content.
- Crystal – a Custom implementation that provides special tools for handling Crystal Reports. A sample JSP file is provided.
- Custom XSL – specifies a custom parser approach that uses an XSL file to transform HTML from the target Website to XML using jTidy.

The Capture Option window displays all the instances of the particular HTML object found on a Web page.

The one-click and DOM strategies start another stage in the capture process. When you select either one-click or DOM, the Web content that you can select from using the cursor displays. If you are using one-click, a list of matching elements from the smallest element closest to the point clicked, to the outer most element (the entire page) displays. If you are using the DOM strategy, HTML DOM paths are used to uniquely locate an element on a page. Because HTML paths exist for all items on a page, you have more items to choose from compared with the one-click format, however, this is not always the best method as the DOM strategy requires that items be “fixed” on a page.

Whether one-click, DOM, or Grid is more effective depends on page content, the likelihood of the content changing rapidly, and how it changes. These formats are the most frequently used and work for most situations. Each of these formats is covered in more detail in the next sections.

## One-click

Unwired Accelerator uses a one-click content capture technology that allows you to create applications from Web-accessible sources.

The one-click capture strategy allows you to target a specific area on a Web page from which to capture content using links to navigate to the page that has the information from which you want to create an application. The content capture technology allows you to select only those pieces of information that you want to use in an application.

You cannot always navigate to every possible page and successfully create applications from all Web accessible sources. You may not be able to navigate to some sites or pages within sites. You also may not be able to create an application using the exact piece of information that is viewable to the user on a Web page. Web sites that use JavaScript to generate some or all of an HTML page, and sites that use frame-based solutions are especially challenging. UA provides several options you can try singly or in combination to overcome these challenges. See “Guidelines for choosing capture options” on page 19.

Once you create an application using a Web-accessible source, the application retrieves the current information from the source. If the layout of the information on the original page changes, the underlying technology attempts to locate the correct piece of information that represents the application’s content. However, if the application loses its content or obtains an incorrect piece of information because of changes in the source site, the application must be re-created.

Web elements do not store static HTML, but instead are defined by a code conversion language (CCL) statement that dynamically retrieves the content from its source. The CCL statement consists of:

- The location of the Web page, and
- The feature tag of the captured content. This feature tag is an electronic signature of the content and consists of a feature list that describes the content.

When Mobile Web Studio executes the CCL statement, the capture technology parses the Web page and returns the piece of content that matches the feature tag. If an exact match cannot be determined, the content with the closest matching feature tag is returned. This allows Mobile Web Studio to render the correct Web element even though the content, location, or formatting of the source may have changed.

## DOM

The Document object model used by many browsers specifies how objects such as text, headers, images, and links on a Web page appear. You can use the DOM strategy as an alternative capture strategy that makes use of the DOM when displaying HTML page content. DOM uses server-side ActiveX during playback, and allows you to locate content on a Web page that uses the HTML layout (more explicitly). This Document object model and its published application programming interface (API) is used by UA in conjunction with server-side ActiveX technology to query, modify, and extract page content.

When you select the DOM strategy, you see a sequence of captures from the page. The captures start from the HTML feature closest to the selected item, and roll outwards. For example, clicking on some text embedded in a row of a table, embedded in a page, results in the text being displayed, then the row, then the table, and then the page. The actual “unwinding” depends on the actual HTML structure of the page. This capture strategy is most effective when the content of the site is static.

If required, the DOM strategy can also be combined with client-side ActiveX navigation (advanced navigation) to capture pages that also use JavaScript heavily and as a result are not able to be captured using the normal navigation (non advanced) approach (or for pages that do not capture for other reasons).

It is not necessary to couple it with client-side ActiveX NavACX—that is required only when JavaScript or other rendering approaches in the browser prevent normal navigation and capture. Selecting Advanced Navigation alone does not automatically result in server-side ActiveX being used at playback. Server-side ActiveX is only used when the DOM format or Advanced Navigation are selected.

To use this functionality, when creating a new application, in the Mobile Web Studio – New Web Element window, select DOM from the Capture Strategy drop-down list.

## Grid

The Grid strategy shows all the tabular data on a page, unlike the One-Click strategy, which only shows the grid that encloses the information you click on. The Grid strategy is useful in creating offline mobile applications, where the intent is to find the tabular data on a page.

## Custom

The Custom strategy is a custom parser approach, which uses a JSP file to extract the desired HTML and transform the content for playback. You must create the JSP file in advance. The JSP file is used at capture time to extract possible captures, and at run time to return a specific capture. See “Custom capture options” on page 159 for more information.

## Crystal

The Crystal strategy is a Custom implementation that provides special tools for handling <div> tags, which are used in Crystal Reports instead of <table> tags. Currently Crystal reports and WebIntelligence reports generated by the Business Object XI server are supported. See “Custom capture options” on page 159 for more information.

## Custom XSL

The Custom XSL strategy is also a custom parser approach, but it uses an XSL file to transform HTML from the target Website to XML using jTidy. You must specify the XSL file. You can preview the tidied HTML. See “Custom capture options” on page 159 for more information.

## Creating a new Web application

This procedure shows the basic steps for creating a new Web application. Steps vary depending on the different formats and options you select.

- 1 Log in to Mobile Web Studio and select Build | Applications. Right-click in the detail view and select New Aggregated Application, or click the New button.
- 2 In the Application Builder, click the down arrow next to Add, and select Web Element.
- 3 You see the Web Element Definition window. Enter the URL of the Web content you want to capture in the Location text field, then click Go.
- 4 When the Web page for the location you entered displays, navigate to the page that contains the content you want, then click Next.

- 5 Select the capture strategy from the Capture Strategy drop-down list. See “Capture strategies” on page 21.

---

**Note** When you create an application using one-click capture, and the application content contains numerous links, some of the links may not work properly due to underlying JavaScript or redirect code in the link content.

---

- 6 Select the options to the left of the Back button. The Capture All and Advanced Navigation options are not exclusive; they are separate, independent options. See “Navigation” on page 19.

Click Next.

- 7 Click the left mouse button in a suitable area, which is indicated by the “Select it” shadow box that trails the mouse arrow.

- 8 The capture option window displays the “best guess” capture from the specified area.

Click the light-blue button to the left of the option you want to select. The button changes to a darker blue when you select it. Click Next.

- 9 Enter a name for the element, then click Finish. You return to the Application Builder where the new element displays.

- 10 Click Save. Regardless of the element type, the procedure to save the element and application is the same. See “Saving applications” on page 76.

---

**Note** See Chapter 11, “Deploying Mobile Applications” for information about deploying applications to mobile devices, and see Chapter 9, “Building Pages” for instructions on adding applications to pages.

---

## Creating XML applications

An XML application is XML-based rather than HTML-based. XML is used to describe data, while HTML is mainly used to display data. XML is a common tool for transmitting and manipulating data in Web development. This type of application is generally used by portal administrators who are familiar with XML to Extensible Stylesheet Language Transformations (XSLT).

**❖ Creating a new XML application**

- 1 Log in to Mobile Web Studio and select Build | Applications. Right-click in the detail view and select New Aggregated Application.
- 2 In the Application Builder, click the down arrow next to Add, and select XML Element. You see the XML Element Definition window.
- 3 Enter:
  - XML URL – the base URL of the XML source, such as `http://rss.cnn.com/rss/cnn_topstories.rss`. This creates a new Common Gateway Interface (CGI) parameter. CGI parameters are discussed in “Using application parameters” on page 166.
  - Content XSLT – click Select to search for the XSL template that you want applied to this application’s content.
    - a In the Find XSLT Template window, click Search.
    - b From the Results list, highlight the appropriate XSLT template and click Select.

You return to the XML Element Definition window.
  - UI XSLT – click Select to search for and choose the XSL template (XSLT) that you want applied to the user interface.
    - a In the Find XSLT Template window, click Search.
    - b From the Results list, highlight the appropriate XSLT template and click Select.

You return to the XML Element Definition window.
  - XSD URL – XSD is another XML document that provides descriptive information about the structure of the main XML document. If the XML is generated from a backend system there is typically an auto generated XSD you can use.

If you enter an XSD schema file for the XML document, you can run the XML through the subsequent Split/Define/Filter grid windows.
- 4 When you are satisfied with your entries, click Next.
- 5 In the Define window, define how you want the records in the table to display. See “Defining record layout” on page 133.

Click Next.
- 6 In the Filter window, specify which rows and columns you want to filter out of the display. See “Adding filter rules” on page 133.

Click Next.

- 7 In the Location window, enter a name for the element in Element Name, and click Finish.
- 8 In the Application Builder, save the application as described in “Saving applications” on page 76.

---

**Note** See Chapter 11, “Deploying Mobile Applications” for information about deploying applications to mobile devices, and see Chapter 9, “Building Pages” for instructions on adding applications to pages.

---

## Creating database applications

A database element allows you to query a back-end database and display the results in an application.

### ❖ Creating a database element

- 1 Log in to Mobile Web Studio and select Build | Applications. Right-click in the detail view and select New Aggregated Application.
- 2 When the Application Builder appears, right-click in the left frame and select New Database Element from the Element menu. You can also click the down arrow next to Add and select Database Element.
- 3 In the Database Element Definition window, select one of either:
  - Connection Cache – this option is the default. In ConnCache Name, select the name of the connection cache you want to use from the drop-down list, such as `sampledb`.

Optionally, you can click Browse to see metadata for the connection cache name you selected from the drop-down list. In Relation Names, select a table, then view column information.

---

**Note** Using a connection pool is more efficient than using the JDBC URL option, which creates a new connection to the database every time the application executes.

---

- JDBC URL – customizes the login or other connection information for specific users. Enter:

- Username – enter the user name used to access the database. The default user name for Adaptive Server Anywhere is dba.
- Password – enter the password used to access the database. The default password for the Adaptive Server Anywhere database is SQL.
- JDBC Connect URL – enter the connection string to access the database; for example:

```
jdbc:sybase:Tds:localhost:port/portaldatabase
```

- *localhost* – the name of the installed database server
- *port* – the port used to connect to the specified database
- *portaldatabase* – the name of the database to which you want to connect

---

**Note** If the database server is installed on the same machine as UA, you do not have to change the “localhost” value.

---

- JDBC Driver – enter the JDBC driver used to access the database; for example:  

```
com.sybase.jdbc3.jdbc.SybDriver
```
- SQL Query String – enter the SQL statement to retrieve the requested data from the database. Click Preview to see the query results. See “@OP parameters” on page 92 for information about using @OP parameters in a database query.
- UI XSLT – (optional) enter the complete URL of the XSLT to be applied to the content returned by the database or click Select to choose from existing XSL templates.

Click Next.

- 4 In the Define window, define the record layout and specify a record header. See “Defining record layout” on page 133. Click Next.
- 5 In the Filter window, specify filter rules to select only the relevant data from the element. See “Adding filter rules” on page 133. Click Next.
- 6 This window displays only if you have defined the record layout. The Configure Parameters window displays the element’s CGI parameters. See “CGI parameters” on page 89 and “Using the database table adapter” on page 202. Click Next.

- 7 In the New Web Element window, enter a name for the database application in the Element Name field. Click Finish.
- 8 In Application Builder, save the application as described in “Saving applications” on page 76.

---

**Note** See Chapter 11, “Deploying Mobile Applications” for information about deploying applications to mobile devices, and see Chapter 9, “Building Pages,” for instructions on adding applications to pages.

---

## Database connection cache options

When you define a database element, you can either enter a JNDI resource name, or a JDBC connection URL for the data source.

---

**Note** Using a connection pool is more efficient than using the JDBC URL option, which creates a new connection to the database every time the application executes.

---

## Using a default JNDI data source resource

When you create a database element, you can use one of the default JNDI resources included with Unwired Accelerator, or you can create a new resource. For information about creating a new JNDI resource, see the *Unwired Accelerator Administration Guide*.

These default JNDI data source resources are included with Unwired Accelerator:

- `java:/com/env/jdbc/portaldb` – connects to the portal database.
- `java:/com/env/jdbc/sampledb` – connects to the sample database.
- `java:/com/env/jdbc/agdb` – connects to the M-Business Anywhere database.
- `java:/com/env/jdbc/msales` – connects to the mobile sales demo database.
- `java:/com/env/jdbc/mpharma` – connects to the mobile pharmacy demo database.

- `java:/com/env/jdbc/uaml` – connects to the MobiLink database.

---

**Note** These data sources are only examples. You must create your own database with your own data source resources.

---

In the Mobile Web Studio – New Application Database Element Definition window, select the JNDI resource name from the drop-down list; for example, “portaldb” for portal database, or “sampledb” for the sample database.

## JDBC connection URL for the data source

When you create a database element, you can use a JDBC URL to connect to the data source. This method is not as efficient as a connection pool, but is useful for testing, training, and demonstrating database applications.

**ASA** To set up a JDBC connection to ASA, supply the JDBC URL and JDBC driver values in this format:

```
jdbc:sybase:Tds:localhost:4747?ServiceName=sampledb
com.sybase.jdbc3.jdbc.SybDriver
```

This method is demonstrated in the *Mobile Application Development Tutorial*.

**SQL Server** To setup the SQL Server cache:

- 1 Copy the *msbase.jar*, *mssqlserver.jar*, and *msutils.jar* to both:
  - `%SYBASE%\tomcat\common\lib`
  - `%SYBASE%\tomcat\webapps\onepage\WEB-INF\lib`
- 2 Edit `%SYBASE%\tomcat\conf\server.xml` and add the following inside the `<GlobalNamingResources>` section:
 

```
<Resource name="jdbc/sqlserver" auth="Container"
type="javax.sql.DataSource"

driverClassName="com.microsoft.jdbc.sqlserver.SQLServerDriver"
url="jdbc:microsoft:sqlserver://win2k3:1433"
username="sa"
password=""
maxActive="20"
maxIdle="10"
maxWait="20000"/>
```

- 3 Edit the *SYBASE\Tomcat\webapps\onepage\META-INF\context.xml* file and add the following inside the `<Context>` section:

```
<Resource name="jdbc/sqlserver" auth="Container"
type="javax.sql.DataSource"

driverClassName="com.microsoft.jdbc.sqlserver.SQLServerDriver"
url="jdbc:microsoft:sqlserver://win2k3:1433"
username="sa"
password=""
maxActive="20"
maxIdle="10"
maxWait="20000"
removeAbandoned="false"
removeAbandonedTimeout="60"
logAbandoned="false"/>
```

- 4 Edit the *SYBASE\Tomcat\webapps\onepage\WEB-INF\web.xml* file and add the following inside the `<web-app>` section:

```
<resource-ref>

<description>SQLServer Sample DB</description>
<res-ref-name>jdbc/sqlserver</res-ref-name>
<res-type>javax.sql.DataSource</res-type>
<res-auth>Container</res-auth>
</resource-ref>
```

That should show the sqlserver cache as a drop-down in the database capture wizard. You will need to modify the hostname, port, username, password in the above entries appropriately.

#### Oracle

To set up Oracle, follow the same steps, but substitute the Oracle JDBC URL and driver values in this format:

```
url="jdbc:oracle:thin:@<host>:1521:<OID>"
driverClassName="oracle.jdbc.driver.OracleDriver"
```

Oracle provides multiple JDBC drivers and above is the thin version. If you use other types your URL will change too. Also, contact the Oracle DBA for the hostname, OID, username, and password values to use.

## Creating JSP applications

UA allows you to capture JavaServer Pages (JSP) Web applications and display them in an application.

---

**Note** JSP applications must be single-element applications even though you build JSP applications by selecting Aggregated Application.

---

### ❖ Creating a JSP element

- 1 Log in to Mobile Web Studio and select Build | Applications. Right-click in the detail view and select New Aggregated Application.
- 2 In the Application Builder, click the down arrow to the right of Add and select JSP Element. You see the JSP Element Definition window.
- 3 Select which option to use for the JSP element:
  - Use Web Application – if the reference is a Web application archive (WAR) file that is on the application server. Go to “Using a Web application” on page 33 to complete the procedure.
  - Use JSP Template – if the JSP element does not reference a WAR file. Go to “Using a JSP template” on page 36 to complete the procedure.
  - Use Remote URL – if the JSP is not part of a Web application in the same Web container as the portal. Go to “Using a remote URL” on page 37 to complete the procedure.

## Using a Web application

- 1 Before you start, locate the Web applicaiton. If the Web application is part of a WAR file, unzip the WAR file within the ...\*onepage* directory structure.
- 2 Select Use Web Application.
- 3 WAR file - enter the path to the Web application.
- 4 Enter the Initial Resource—the initial JSP page to display.
- 5 Select from these options:
  - Web App Qualified URLs – enter an URL with parameters to call a Web application from the JSP application. See “Using a remote URL” on page 37.

- Enable Grid Rules – select to enable the use of grid rules. See “Using grid rules” on page 135.

---

**Note** Grid rules do not apply to forms, or tables/grids that include form UI. To use grid rules with a form, you need to use JSP code to return the table after processing the element.

---

- Single Sign-On Required – select if single sign-on is required. See “JSP application single-sign-on” on page 36. When you select this option, three more options become available:
  - JSP Include – use the `javax.servlet.RequestDispatcher.include(...)` method to enable the target JSP to retrieve the JSP’s contents. This method returns the target JSP’s output to the portal and allows you to manipulate, filter, and integrate that output with other elements in an application with multiple elements—an aggregate application. Aggregate applications use either an `<IFRAME>` or `<DIV>` tag style.

The target JSP gets a `contextPath` in the `ServletRequest` object that matches the application name under which the JSP is installed.

JSPs using the `.include()` method have a `<BASE HREF=“targetJSP path”>` element inserted into the HTML `<HEAD>` element. This ensures that relative HREFs or IMG links generated from that JSP are loaded correctly.

- JSP Forward – use the `javax.servlet.RequestDispatcher.forward(...)` method to enable the target JSP to retrieve the JSP’s contents. This method does not return the target JSP’s output back to the portal.

This application style must run in an `<IFRAME>` tag and cannot be included in an aggregate application (application with multiple elements). The `forward` method ensures that the target JSP receives the `contextPath` with which it was installed. There is no opportunity for the portal to insert the `<BASE HREF=...>` element into the returned HTML, so if the `<BASE HREF=...>` information is not provided by the application, relative HREFs or IMG links may not work.

- 302 Redirect – this option generates a HTTP Status 302 redirect back to the browser. When the browser receives a 302 redirect status, it executes a get method to the URL contained in that 302 message. This application style must run in an <IFRAME> tag, otherwise the application overlays the entire screen.

One benefit of this application style is that the declarative J2EE security constraints that are part of the target Web application are enforced by the container. The J2EE standard does not require the container to apply declarative security constraints for RequestDispatcher include or forward.

---

**Note** Web applications in J2EE containers can have security descriptors that are mentioned in *web.xml*. This section of the XML puts a constraint on all the resources of the Web application. Access is granted only to users who are in the manager role. Users who try to access the Web application from the browser (or other client program) must be authenticated. If the type of authentication required is basic authentication, the browser prompts the user for a user name and password.

If the Web application is added as an application to a page, users need not re authenticate with the application. The portal uses container authentication to authenticate users when they log in to the portal. The same session context authenticates the user with the Web application used to authenticate the user with an application. If the user roles are the same across the Web application, the user is authenticated into the Web application and the contents display in the application.

See the *Enterprise Security Administration Guide* for details.

---

- 6 Input Parameters – enter the input parameter names as defined by the JSP code.
  - a You must use a name, value pattern like this:
 

```
name1=value1&name2=value2&name3=value3
```
  - b The name must be exactly the same as your JSP code's input parameter's name. For example, if the JSP page's code is:

```
EPLogin:<input type="text" name="epuser">
EPPswd:<input type="password" name="epassword">
HostName:<input type="text" name="hostname">
AssetDN:<input type="text" name="asset">
PAI URL:<input type="text" name="paiURL">
```

The input parameters must then be:

```
epuser=tester&epassword=test123&hostname=drillbit2k&asset=a1=pubs,  
dc=sybase,dc=com&paiURL=jdbc:sybase:Tds:sapphire:5000/pubs2
```

## JSP application single-sign-on

When you create a JSP application from a WAR file in Unwired Accelerator, there is a check box labeled Single Sign On (SSO). If you do not select this option, the implications are:

- The JSP application is invoked without any credentials passed to it.
- If the JSP application has the directive `<% page session="true" %>` in the JSP page, a new HTTP session is created whenever the JSP application is invoked.
- A cookie is sent back to the portal, but the cookie is not remembered. Therefore, any state stored within the HTTP session is lost.
- Since the HTTP session does not go away quickly—the default is 30 minutes in EAServer—repeated use of the JSP application can create a severe memory drain to the system.

## Using a JSP template

- 1 Use JSP Template – select if the JSP element does not reference a WAR file.
  - JSP Template – select a JSP template using the Find Template window.
- 2 Input Parameters – enter the input parameter names as defined by the JSP code.
  - a You must use a name, value pattern like this:

```
name1=value1&name2=value2&name3=value3
```
  - b The name must be exactly the same as your JSP code's input parameter's name. For example, if the JSP page's code is:

```
EPLogin:<input type="text" name="epuser">  
EPPswd:<input type="password" name="epassword">  
HostName:<input type="text" name="hostname">  
AssetDN:<input type="text" name="asset">  
PAI URL:<input type="text" name="paiURL">
```

The input parameters must then be:

```
epuser=tester&epassword=test123&hostname=drillbit2k&asset=a1=pubs,
dc=sybase,dc=com&paiURL=jdbc:sybase:Tds:sapphire:5000/pubs2
```

## Using a remote URL

- 1 Use Remote URL – select this option if the JSP is not part of a Web application in the same Web container as the portal.

---

**Note** Since that JSP is not deployed in the portal Web container, you cannot fill in the fields associated with the Use Web Application option. You also cannot select Single Sign-on.

---

- Remote URL – specify the fully-qualified URL to the remote JSP, starting with HTTP(S).
- 2 Input Parameters – enter the input parameter names as defined by the JSP code.
    - a You must use a name, value pattern like this:

```
name1=value1&name2=value2&name3=value3
```

- b The name must be exactly the same as your JSP code's input parameter's name. For example, if the JSP page's code is:

```
<BR>EPLogin:<input type="text" name="epuser">
<BR>EPPswd:<input type="password" name="epassword">
<BR>HostName:<input type="text" name="hostname">
<BR>AssetDN:<input type="text" name="asset">
<BR>PAI URL:<input type="text" name="paiURL">
```

The input parameters must then be:

```
epuser=tester&epassword=test123&hostname=drillbit2k&asset=a1=pubs,
dc=sybase,dc=com&paiURL=jdbc:sybase:Tds:sapphire:5000/pubs2
```

- 3 When you are satisfied with your entries, click Next.

---

**Note** The Preview button does not work for JSP applications.

---

- 4 Enter an element name, then click Finish. You return to the Application Builder.

Regardless of the element type, the procedure to save the element and application is the same. See “Saving applications” on page 76 for instructions.

## Using Web service applications

A Web service is software that is available over the Internet and uses a standardized XML messaging system to invoke the service and supply the response. Web services are a useful way to provide data to a variety of consumers; for example, traffic reports, stock quotes, and travel services. This section provides basic introductory material about Web services and describes how to implement Web services in your portal.

---

**Note** This book does not provide the detailed information necessary to fully understand and use Web services and Web service protocols. You can find detailed information at <http://searchwebservices.techtarget.com/> and at <http://www.w3.org/2002/ws/>.

---

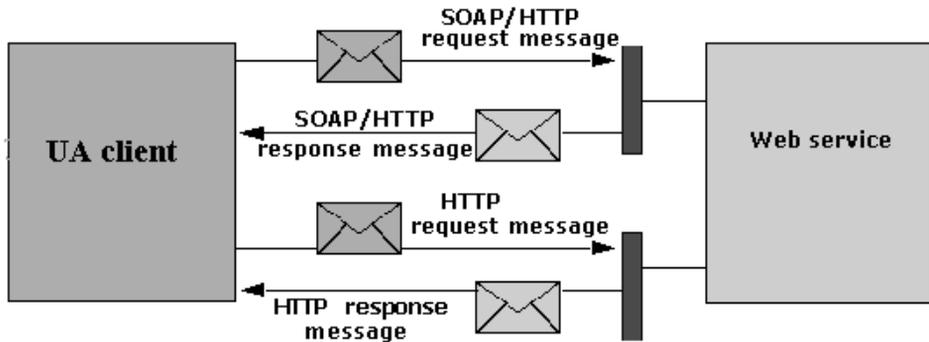
## Examples

A list of publicly available Web services can be found at <http://www.xmethods.com/>, including a stock-quote service, a traffic-report service, and a weather service.

Each service is independent, reusable, and performs a distinct task. These services can be linked together across or within enterprises using XML. You can stack services like toy blocks, which allows you to build applications from Web services quickly, borrowing blocks from UDDI registries when you need them.

## Using Unwired Accelerator Web services

Web service support is implemented using Apache SOAP 2.3.1, and has been tested with the client side of the same version. UA has also successfully performed compatibility tests with Apache Axis.

**Figure 3-4: Web services illustration**

Unwired Accelerator allows you to create Web services elements and incorporate those elements into applications. An example of a Web service might be weather conditions based on postal code or stock quote results based on company symbols.

You create Web service elements that use standard WSDL files or service keys (globally unique identifiers that refer to services published in a UDDI Business Registry), then add the elements to applications that execute local or remote Web services to extract and display application content.

Unwired Accelerator also lets you create Web services applications and publish the applications as Web services to either of these UDDI registries—TrySybase, and Microsoft test.

## Requirements

To use Web services with Unwired Accelerator:

- You must use the version of *global.properties.xml* installed with Unwired Accelerator, which includes the `WebServiceRoot` property in the UWP property group.
- Unwired Accelerator comes preconfigured to use Web services. You can verify or change Web services configuration settings in the *webservices.properties.xml* file.

Both files are located in:

Tomcat `%SYBASE%\tomcat\webapps\onepage\config`

See the *Unwired Accelerator Administration Guide* for information about the configuration settings.

- If you get a Secure Sockets Layer (SSL) certificate error while trying to publish a Web service to a registry using HTTPS, use one of these options to resolve this:
  - Install a new version of Java (for example, JDK 1.5.0.x) and run Unwired Accelerator with that version. The root certificate on the registry site should already be installed in your default keystore file—*cacerts*.
  - Use your browser to download a root certificate (issued from Verisign), then use `keytool` to import this certificate to your default keystore (*cacerts*), or a newly created keystore. If you import the certificate to a new keystore, you must also point the client keystore property to the new keystore:

```
[System.setProperty("javax.net.ssl.keyStore", "...")]
```

- Register accounts with the UDDI public registries, then publish your Web service.

---

**Note** The TrySybase registry does not provide a site to register an account; use `guest` (user name) and `guest` (password) as the account login to publish to the Sybase registry.

---

## Complex Web service types

WSDL documents can use the `<types>` element as a container for data type definitions that are relevant for exchanged Web services messages. For maximum interoperability and platform neutrality, WSDL uses the XML Schema Definition (XSD) language for data type definitions. For example:

```
<definitions .... >
  <types>
    <xsd:schema .... />*
  </types>
</definitions>
```

Unwired Accelerator can parse all WSDL files that use the `complexType` data type and other compound data types using the format that complies with WC3 XML Schema specification.

---

**Note** Parsing WSDL takes longer the first time the WSDL is accessed.

---

You can define these data types within the `<types>` element to represent all or part of input or output messages. When you create Web services applications and elements, you must provide inputs to these fields to generate input messages for the Web service's invocation.

---

**Note** Verify that the Java compiler command (`javac`) is available from the UA server process (Tomcat).

---

## Defining the login method

Some Web services require that an authenticated session header be provided with invocation of all Web service methods. These Web services provide a login method that must be called first to get a session handle, which then must be used with subsequent method requests.

To define a login method, select the Define Login Method option in the Web Service Element Definition window, then choose the login method from the drop-down list, and enter the user name and password to use with the selected login method.

## Enabling basic authentication

Some Web services definitions include the Action SOAP URL that requires basic authentication so that the HTTP request accessing the SOAP URL must provide a user name and password before the Web server accepts the Web service method request.

UA provides an option you can select when you create a Web service application that allows you to enter a user name and password for SOAP URLs that require basic authentication. Simply select Enable Basic Authentication in the Web Service Element Definition window, and enter the user name and password in the fields that display.

## Creating Web service applications

When you create a Web service application, you must have access to where that service resides on the Internet. UA allows you to identify Web services that you want to use in an application either of two ways—using a service key or using WSDL.

- Service key – is a globally unique identifier that refers to a service published in a UDDI Business Registry.
- WSDL – a normalized description of disparate software that allows users to access the software independent of protocol or location.

❖ **Creating a Web service element**

Use this procedure to create a Mobile Web Studio element that uses the standard Web Services Description Language (WSDL).

- 1 Specify the location in which the Web service files will be generated and from where they are exposed by opening the *global.properties.xml* file located in:

Tomcat

*%SYBASE%\tomcat\webapps\onepage\config*

Locate the `WebServiceRoot` property in the UWP property group and modify the default value if necessary. There are two ways to specify the location of generated Web service files:

- Enter the absolute location; for example, if you enter *d:\work\service\ws*, the system adds files directly to the specified location. Ensure that the location can be accessed from a URL, such as *http://localhost:8085/onepage/ws*.
  - Enter the relative location. For example, if you enter *service\ws*, the system adds files to the location relative to the current Web application. Ensure the location can be accessed from a URL, such as *http://localhost:8085/onepage/service/ws*.
- 2 Select **Build | Applications**. Right-click in the detail view and select **New Aggregated Application**.
  - 3 In the **Application Builder**, click the down arrow beside the **Add** button, and select **Web Service Element**. You see the **Web Service Element Definition** window.
  - 4 In the **Web Service Element Definition** window, enter:
    - **Service Type** – select from:
      - **WSDL**
      - **UDDI Key** – enter the UDDI key.
    - **Use Standardization XSLT** – indicates whether to use an XSLT file.
    - **Standardization XSLT** – search for the XSLT file to use.

- UDDI Registry – select the UDDI Registry from the drop-down list.
- WSDL URL – enter a Web service WSDL location. For example:

```
http://www.xmethods.net/wsdl/query.wsdl
```

---

**Note** This is a sample Web service that provides a SOAP remote procedure call (RPC) interface to the xmethods.net Web site for query operations. xmethods.net provides a listing of publicly available Web services.

---

- Method – when you click Get Method, the system retrieves the WSDL XML. The list of defined services is extracted from the WSDL file and populates the Method drop-down list.

Select a method from the drop-down list and click Input to see a list of the input parameters of the selected Web service method in the bottom pane.

If you select a method where the Web service requires session information in the SOAP header, you must also select Define Login Method.

- Enable Grid Rules – shows the output in grid format.

---

**Note** If you want the Web service application to be viewable on a mobile device, you must check the Enable Grid Rules box, define content XSLT, and provide input parameters before going to the next step.

---

- Define Login Method – select this option if you select a method, where the Web services require session information in the SOAP header.

When you select this option, two new fields appear:

- Login Method – select from the list of methods.
- Login Param – enter the user name and password to use with the selected login method.
- Enable Basic Authentication – allows you to enter a user name and password for basic authentication when working with authentication protected URLs. When you select this option, two new fields appear:
  - User Name
  - Password

The user name and password you enter are used internally by the Web services element for basic authentication of HTTP requests of SOAP calls.

- Content XSLT – search for and select a content XML-to-XSLT transformation file, or create content XSLT.

---

**Note** In UA Web services, UA uses cached information for the templates. If you make changes to the cached Content XSLT template, the changes do not take effect until the next Tomcat server restart.

---

- UI XSLT – search for and select a UI XML-to-XSLT transformation file.

Click Next.

- 5 If you selected Enable Grid Rules, you see the Define, and Filter windows. Provide the input for each window, and click Next.
- 6 In the Finish window, in Element Name, enter an element name, and click Finish.
- 7 Save the application. Regardless of the element type, save the element and application as described in “Saving applications” on page 76.

❖ **Creating a Web service element using a service key**

In this procedure, you use a globally unique identifier (service key) to reference a Web service in the UDDI Registry. Service keys are hexadecimal strings generated when a Web service is registered with a UDDI registry.

For example, when you search the Web service provider at [www.bindingpoint.com](http://www.bindingpoint.com) for UDDI Web services registries, you see the service key in addition to other information about the service.

Search Rated UDDI

Description: Search UDDI using rated and validated information only. This service looks at all UDDI entries and allocates a weighting to each record depending on validity and completeness. You can then search UDDI registry and retrieve only valid records.

Service Key: 5ad60ec4-846a-4517-b391-724bcdbdbe26  
Provider: Lucin  
Date Added: Aug 12, 2002  
Average response time: 0.32 seconds  
Price (per transaction): Free

- 1 Select Build | Applications. Right-click in the detail view and select New Aggregated Application.
- 2 When the Application Builder displays, click the down arrow beside the Add button, and select Web Service Element.

You see the Web Service Element Definition window.

- 3 In the Web Services Element Definition window:
  - a Service Type – select UDDI Key.
  - b UDDI Registry – select the UDDI registry that specified where to look for the key definition.
  - c UDDI Key – enter the Web services key. For example:

8BD03750-BBE2-11D6-B8B8-000629DC0A7B

You obtain this key from the UDDI Registry where the associated Web service is published.

- d Click GetMethod. The system retrieves the WSDL XML. From the WSDL, the list of defined services is extracted and populates the Method drop-down list.
  - e Select a method from the Method drop-down list.
  - f Click Preview. The system extracts the input parameters of the selected Web service method and lists them in the preview pane.
- 4 Click Next.
- 5 In the Finish window, enter an element name in Element Name, and click Finish.
- 6 Save the application.

Regardless of the element type, the procedure to save the element and application is the same. See “Saving applications” on page 76 for instructions.

❖ **Publishing an application to a UDDI registry as a Web service**

- 1 Log in to Mobile Web Studio and select Build | Applications.
- 2 In the Application Manager, select Approved, then from the detail view, select the application that you want to publish as a Web service.
- 3 On the Application Manager toolbar, click Publish to UDDI. You see the Publish to UDDI window.

- 4 Complete the fields in the Publish to UDDI window:
  - Service Name – enter the name by which this application is to be published; for example, “Welcome.”
  - Provider Name – select or define a new Web service provider under whose name the application Web service is to be published. To update or create a new provider, click Details.

When you define a new provider, you see the window where you can enter or update the information about the provider.

- a Enter or update information on the Provider Details window:
    - Provider Name – the provider of the Web service.
    - Person Name – the contact person.
    - E-mail Address – the provider’s e-mail address.
    - Provider Description – optional description of the provider.
  - b Click OK.
- 5 Complete the remaining fields on the Publish to UDDI window:
    - Category – select a category from the drop-down list for the Web service.
    - UDDI Registry – select from a list of preconfigured UDDI Registries. See “Creating new UDDI registry entries” on page 49 for instructions on adding new entries to the registry list.
    - User name – enter the user name used to access the UDDI registry.
    - Password – enter the password used to access the UDDI registry.
    - Respond with HTML – receive a response in HTML rather than an XML format.
    - Remember Settings – have the system remember the settings on this page for this session.
  - 6 Click OK.

The system generates the Web service definition interface, the implementation WSDL files, and the deployment file in the directory specified for the WebServiceRoot property entry in *global.properties.xml*. You may have to wait for several seconds before the system responds.

## Creating new UDDI registry entries

---

**Note** The Web service implementation in Unwired Accelerator uses Apache SOAP 2.3 and UDDI4J version 2.0.

---

When you create a new Web services element and select a UDDI registry from which to select a service, or enter information to publish a Web service to a UDDI registry, you see a drop-down list. The entries on these lists are extracted from the *webservices.properties.xml* file in `%SYBASE%\tomcat\webapps\onepage\config` if you are using Tomcat. An example of this file follows.

- To add a new UDDI registry from which to extract Web services to use in new Web service elements, use a text editor to add a new `<UDDIRegistry>` section to *webservices.properties.xml*.
- To update the default publishing information with which you want to publish the Web service, use a text editor to update the value for the corresponding `<Property>` to the `<DefaultPublishInfo>` section of *webservices.properties.xml*.

```
<WebServiceProps>
<UDDIRegistry name="IBMTTestRegistry">
  <Property name="displayname" value="IBM Test Registry" />
  <Property name="id" value="0" />
  <Property
name="inquiryURL" value="http://uddi.ibm.com/testregistry/inquiryapi" />
  <Property
name="publishURL" value="https://uddi.ibm.com/testregistry/publishapi" />
</UDDIRegistry>
<UDDIRegistry name="IBM">
  <Property name="displayname" value="IBM Public Registry" />
  <Property name="id" value="1" />
  <Property name="inquiryURL" value="http://uddi.ibm.com/ubr/inquiryapi" />
<Property name="publishURL" value="https://uddi.ibm.com/ubr/publishapi" />
</UDDIRegistry>
<UDDIRegistry name="MicrosoftTestRegistry">
  <Property name="displayname" value="Microsoft Test Registry" />
  <Property name="id" value="2" />
  <Property
name="inquiryURL" value="http://test.uddi.microsoft.com/inquire"/>
  <Property
name="publishURL" value="https://test.uddi.microsoft.com/publish" />
</UDDIRegistry>
<UDDIRegistry name="Sybase">
```

```
<Property name="displayname" value="TrySybase" />
<Property name="id" value="3" />
<Property
name="inquiryURL" value="http://uddi.trysybase.com/juddi/inquiry" />
<Property name="publishURL" value="http://uddi.trysybase.com/juddi/publish" />
</UDDIRegistry>
<DefaultPublishInfo name="DEFAULT">
  <Property name="provider_name" value="Sybase EP Services" />
  <Property name="person_name" value="Sybase" />
  <Property name="email_address" value="sybase@sybase.com" />
  <Property name="provider_desc" value="Sybase EP Web Services" />
  <Property name="username" value="sybase00" />
  <Property name="password" value="sybase" />
  <Property name="respond_with_html" value="false" />
  <Property name="remember_settings" value="false" />
</DefaultPublishInfo>
<Properties name="Properties">
  <Property name="ArraySize1" value="2" />
  <Property name="ArraySize2" value="3" />
  <Property name="WSCache" value="true" description="(true/false) true to
turn on caching for processed complexTypes and definitions of wsdl" />
  <Property name="DefaultExpireLength" value="259200000"
description="default expire length for cached WS element, eg. 3 days" />
</Properties>
</WebServiceProps>
```

## Creating HTML-based applications

HTML elements are pieces of HTML code that you can arrange within an application with other element types.

### ❖ Creating an HTML element

- 1 Log in to Mobile Web Studio, select Build | Applications, right-click in the detail view, and select New Aggregated Application.
- 2 In the Application Builder, click the down arrow beside the Add button and select HTML Element.
- 3 In the HTML Element Definition window, complete these options:
  - Select Enable Grid Rules to enable grid rules. See “Using grid rules” on page 135.
  - HTML – enter the HTML code; for example:

```
<IFRAME STYLE="width:100%; height:552;"  
SRC="http://is.marketwatch.com/?pcor=news">  
</IFRAME>
```

When you are satisfied with your entries, click Next.

- 4 In the next three windows—Split, Define, and Filter, click Next.
- 5 In the New Web Element window, in Element Name, enter a name for your application, and click Finish.
- 6 In the Application Builder, click Save. See “Saving applications” on page 76.

## Creating document applications

The document element allows you to load an application document (Microsoft Word, Microsoft Excel, PDF, and so on) into an application.

When you create document elements:

- Applications built with the document element may require that the Web browser plug-in for the application in which the document was authored be installed on the machine from which the user is browsing before they can view the application’s content. For example, to view an application with PDF content, the Adobe Acrobat Reader plug-in for your Web browser must be installed on the machine from which you are browsing. Refer to your Web browser’s Internet site for more information.
- If you upload a document with the same name as a previously uploaded document, the second document overwrites the first, unless you create a subdirectory below the upload directory and upload the second document to the subdirectory.
- Document elements must be saved with the IFRAME attribute.
- You can use only the default template except for the content types text/HTML and text/plain.
- You cannot import or export a document application across servers.
- You can create a document application with a new content type that is not included in the drop-down menu if there is an application that supports the plug-in.

- You must set the content type of document element applications to the correct type.
- Applications that use the document element (except for the content types text/HTML and text/plain) can contain only one element.

❖ **Creating a document element**

- 1 Log in to Mobile Web Studio, select Build | Applications, right-click in the detail view, and select New Aggregated Application.
- 2 In the Application Builder, click the down arrow beside the Add button and select Document Element.
- 3 In the Document Element Definition window, complete these fields:
  - Document to Upload – enter the full path to the document you want to upload, or click Browse to search for and select the document.
  - Destination Location – accept the default folder name or append the default name with an additional folder name. Enter only the folder name; that is, do not enter the complete path. This is the location to which the document is uploaded.

---

**Note** You can configure the default upload folder name in *global.properties.xml*. See the *Unwired Accelerator Administration Guide*. The parent location of the document root folder must have “write” permissions.

---

Mobile Web Studio creates a document root level directory on the drive of your UA installation and uses the current user’s login name as the default folder. For example, if the default for the Destination Location displays “Smith”, and the UA installation drive is D:\, the uploaded document is saved to *D:\tmp\PortletDocs\Smith*.

To create a different folder using this structure, enter another file name after the name that displays in the Destination Location field. For example, if the field displays “Smith” and you enter “\Smith,” the file is uploaded to *D:\tmp\PortletDocs\Smith\Smith*.

- Document Name – this is automatically filled in when you click Upload.
- Content Type – select the content type of the document, for example, “application/msword.

If you do not see the content type you want, select Add New from the drop-down list to add a new content type.

- New Content Type – displays only when you select Add New from the Content Type drop-down list. Enter the new type in this format:

[application | image | text]/plug-in name

---

**Note** There must be an application and the plug-in installed for the content type specified. See “Adding new content types” on page 51.

---

- 4 Click Upload. When a message displays confirming that the upload was successful, click OK.

---

**Note** The length of time before the confirmation message displays depends on the document’s size.

---

- 5 Click Next.
- 6 In the Finish window, in Element Name, enter a name for the element and click Finish.
- 7 In the Application Builder, click Save. See “Saving applications” on page 76.

## Adding new content types

When you create a document element and add a new content type, you must enter the name and type of the plug-in that allows you to view the document in a Web browser. The plug-in must also be registered with the Web browser. Registering plug-ins or helper applications is different for Internet Explorer and Netscape.

### ❖ Registering new content types with Internet Explorer

- 1 Go to <http://support.microsoft.com>.
- 2 Enter 306790 in the Search the Knowledge Base field, and click the Go arrow.

When a link to Microsoft Knowledge Base Article - 306790 “Description of Internet Explorer Support for Netscape-Style Plug-ins,” click the link.

This document describes how to register content types with Internet Explorer and how to specify the application associated with the document you want to view in the Web browser.

❖ **Adding a new plug-in for Netscape**

- 1 In Netscape, select Edit | Preferences.
- 2 When the Preferences window displays, select Navigator | Applications if you are using Netscape 4.76. If you are using Netscape 7.0, select Navigator | Helper Applications.
- 3 To add a new helper application (plug-in) to the list, click New Type.
- 4 When the New Type window displays:
  - a Enter a description of the type of files the application works with; for instance, “movies” or “sound files.”
  - b Enter the file extension (suffix) used by this file type. If this type of file uses more than one suffix, enter them all, separated by spaces. Examples: text documents often have the suffix “txt”; some image files have the suffix “gif”; and HTML documents have the suffix “htm” or “html.”
  - c Enter the MIME type. MIME types consist of two words separated by a slash (/). For instance, a TIFF image file’s MIME type is “image/tiff”. If you do not know the MIME type, leave the field blank.
  - d Click Browse and select an application that handles files of this type.
  - e Click OK.
- 5 You can specify how Navigator should handle downloaded files. Select a file type from the list and click Edit.
  - a Choose a “Handled By” option to open downloaded files in a Navigator window, save them on your hard disk, or open them in an application that you specify. Click Browse to specify the application.
  - b Check “Ask me before opening downloaded files of this type” if you want to be notified before downloading files that are handled by applications other than Navigator.
  - c Click OK.

## Creating file applications

Unwired Accelerator lets you create applications that are based on files. The file must contain data arranged in a format that can be mapped to grid data (see “Using grid rules” on page 135 for information about grid rules, for example:

- Delimited columns – a file that has data separated by commas, tabs, semi-colons, colons, or spaces, such as a file with a *.csv* extension, or an Excel spreadsheet.
- Fixed-size columns – a file with data arranged in columns of a fixed size with a *.txt* extension.

Using either file type, you can create an application element that uses the grid-rule functionality of Unwired Accelerator.

---

**Note** UA uses the uploaded copy of the file, not the copy that is local to the browser. In other words, changes to the original file have no impact on the application created from the uploaded file. To have the changes take effect, you must upload the application again.

---

For Excel spreadsheets, you can capture data from the Excel file, choose the Excel sheet from the available list of sheets, and define grid rules on the captured data. Keep in mind the following:

- You can use Excel 95, 97, 2000, XP, and 2003 worksheets.
- All static data displayed in cells is captured, including positive and negative numbers. For negative numbers, if the format specifies parentheses, the negative number is captured within parentheses.
- UA supports these cell formats: General, Date, Time, Currency, Percentage, Accounting, Fraction, and Text.
- UA captures all data in the spreadsheet, regardless of whether filter rules are defined in the Excel file.
- Data in merged cells is captured, but will not be displayed as merged cells; for example, if cells in column 1, 2, and 3 are merged, data displays in column 1. If cells in column 2 and 3 are merged, data displays in column 2.
- Non-ASCII data in Excel files, such as cells, sheet names, and file names, is supported.
- Excel capture functionality is supported for BlackBerry and PDA online and offline clients, and for Smartphone offline clients.

The following capabilities are not supported:

- Password-protected Excel files
- Update, delete, and insert operations
- Cannot capture charts, formulas, and comments
- UA does not support these cell formats: Special and Custom (although the value, without formatting, is captured)
- You cannot remove the uploaded Excel file from Mobile Web Studio. You can navigate to the file and delete it manually.

---

**Note** The `excelfilelimit` property in the `global.properties.xml` file specifies the maximum file size allowed in KBs. If you have problems capturing an Excel spreadsheet, you may need to increase the file size from the default of 200 MBs to a larger size.

---

❖ **Creating a file element using an Excel spreadsheet**

You can create file elements using an Excel spreadsheet.

- 1 Log in to Web Studio, select Build Applications, select New from the Application Manager Status menu, then click New to launch Application Builder.
- 2 Click the arrow to the right of the Add button and select File Element. You see the New File Element window, which allows you to identify the file to upload from your local machine to the Unwired Accelerator server.
- 3 Complete the File Element Definition options:
  - File to Upload – enter the file name of your source data, or browse the file structure to select the file.
  - Destination Location – click Upload to accept the default location, which is created automatically in `SYBASE\PortletDocs\masuper`.  
Click OK to confirm.
  - File Name – leave blank. This field is automatically filled in once you upload the file.
  - Choose Worksheet – this option displays if you uploaded a worksheet. Select a sheet from the drop-down list.
- 4 Click Preview to see the result. Click Next.

- 5 In the Define window, define the record layout and specify a record header. See “Defining record layout” on page 133. Click Next.
- 6 In the Filter window, specify filter rules to select only the relevant data from the element. See “Adding filter rules” on page 133. Click Next.
- 7 In the Preview window, enter a name for the file element, and click Finish.
- 8 In the Application Builder, click Save. See “Saving applications” on page 76.

❖ **Creating a file element from a delimited file**

You can create file elements using files that use commas, tabs, semi-colons, colons, or spaces to separate data.

- 1 Log in to Web Studio, select Build Applications, select New from the Application Manager Status menu, then click New to launch Application Builder.
- 2 Click the arrow to the right of the Add button and select File Element. You see the New File Element window, which allows you to identify the file to upload from your local machine to the Unwired Accelerator server.
- 3 Complete the File Element Definition options:
  - File to Upload – enter the file name of your source data. You can browse to select the file
  - Destination Location – click Upload to accept the default location, which is created automatically in *SYBASE\PortletDocs\masuper*.  
Click OK to confirm.
  - File Name – leave blank. This field is automatically filled in once you upload the file.
  - Delimiter – this option displays if you uploaded a delimited file. Select the delimiter used in the source data file from the drop-down list. Options include tab, semi-colon, colon, comma, and space.
  - Merge Repeated Delimiters – this option displays if you uploaded a delimited file. Specifies whether repeated delimiters should be replaced with a single delimiter. This is useful if your source data includes blank columns. Select the checkbox to indicate “true” or leave it blank to indicate “false”.
  - Text Quoter – this option displays if you uploaded a delimited file. Specifies the quote character used to define a section of “as is” text that may include delimiters.

- 4 Click Preview to see the result.
- 5 Click Next.
- 6 In the Define window, define the record layout and specify a record header. See “Defining record layout” on page 133. Click Next.
- 7 In the Filter window, specify filter rules to select only the relevant data from the element. See “Adding filter rules” on page 133. Click Next.
- 8 In the Preview window, enter a name for the file element, and click Finish.
- 9 In the Application Builder, click Save. See “Saving applications” on page 76.

## Creating Domino applications

Using UA, one can capture Domino applications and can use them offline on the mobile devices, such as BlackBerry, J2ME, PocketPC or Palm devices.

The Domino wizard provided by UA allows you to select which views and agents to use to build your application. The UA Domino wizard also allows you to pass custom parameters as part of your business process implementation.

To capture a Domino database, UA requires Domino agents to be defined in the respective database which will take care of providing relevant data (such as values to be filled in combination, Actions possible on the document, and so forth). The Domino agent is also expected to perform validations and to update/insert Notes documents.

UA provides a way of passing custom parameters to an agent using the “Input Parameters” option. Users can specify custom parameters as per business needs which will be passed to the Domino agent. (Please refer to UA-Domino Agent Data flow section).

You need Domino connection information, permission to access Domino data, and agents. See the *Unwired Accelerator Administration Guide* for information.

## Setting up connection pools for Domino

Set up connection pools for Domino as described in the *Unwired Accelerator Administration Guide*. Check the release bulletin as well. In summary, set the Domino-related properties in *global.properties.xml*:

- **ConnectionPools** – by default, ConnectionPools is set to “defaultdomino.” The default ConnectionPool “defaultdomino” has a corresponding property file *defaultdomino.properties* in which Domino connection properties are defined. Each ConnectionPool name in the list must have its own property file *<poolname>.properties* in *...\\webapps\\onepage\\WEB-INF\\classes*, as described in the *Administration Guide*.
- **ConnectionType** – by default, ConnectionType is set to “WebSession.” Keep the default or change to either “diiop”, or “nrpc.”
  - **WebSession** – A Domino connection type used to create a connection between Unwired Accelerator and a Domino server. The Domino connection is required if you plan to develop mobile applications using Domino Business Objects. A WebSession connection requires that the Domino server is installed on the same machine as the UA server. Other connection types include “nrpc”, and “diiop.” websession is the default.

```
database_path=
username=
password=
password_encrypted=false
```

Modify the parameters for your environment.

- **DIIOP** – Domino Internet Inter-ORB Protocol (CORBA). A Domino connection type used to create a connection between Unwired Accelerator and a Domino server. A DIIOP connection requires a DIIOP port.

```
server=
diiop_port=
database_path=
username=
password=
password_encrypted=false
```

Modify the parameters for your environment.

- **NRPC** – Notes Remote Procedure Call. A Domino connection type used to create a connection between Unwired Accelerator and a Domino server. A NRCP connection type requires the Notes client be installed on the same machine as the UA server, and the user must provide an ID file for authentication.

```
server=  
database_path=  
user_ID_file=  
password=  
password_encrypted=false
```

Modify the parameters for your environment.

## Setting up Lotus Notes and Domino on one machine (WebSession)

If you want to run Lotus Notes e-mail on the same machine on which you are creating Domino elements, and you are using a WebSession connection, you will encounter a login conflict. To get around the problem, you must make some modifications to the Lotus Notes environment, which includes adding EXTPWD to the *notes.ini* file.

Thereafter, you must comment out EXTPWD to start Lotus Notes, enabling you to display the password prompt and log in; and then expose EXTPWD to start Domino, enabling you to bypass the password prompt that appears on the Tomcat console.

Complete the “Running Lotus Notes simultaneously” procedure before you continue.

### ❖ Running Lotus Notes simultaneously

- 1 As a safeguard, back up your *%LotusNotesDir%/notes/notes.ini* file.
- 2 Change *%LotusNotesDir%/notes/notes.ini* to include the EXTPWD value as follows:

```
EXTMGR_ADDINS=EXTPWD
```

You can append EXTPWD as comma separated if there are already values present; for example, `EXTMGR_ADDINS=NCExtMgr, EXTPWD`.

- 3 Copy the *nextpwd.dll* file to the *%LotusNotesDir%/notes/* directory.
- 4 Ensure that the PATH variable includes the path to the *%LotusNotesDir%/notes/* directory, and precedes the PATH of *%DominoServersDir%/Domino*.

- 5 Once this configuration is in place, use these steps to get Lotus Notes and Domino up and running:
  - a In the *notes.ini* file, comment out this value:

```
EXTMGR_ADDINS=EXTPWD
```
  - b Start Lotus Notes e-mail, log in, and make sure it is running.
  - c In the *notes.ini* file, remove comments so this value is exposed:

```
EXTMGR_ADDINS=EXTPWD
```
  - d Restart Tomcat.
  - e Run Domino.

## Creating Domino elements

This section assumes you have configured the Domino data source connection as described in the *Unwired Accelerator Administration Guide*.

- 1 In the Application Builder, click the down arrow next to the Add button and select Domino Element.
- 2 In the Domino Element Definition window:
  - a Select the Connection Type:
    - Connection Pool – allows you to choose from a list of preconfigured connection pools. The list of available preconfigured connections comes from the list specified in *global.properties.xml*.
    - Domino URL – allows you to manually configure connection properties. The user gets relevant fields based on the ConnectionType value selected in *global.properties.xml*:

For WebSession:

- Database Path – the path to the Domino database, such as `HelpdeskSystem.nsf`.
- Username – the Domino user name.
- Password – the internet password; for example, `sybase`.

For DIIOP:

- Server – the Domino server; such as `ua-dom1.sybase.com`.

- Port – DIIOP port value, such as 63148.
- Database Path – the path to the Domino database, such as HelpdeskSystem.nsf.
- Username – the Domino user name.
- Password – the internet password; for example, sybase.

For NRPC:

- Server – the Domino server; such as ua-dom1.sybase.com.
- Database Path – the path to the Domino database, such as HelpdeskSystem.nsf.
- User ID file – the Domino administrator’s user id file, such as //ua-dom1/cdata/admin.id.
- Password – the ID file password; for example, sybase.

- b Click Find. The “View,” “Agent,” and “Field” names from the View selection boxes are populated, according to your selection.
- 3 In the Domino Element Definition window, under “Application Type,” indicate whether to use Views to filter a Domino view, or to capture data using an agent:
    - Using Views to filter a Domino view:
      - 1 Select View to filter on Domino Views.
      - 2 Select a View name from the drop-down list; for example, Help desk View.
      - 3 Optionally, select a UI XSLT template, and return to the Domino Element Definition window.
    - Using Agents to capture data:
      - 1 Select Agent to use an agent to capture data.
      - 2 Select the appropriate Agent name to use. The agent returns data to be captured.

---

**Note** For information about agents, see “Understanding UA-Domino agent data flow” on page 61 and “Sample code for Domino agents” on page 64.

---

- 3 To pass parameters to the Agent:

- Select “Field names from View” if you want parameters to come from a Domino View, or  
Select a view name from “Field Names using View” drop-down list; for example, Help desk View.
  - Select “Input Parameter” if you want to specify Custom parameters.
  - Enter values as name value pair.; for example, param1=value1,param2=value2 and so forth.
- 4 Optionally, select a UI XSLT template, and return to the Domino Element Definition window.
  - 4 In the Domino Element Definition window, click Preview to view the document for the selected options.
  - 5 Click Next.
  - 6 In the Define window, define the record layout and specify a record header. See “Defining record layout” on page 133. Click Next.
  - 7 In the Filter window, specify filter rules to select only the relevant data from the element. See “Adding filter rules” on page 133. Click Next.
  - 8 The Configure Parameters window displays the element’s CGI parameters. See “CGI parameters” on page 89 and “Using the database table adapter” on page 202. Click Next.

---

**Note** While creating update applications, UA requires you to pass NoteID and Form fields to the agent. On the parameter definition screen, select the Type as “None” and Kind as “Update”. This will pass the NoteID and Form field values to the agent, but will not permit the user to modify it.

---

- 9 In Window Preview, enter an element name.
- 10 Click Finish, then Close to return to the Application Builder.
- 11 In the Application Builder, click Save. See “Saving applications” on page 76.

## Understanding UA-Domino agent data flow

Unwired Accelerator acts as a middle-man, and does not update the Actual Notes Documents itself, but provides a way of executing business logic.

UA gathers input given by user and passes a set of parameters to the Agent using a temporary Notes Document. The Domino Agent handles the business logic (validations, and so forth), and then saves the results to the actual Document.

The data flow is as given below:

- 1 User provides input either using “Input Parameters” or using “Field names using View” option. (This is while using an Agent “Application Type”).
- 2 UA gathers this input and creates a temporary Notes Document.
- 3 UA puts all the input provided by the user, along with some other fields (see Table 3-3 for fields) into the temporary Notes Document.
- 4 UA calls the Domino Agent specified, and passes the Temporary Notes Document ID to it for processing.
- 5 Domino Agent, based on the inputs provided, takes the necessary actions, and puts the Status message in the “domino\_status” field, and the output in the “domino\_output” field.
- 6 UA again reads the temporary document, extracts “domino\_status” and “domino\_output” field values, and passes them to the user. The user gets an appropriate message in case of failure.

Table 3-3 describes the format of the temporary document.

**Table 3-3: Sample format of a temporary document**

Field name	Value	Sample data	Description
form	ua_tempdoc	ua_tempdoc	Name of the Temporary Document’s Form field is “ua_tempdoc.”
dmno_server	<domino server name>	demo-server	Server name provided by user.
dmno_db	<Domino Database path>	demohelpdesk.nsf	Domino Database path (relative to Domino Server).
dmno_username	<domino user name>	demouser/sybase	Domino user name provided by the user
dmno_userid	<domino User ID file>	//demo-server/lotus/demouser.id	User Id file provided by the user. This is used only for NRPC connections.
dmno_diiop_port	<domino diiop port>	63148	Domino DIIOP Port. This is used only for DIIOP connections.

Field name	Value	Sample data	Description
orig_formname	orig_formname	<Original Form name>	HelpDesk Name of the actual form to which the Document data belongs.
orig_noteid	<Original NoteId>	3fe	NoteID of the actual form to be updated. The default is a Hex value; or blank in case of Insert.
domino_status	<Status Message>	ok	If the Agent execution was successful, the value should be "ok;" otherwise an error message displays.
domino_output	<Domino output in XML>	<pre> &lt;Record&gt;   &lt;Field label="Index"     position="1"&gt;Status   &lt;/Field&gt; &lt;/Record&gt; &lt;Record&gt;   &lt;Field label="Index"     position="1"&gt;     Accepted   &lt;/Field&gt; &lt;/Record&gt; &lt;Record&gt;   &lt;Field label="Index"     position="1"&gt;     Rejected   &lt;/Field&gt; &lt;/Record&gt; &lt;Record&gt;   &lt;Field label="Index"     position="1"&gt;New   &lt;/Field&gt; &lt;/Record&gt; &lt;Record&gt;   &lt;Field label="Index"     position="1"&gt;Pending   &lt;/Field&gt; &lt;/Record&gt; </pre>	<p>Resulting output in XML format:</p> <pre> &lt;Record&gt;   &lt;Field&gt;&lt;/Field&gt; &lt;/Record&gt; </pre>

Field name	Value	Sample data	Description
UA_ACTION	<Action value>	Blank	The Action value selected by user, such as Accept, Reject, and so on. This is an important field for making business decisions. The developer of the Agent must be aware of the possible actions and tasks to be performed, based on the action chosen by the user.
Custom Parameters	<custom parameter value>	Custom1	The custom parameters are either selected using a View or specified in the “Input Parameter” box as name, value pair. For each custom parameter a field name with the same name is created. This can then be used by Agent code to update/insert the actual document.

## Sample code for Domino agents

This section provides sample Domino agent code. The code examples are only meant to give you an idea as to how Domino Agents works. You will need to modify the agent code to suit your application business logic.

To create the Domino Agent, you need to have Designer access. Agent can be created in Notes Client by selecting Create | Agent. Specify the name of the agent and in the Runtime Section select “Agent list selection” and Target as “None”. This is important for UA to execute the Agent.

The Agent needs to return Data in the “domino\_output” field in the following format:

```
<Record>
  <Field label="Index" position="1">Value1</Field>
  <Field label="Index" position="1">Value2</Field>
  <Field label="Index" position="1">Value3</Field>
  ...
</Record>
```

## Data provider agent

The following code is a sample data provider agent, which was created using LotusScript.

```
Sub Initialize
  Dim Session As New NotesSession
  Dim doc As NotesDocument
  Dim db As NotesDatabase
  Dim paramDocId As String

  Dim agent As NotesAgent
  Dim xmlRet As String

  Set db = Session.CurrentDatabase
  Set agent = Session.CurrentAgent
  paramDocId = agent.ParameterDocID
  Set doc = db.GetDocumentByID(paramDocId)

  xmlRet = "<Record><Field label=""Index"" position=""1"">Status</Field>"
  xmlRet = xmlRet + "</Record>"
  xmlRet = xmlRet + "<Record><Field label=""Index"" position=""1"">1</Field>"
  xmlRet = xmlRet + "</Record>"
  xmlRet = xmlRet + "<Record><Field label=""Index"" position=""1"">2</Field>"
  xmlRet = xmlRet + "</Record>"
  xmlRet = xmlRet + "<Record><Field label=""Index"" position=""1"">3</Field>"
  xmlRet = xmlRet + "</Record>"
  xmlRet = xmlRet + "<Record><Field label=""Index"" position=""1"">4</Field>"
  xmlRet = xmlRet + "</Record>"

  Call doc.AppendItemValue("domino_status", "ok")
  Call doc.AppendItemValue("domino_output", xmlRet)
  Call doc.Save(False, False)
End Sub
```

## Update agent

The following code is a sample update agent, which was created using LotusScript. The agent performs some tasks based on the UA\_ACTION value (such as, Updates, Inserts), and sends e-mail with updated values in the e-mail's body.

```
Sub Initialize
  Dim Session As New NotesSession
  Dim maildoc As NotesDocument
  Dim doc As NotesDocument
  Dim orig_doc As NotesDocument
```

```
Dim db As NotesDatabase
Dim paramDocId As String

Dim agent As NotesAgent
Dim sendTo As String
Dim subject As String
Dim str1 As String
Dim suffixMsg As String

'field variables specific to business logic
Dim ticketNo As Variant
Dim users As Variant
Dim otherusers As Variant
Dim priority As Variant
Dim supporter As Variant
Dim department As Variant
Dim category As Variant
Dim type1 As Variant
Dim itemvalue As Variant
Dim solution As Variant

Dim requestValues As Variant

Dim actionValue As Variant
Dim ccToLength As Integer

Dim orig_noteid As Variant
Dim orig_formname As Variant
On Error Goto errorhandler

The agent has the skeleton code ready to send mail
'TODO :
' 1. Fetch values from document
' 2. Based on values fetched send mail
Print "Entered agent"

Set db = Session.CurrentDatabase
Set agent = Session.CurrentAgent
paramDocId = agent.ParameterDocID
Print "Before doc " + paramDocId
Set doc = db.GetDocumentByID(paramDocId)
Print "after doc"
'gather field values to mail
'doc.UserName is equivalent of doc.GetItemValue("UserName')
ticketNo = doc.TicketNo
users = doc.UserName
priority = doc.Priority
```

```
supporter = doc.Supporter
department = doc.Department
category = doc.Category
type1 = doc.Type
itemvalue = doc.Item
solution = doc.Solution
actionValue = doc.UA_ACTION

requestValues = doc.RequestValues
Print "Printing the request values.. "
Print Cstr(requestValues(0))
sendTo = Cstr(users(0))
Print "Before Substring"
str1 = Mid$(Cstr(actionValue(0)), 4)

    If(Strcompare(str1, "Save") = 0) Then
        suffixMsg = "Saved"
    ElseIf(Strcompare(str1, "CloseTicket") = 0) Then
        suffixMsg = "Closed"
    ElseIf(Strcompare(str1, "CreateTicket")) Then
        suffixMsg = "Created"
    ElseIf(Strcompare(str1, "Accept") =0) Then
        suffixMsg = "Accepted"
    ElseIf(Strcompare(str1, "Reject") =0) Then
        suffixMsg = "Rejected"
    Else
        suffixMsg ="Default"
    End If

If(str1 <> "Cancel") Then
    Print "Inside If"
    orig_noteid = doc.orig_noteid
    orig_formname = doc.orig_formname
    Set orig_doc = db.GetDocumentByID(Cstr(orig_noteid(0)))
    If(orig_doc Is Nothing) Then
        Print "Original doc not found for id - " + Cstr(orig_noteid(0))
    End If
    If(Cstr(orig_noteid(0)) = "") Then
        Print "Original note id is empty : CREATEDOC"
        'create new document
        'place to do validation and creating a new document
        Set orig_doc = db.CreateDocument
        Call orig_doc.AppendItemValue("Form", orig_formname(0))
        Call orig_doc.AppendItemValue("TicketNo", ticketNo(0))
        Call orig_doc.AppendItemValue("UserName", users(0))
        Call orig_doc.AppendItemValue("Priority", priority(0))
        Call orig_doc.AppendItemValue("Supporter", supporter(0))
```

```
Call orig_doc.AppendItemValue("Department", department(0))
Call orig_doc.AppendItemValue("Category", category(0))
Call orig_doc.AppendItemValue("Type", type1(0))
Call orig_doc.AppendItemValue("Item", itemvalue(0))
Call orig_doc.AppendItemValue("Solution", solution(0))
Call orig_doc.AppendItemValue("UA_ACTION", actionValue(0))
Call orig_doc.Save(False, True)
Else
Print "Original note id non empty : UPDATEDOC"
'update existing document
'place to do validations
'Dim richtext As New NotesRichTextItem(orig_doc,"Body")
Dim item As NotesItem

Set item = orig_doc.ReplaceItemValue("TicketNo", ticketNo(0))
Set item = orig_doc.ReplaceItemValue("UserName", users(0))
Set item = orig_doc.ReplaceItemValue("Priority", priority(0))
Set item = orig_doc.ReplaceItemValue("Supporter", supporter(0))
Set item = orig_doc.ReplaceItemValue("Department", department(0))
Set item = orig_doc.ReplaceItemValue("Category", category(0))
Set item = orig_doc.ReplaceItemValue("Type", type1(0))
Set item = orig_doc.ReplaceItemValue("Item", itemvalue(0))
Set item = orig_doc.ReplaceItemValue("Solution", solution(0))
Set item = orig_doc.ReplaceItemValue("UA_ACTION", actionValue(0))
Call orig_doc.Save(False, True)
Print "Replaced values"
End If
Set maildoc = New NotesDocument( db )
'prepare to send mail
Dim body As NotesRichTextItem
Dim lintTo As NotesDocument
Print "about to send mail to " + sendTo
maildoc.Form = "Memo"
maildoc.SendTo = sendTo
'maildoc.CC = supporter(0)
subject = "Your ServiceDesk Ticket# " + Cstr(ticketNo(0)) +
" has been " + suffixMsg
maildoc.Subject = subject
Print "preparing body"
Set body = maildoc.CreateRichTextItem("Body")

Print "created body"
body.appendText("Helpdesk values are, ")
body.AddNewline(1)
body.appendText("Ticket No = " + ticketNo(0))
```

```

body.AddNewline(1)
body.appendText("UserName = " + users(0))
body.AddNewline(1)
body.appendText("Priority = " + priority(0))
body.AddNewline(1)
body.appendText("Supporter = " + supporter(0))
body.AddNewline(1)
body.appendText("Department = " + department(0))
body.AddNewline(1)
body.appendText("Category = " + category(0))
body.AddNewline(1)
body.appendText("Type = " + type1(0))
body.AddNewline(1)
body.appendText("Item = " + itemvalue(0))
body.AddNewline(1)
body.appendText("Solution = " + solution(0))
body.AddNewline(1)
body.appendText("Action taken = " + actionValue(0))
body.AddNewline(2)
'If(Cstr(orig_noteid(0)) <> "") Then
Call body.AppendDocLink(orig_doc, "Double click to open it")

End If
'maildoc.Body = body
Call maildoc.Save(False, True)
Print "about to send"
Call maildoc.Send( False )
Print "mail sent"
xml="<Record><Field label=""Index"" position=""1"">Status</Field>
xml = xml + "</Record><Record>"
xml = xml + "<Field label=""Index"" position=""1"">Record updated
      successfully</Field>"
xml = xml + "</Record>"
'Call doc.AppendItemValue("form", "xmldoc")
Call doc.AppendItemValue("domino_status", "ok")
Call doc.AppendItemValue("domino_output", xml)

Exit Sub ' forcibly exiting
End If

errorhandler :
Set doc = db.GetDocumentByID(paramDocId)
xml = "Msg from postprocessor: Completed with errors"
'Call doc.AppendItemValue("status_desc", xml)

Call doc.AppendItemValue("domino_status", xml)
Call doc.AppendItemValue("domino_output", xml)

```

```
Call doc.Save(False, False)
' Display the values on entry to the
' error-handling routine.
Print "Error: " Error(), " Err:" Err(), " Erl:" Erl()
Print "End.."+paramDocId
Resume Next
End Sub
```

## Creating SAP applications

Using UA, you can mobilize your SAP ERP application data securely to your mobile users. Mobile users who use BlackBerry, J2ME, PocketPC or Palm devices can access their SAP business applications from their devices. This gives mobile users the ability to access information at all times, regardless of network connectivity and coverage.

---

**Note** When using SAP applications in online mode, or in Portal Interface, you are able only to read and navigate through them. You cannot insert or update – insert and update functionality is available only in offline mode.

---

The SAP Business Object API (BAPI) interface consists of business objects such as accounts, customers, companies, which are grouped according to the application in which the objects are used, for example financial, human resources, customer support, and health care. Each object has a set of methods that allow you to search, create, and modify data related to that object. These BAPI calls are internally translated to Remote Function Call (RFC) calls by the UA runtime.

The SAP wizard provided by UA allows you to select which BAPI business object to build your application with then navigates to the specific method on that object that the application should call. The UA SAP wizard also allows you to access custom RFCs that you developed as part of your business process implementation. Accessing custom RFCs is the same as accessing the standard BAPI objects that come with a default SAP installation. UA also allows you to choose from among your SAP systems in cases where you may have multiple SAP installations.

The UA wizard for mobile application development provides online API documentation for business objects, methods, method parameters, structures, and tables. As you develop the SAP application, the wizard retrieves and displays the relevant online documentation.

## Creating SAP elements

- 1 In the Application Builder, click the down arrow next to the Add button and select SAP Element.
- 2 In the SAP Element Definition window:
  - a Select the Connection Type:
    - Connection Pool – allows you to choose from a list of preconfigured connection pools. The list of available preconfigured connections comes from the list specified in *global.properties.xml*. The default ConnectionPool is used for both capture and playback.
    - Direct – allows you to manually configure connection properties. The direct connection is used for playback only.

Click Details to launch the Direct Connection configuration window.

The configured direct connection properties are @OP parameterized in an application definition to allow for further configuration for a setup screen when being played back. This also allows for auto-fill for single sign-on.

When you are finished entering the connection properties, click OK to return to the SAP Element Definition window.
  - b Function – click Find to launch the Find Function window. In the Find Function window, you can browse through all available objects and their functions.

Click Select to choose an object or function.

---

**Note** When you move the mouse over an object or function, a tool tip appears with a brief description of the object or function.

---

If you know which function you want, you can enter the name of the function in the Function field, and click Find.

When you are finished, click OK to return to the SAP Element Definition window.

- c Once you identify a RFC, the Input Tables and Parameter Tables fields are automatically populated. Hovering the mouse over a table or parameter in any list brings up a brief tool tip description of the item under the cursor.

Use the Add, Add All, Remove, and Remove All buttons to move input tables between the Input Tables Available and Selected boxes.

- d Use the Add, Add All, Remove, and Remove All buttons to move parameters between the Parameters/Tables Available for Output and Selected boxes.
- 3 Click Enter Value to launch the Input Values window, where you must provide the information for all the selected input parameters and tables. Click OK when you are finished. You return to the SAP Element Definition window.
  - 4 In the SAP Element Definition window, click Preview to verify the BAPI executes successfully before proceeding further.
  - 5 Click Next.
  - 6 In the Define window, define the record layout and specify a record header. See “Defining record layout” on page 133. Click Next.
  - 6 In the Filter window, specify filter rules to select only the relevant data from the element. See “Adding filter rules” on page 133. Click Next.
  - 7 This window displays only if you have defined the record layout. The Configure Parameters window displays the element’s CGI parameters. See “CGI parameters” on page 89 and “Using the database table adapter” on page 202. Click Next.
  - 8 In Window Preview, enter an element name.
  - 9 Click Finish, then Close to return to the Application Builder.
  - 10 In the Application Builder, click Save. See “Saving applications” on page 76.

## SAP login

Before you can access SAP functions, you must log in to the SAP server. You can configure SAP with a security policy to control access to data based on user IDs. When you build SAP applications, you must decide if the application should always use some fixed-system user to access SAP data, or if each user's SAP user ID should be used for access.

## Client connections and connection pools

If a shared SAP user ID is used, then the connections to SAP must be defined as part of a connection pool. The connection pool allows multiple, concurrent end users to access SAP efficiently.

The SAP information in the *global.properties.xml* file is shown below:

```
<PropertyGroup name="SAPGroup" description="properties
for configuring SAP in the Portal and Studio.">

    <!-- define the "Default" SAP connection pool to use.
The username provided here must have access rights to
call a number of RFMs for metadata repository access
(Authorization Object: S_RFC, ACTVT: 16, FUGR). -->
    <Property name="DefaultConnectionPoolName"
value="sapjco" description="connection pool name for
the default SAP connection" menugroup="100" />
    <Property name="ConnectionPools"
value="Default,custom1,another2" description="Comma
separated list of connection pool names"
menugroup="100" />
</PropertyGroup>
```

The ConnectionPools property gives a list of different connection pools that are defined. Each connection pool on this list has a corresponding *<poolname>.properties* file in the *..\webapps\onepage\WEB-INF\classes* directory. There is one sample connection pool properties file provided called *sapjco.properties* with comments on how to set the connection pool properties. Copy this file and modify it to create additional connection pools. The contents of this file:

```
# These properties are used to establish the default SAP
# connection pool.

# Other connection pools can be created by copying this
# sapjco.properties
```

```
# file to "poolname".properties and modifying it.
#
# The complete set of properties and their meanings are
# documented in
# Java Connector (JCo) JavaDocs for the
#JCO.createClient(java.util.Properties) method.
#
# The first 2 properties are NOT part of the connection
#information:
# specify the maximum number of connections the pool
should
#allow - default is 10
#sybase.maxconnections=10
# maxwaittime is time in milliseconds to wait for a
connection
#to become
# available when maxconnections are already in use. The
#default is 30 seconds.
#sybase.maxwaittime=30000
#
# The default connection pool is used to access the
#JCO.Repository.
# In order to retrieve the necessary information from
the SAP
#system's data
# dictionary, the repository needs to call a number of
RFMs
#for which the
# access rights have to be granted (Authorization
Object:
#S_RFC, ACTVT: 16, FUGR).
# R/3 Release Function Groups
```

```
# since 3.1H RFC1, SG00, SRFC, SUNI, SYST
# since 4.0A RFC1, SDIF, SG00, SRFC, SYST, SYSU, SUNI
# since 4.6A RFC1, SDIF, SG00, SRFC, SYST, SYSU
# since 4.6D RFC1, SDIFRUNTIME, SG00, SRFC, SYST, SYSU
jco.client.client=400
jco.client.user=sap
jco.client.passwd=lsap
jco.client.lang=EN
jco.client.ashost=sap.try.sybase.com
jco.client.sysnr=00
```

The `DefaultConnectionPoolName` property indicates which of the pool names in the `ConnectionPools` list is the one to use by default.

## Reordering columns

Data returned by SAP RFCs can have very large numbers of columns (from 30-100) for each row. The order of the columns returned is not in order of importance, and custom columns are typically at the end. UA allows you to easily define label names and order them with the Label functionality in the Application Builder. Once you define and order the labels on the Application Labels window, you can map the new labels to existing column names in the Application Builder. See “Labels” on page 94.

Clicking on Import Label in the Application Labels window automatically imports the currently defined labels from the application definitions, which means you do not have to manually type the individual labels in. Selecting Map All Labels automatically maps labels to existing header column names by comparing the existing header column names with label names.

## Selecting the header row

The header column for SAP applications is usually defined as the first row. This row of data is the name of columns returned by the BAPI/RFC, and are typically names of the database columns. UA returns a second row of data in the grid that includes the descriptive names for the columns so that you can decide which row of data should be used for labeling the columns, whether it is the database name or the descriptive name.

## Saving applications

---

**Note** When you save a secure application, it does not automatically appear in the Application Manager detail view when you select the New status. To view the application in the detail view, select a different status, then reselect New. The new secure application listing should be visible.

---

When you create and save an application with the Application Builder for the first time, you see the Finish window. The options for saving the application display on these tabs, depending on the application type:

- Content tab
- Mobile tab
- Roles tab
- Presentation tab
- List/Detail tab
- Data Validation tab
- Administration tab
- Answers Anywhere

You can also edit an application any time by clicking Properties in the Application Builder. This section describes the available options when you save or edit an application.

### ❖ Saving applications

- 1 When you finish creating an application and return to the Application Builder, click the down arrow next to the Save button to save the application.
- 2 In the New Application Finish window, complete the properties for the new application by clicking any of the applicable tabs.  
  
You must enter a name for the application in the Content tab. All other properties are optional.
- 3 Click Finish. After a message displays that the application was saved successfully, click OK to confirm and return the Application Builder.

## Content tab

These options let you set content-specific options, such as the character set in which an application's text displays. The options on this tab are:

- Name – a unique name for the application that contains this element.
- Content Type – the type of content the application displays. You can select any of these options from the drop-down list:
  - Application/msword
  - Application/pdf
  - Application/x-msexcel
  - Application/x-mspowerpoint
  - image/gif
  - image/jpeg
  - text/html
  - text/plain
  - text/rtf
  - text/xml
  - Add New – type the new content type in the box that displays when you select this option.
  - XSL template – displays only when you select “text/xml” as the Content Type. Select an existing XSL template, or create a new one to apply to the application's content.
  - Output Type – displays only when you select “text/xml” as the Content Type. Select the type of output in which the application's content displays.
- Charset – the character set in which the application's content displays, such as `utf-8`.
- Content Cache Interval – the interval at which the content of an application is refreshed for the defined source. If this value is not Real Time, the content is held by the cache for the duration of the specified interval. All requests made during this interval receive the cached content.
  - The default is Real Time, which means that application content is retrieved from its source on every playback request.

- Specify a preconfigured value either in hours or minutes.
- Specify custom values by selecting Add New from the drop-down list. Select the minute/second drop-down list and enter a value in the text box.

Using content caching can save time when the application loads, especially if the application contains a lot of data. When content cache expires, the next time the application is loaded, the content is pulled from the source again, and the new content is stored in the cache. In the case of data-heavy applications, this can take some time. One way you can avoid slow loading of applications is to create an agent to run the application periodically so that the agent refreshes the cache when it is run after the cache expires. See “Using agents” on page 241.

- Parameter – create an invisible parameter to which you can assign click-across events.

Invisible parameters is a Portal Interface feature. The idea for this feature comes from client-side click-across when the event is across pages. In an across-page client-side event, focus is shifted to the page where the listener portlet resides. However, if the portlet you want to shift focus to has no parameters, it cannot be a client-side listener and therefore you cannot shift focus to the portlet’s page.

To fix this problem, check the Invisible Parameter option for the application that should receive the focus to create a fake parameter on the application. Using this fake parameter, you can define listeners on the application, enabling it to receive focus in across-page client-side events.

On the Define Listeners window the fake parameter name appears as “Portlet Param.”

- Secure – whether this is a secure application.

- In Context – indicates that the click-thru session should remain in context (that is, continue to use the internal HTTP client to retrieve destination pages pointed to by links on the page), rather than opening a separate browser connection to each link’s target.

---

**Note** The In Context option helps address capture problems when a session state is associated with a played-back page, and the user clicks an application link and the linked content displays in a new browser window, which results in the session state being lost.

---

- No URL Stretch – disables URL stretching (rewriting) on played-back application. If you are having problems with broken images in a played-back application, this sometimes solves the problem.

---

**Note** On BlackBerry devices, when using the Sync with Images option on the Sync All screen, the images do not display correctly unless you select this option.

---

## Mobile tab

“Make available for disconnected mobile devices” – if you are using any offline mobile device, select this option to make the application viewable on the device, or on a device simulator.

Linked Application – allows you to set up a link between two applications so you can update the application’s source content from a PDA and link the source application with the update application. Once you have linked the applications, specify the following:

- Application
- Display Name
- Action Type (Insert, Update, Delete, Verification, Login)
- Menu Level (1, 2, 3, 4, 5)

To remove the link between the applications, click the X box to the left of the application.

See “Linking applications” on page 144 for more information.

---

**Note** This functionality is for use with mobile applications only. Linked applications do not work in Portal Interface.

---

## Roles tab

Unwired Accelerator provides system roles, such as StudioAdmin and PortalAdmin, and user-defined roles, which are created by the administrator. Roles provide individual accountability for users performing operational and administrative tasks.

Users with roles in the Assigned Roles list are the only users that can access the application. Select roles from the Available Roles list and click Add to move the selected roles to the Assigned Roles list. See the security chapter in the *Unwired Accelerator Administration Guide* for detailed information about managing roles.

## Presentation tab

The options on this tab define how the application and its content display.

- Title URL – the Web address, or URL, of the captured content.
- Help URL – the location of the online help file associated with the application. Click Select to browse for and select the file.

See Chapter 6, “Building Templates” for information on creating help files.

- Custom Setup Page – use this option to specify the file path to an alternate setup page that contains whatever validation code you want to write. The default setup page is located in *onepage\oembase\windows\setupPage.jsp*.

You can create a copy of the default setup page and add custom validation code near the top of the JavaScript function called

```
doPersonalizeOk<%=windowId%>.
```

- Refresh Interval – the interval at which the application’s content is refreshed. The browser automatically makes a request for the application content when this interval runs down to zero.
  - The default is Daily, which translates into 1440 minutes.

- Specify a pre-configured value either in hours or minutes.
- Specify custom values by selecting Add New from the drop-down list. Two additional fields display where you enter the new custom value and select whether that value is in minutes or seconds.
- Default Size – this option is available only when Display Within IFrame is not selected. When you unselect Display Within IFrame, you can choose from several predefined heights, or select Add New. When you select Add New, an input field appears where you enter the application height as a numeric value, in pixels. The field accepts only numeric values, for example “200.”
- Display Direction – the direction in which the application’s content displays—LTR (left to right) or RTL (right to left).
- Display Within IFrame – displays the application in an inline frame that displays inside of another HTML document or displays directly on a page with other elements flowing around it.

---

**Note** Unlike frames created using `<FRAMESET>` and `<FRAME>`, `<IFRAME>` creates a frame that sits in the middle of a regular nonframed Web page. `<IFRAME>` works like `<IMG>`, only instead of putting a picture on the page, it puts another Web page on the page.

---

These two options are available only when Display Within IFrame is selected.

- No Popup – enables this application’s links to display inline; that is, links do not display in a separate pop-up window.
- Height – the height at which the application displays. You can show all heights, select a preconfigured height, or select Add New to add a new height in pixels.
- Editable – whether the application can be edited by users in the roles to view the application.
- Show Last Refresh – display the latest refresh image when the application displays.
- Alert – schedule or externally trigger application content for processing.

## List/Detail tab

This is used to identify the columns to include in the list and detail views that display on the mobile device. By default, all columns are in the Selected list, indicating all columns will be displayed in the list and detail views on the mobile device client. You can move columns in or out of a view to customize what displays on the mobile device.

---

**Note** The List/Detail tab displays only if the content includes grid or list data.

---

In addition to setting up default list and detail views, on the List/Detail tab you can:

- Configure custom list and detail views that override the default settings for the individual screen resolution; for example, you may choose to display five columns in the detail view of a wide screen, but only four columns in the detail view of a smaller screen.
- Select a template for the individual screen resolution; for example, you may have developed a template that uses soft colors for a smaller screen, and one that uses vibrant colors for a larger screen.

### ❖ Configuring individual devices from the List/Detail tab

- 1 Under Device Config, select the radio button to the right of the device you want to configure; for example, select 320x240.
- 2 For the device selected you can:
  - a Click Find to select a template for the device. Use Find Template to search for and apply a template, such as Nokia Online 320x240. Close the window. The template name displays to the right of the device type.
  - b Click Custom to modify the List/Detail view for the device. Use Application Template Customization to make List/Detail entries for the device type.

**<device> Device** Enter a name for the template, or use the default.

**List/Detail** Select the menu options to appear on the List view and the Detail view on the mobile device.

Save and click OK to confirm, then click Close.
  - c Click Preview to preview your template selections. You can see the proposed changes in the pop-up window. Close the window.

- d Optionally, click Clear to remove the template and start over. The template name is removed from List/Detail Template column to the right of the device.
- 3 You can select the radio button to the right of another device. Any device that you do not change uses the default List/View.

## Data Validation tab

The Data Validation tab allows you to validate user-entered data of an application that contains grid data by entering regular expressions.

- Validation Expression – the validation rule in the form of a regular expression.
- Test – click this button to launch the Regular Expression Tester popup window, where you can test the regular expression.
  - Regular Expression – enter the regular expression you want to test.
  - Test Value – enter a test value.
  - Test Result – the results of the test appear after clicking Test.
- Common Expressions – this drop-down list provides a set of predefined, frequently used regular expressions you can select from.
- Detailed Instructions – this is an optional field you can use to provide additional information to the client mobile device user, for example, `Include area code with phone number OR Enter full 9 digit zip code`. On the client device, the Detailed Instruction message displays near the input fieldnote that is included with the form field on the client device. This may not apply for all client devices—it depends on the screen size and capabilities of the device.
- Error Message – enter a customized error message to be sent to device client device users when their entered data does not match the validation expression. Instead of receiving the usual generic error message, the user receives a more informative message, for example, `Zip code must have nine digits`.

## Administration tab

These options allow you to organize applications into categories, enter development notes, indicate whether the application is inactive (all applications are active by default), and apply versioning controls.

- Category/Subcategory – a category and subcategory that best describe the application.
- Description/Notes – an optional description or note.
- Active – indicates whether the application is active, which is the default. Unselect the option to inactivate the application. Only applications with an approved status and marked as active are available to users.
- Versioning – creates a new version of the application each time you save it. Each “Save” creates a new version number.
- Status – display or modify application status (New, Pending, Approved, Rejected, Archived, Broken, and Deleted). This gives you an option to approve the application as you are creating it, rather than using “Approving applications” on page 87.

## Answers Anywhere

The Answers Anywhere option enables you to access application content using natural language queries using a browser or mobile device. Answers Anywhere must be enabled in the `global.properties.xml` file as described in the *Unwired Accelerator Administration Guide*.

- Application Synonyms – allows you to specify alternate terms to identify the application in user queries.
- Field Synonyms
  - Available Fields – fields that should not be used to identify an application in a query. Leave common fields like name and ID in the available list.
  - Assigned Fields – fields you can use to identify the application. For example, the query “Get manager for John Doe” works only if the “manager” field is in the assigned list. Do not assign common fields like name and ID that may cause ambiguities.

If the application contains grid data, the result of the query displays data only from the column that matches the name used to identify the application. For example, if the application has three columns named *firstname*, *lastname*, and *phone* respectively, and your query is “get phone for Howard,” only the data in the *phone* column is returned.

- Field Synonyms – alternate names for the assigned field, such as “boss” or “supervisor” for the “manager” field.
- Parameter Synonyms
  - Parameters – list of all exposed parameters in the application (the ones you selected in the Parameters window).
  - Assigned Synonyms – assign alternate names to parameters. For example, a Yahoo stock parameter is called “s,” which has no real meaning, so you can assign a synonym that has more meaning, such as “ticker” or “symbol.”
- Check Conflicts – this button launches a window that alerts you to any potential conflicts in synonyms with other applications. Only synonyms that are used to identify an application are checked. In other words, application synonyms are tested against application synonyms and field synonyms from other applications, and field synonyms are tested against the application synonyms and field synonyms of other applications. Parameter synonyms are not checked for conflicts because they are not used to identify applications.
- Default Filter Field – specify which field to use as a default when dynamically creating filter rules.

For example, if you have a database application that shows customer names and contact information in one big table in the application and you want to filter out only one company’s data, you must normally create a query that says something like “get CustInfo *name* Company A,” where *name* is the header of the name field in the table.

If you set “name” as the default filter field, you can simply say “get CustInfo from Company A” and UA automatically acts upon the “name” field.

- Maximum Number of SMS Messages – specify the maximum number of chunks to send if the response content of an SMS query exceeds the 160-character limit and, therefore, must be divided into chunks.

## Previewing applications

The Application Builder right pane displays a preview of the application. You can also preview the application for several device type simulators.

### ❖ **Previewing an application on device simulators**

- 1 In Application Builder, select an option from the Device Type drop-down list:
  - Portal – this is the default setting used to display the preview in the Application Builder right-pane.
  - BlackBerry Online
  - Nokia Online
  - Palm Online
  - PPC Online
  - Smartphone Online
  - WAP-HTML
  - WAP-WML
  - PDA Online
- 2 Click Preview. The mobile application displays in a separate window, using the current device template associated with the application.
- 3 You can make selections, drill down from List View to Detail View, and so forth. You may decide to edit the mobile application or the device template based on what you experience.
- 4 Close the window, or repeat the procedure to view the mobile application on a different device type. You can display several device type windows at once for comparison.

## Approving applications

When you finish creating and saving an application, Mobile Web Studio automatically gives it a status of New and it is marked Active. Only applications that are marked as approved and active can be assigned to pages and deployed to a mobile device or Portal Interface.

---

**Note** You can also approve an application when you create it, or when you edit it, as described in “Presentation tab” on page 80.

---

- 1 To change an application’s status, select Build | Applications.
- 2 Select the status of the application you want to reassign.
- 3 Right-click the application in the detail view, select Status, then select the new status that you want to assign the application.

## Viewing and editing applications

Use the Application Builder to view and edit applications. Some modifications can only be made after you save the application.

### ❖ Viewing or editing applications

- 1 In Mobile Web Studio, select Build | Applications.
- 2 In Application Manager, select Status | Approved.
- 3 Right-click the application in the detail view, and select Edit. The Application Builder window displays.
- 4 Select an option from the Application toolbar:
  - Properties
  - Parameters
  - Labels
  - Data types

- Templates – see Chapter 6, “Building Templates”

---

**Note** Information about the Application toolbar is provided in “Application Builder toolbar” on page 16.

---

- 5 Select another option from the toolbar.
- 6 When you are finished, click Save, and OK to confirm.

## Properties

The Properties option lets you view and edit the properties that were specified when the application was created. This section describes the properties you can edit after you save an application. For all other application properties, you can edit them as described in “Saving applications” on page 76.

- ❖ **Viewing or editing application properties**
  - 1 Click Properties on the Application Builder toolbar. The Properties Editor window displays, which is similar for all element types.
  - 2 Select any tab and make changes.
  - 3 When you finish, click OK to save the changes.

## Content tab

On the Content tab, you cannot change the Name, but you can change any of the other properties and options that display.

## Statistics tab

The Statistics tab displays only when you deploy the Properties Editor, not when you are saving an application. This tab displays the user who created or modified the application, the creation and modification dates, and other reference information such as the version, application ID, and resource ID. Users are grouped under resources. Each resource corresponds to an Unwired Accelerator co-brand.

## Parameters

The application parameters identified for an element can be configured during application creation in one of the element definition windows or after application creation using the Application Builder Configure Parameter wizard. You launch the wizard by clicking the Params button on the Application Builder toolbar.

Mobile Web Studio allows you to identify or create application parameters that control the application content. These can be CGI script parameters for Web elements or custom @OP parameters for XML and database elements. @OP parameters are also automatically configured for the input parameters of SAP applications. You can specify these parameters as variables, which causes the application to display input fields during playback. End users can submit their own parameter values into these input fields to return personalized content.

## CGI parameters

A CGI program, or script, is a way users interact dynamically with Web pages; for example, submitting a string to a search form. When parameters are submitted to a CGI script to navigate to the Web page, the program automatically detects these parameters and their values. You can designate one or more CGI parameters as variables. Applications with variable parameters display a window containing input fields for these parameters before application content is rendered. This allows portal users to submit their own parameter values to return personalized content.

When the element contains data-capable parameters, a window displays during the element creation process that displays all parameters that you can designate as variables.

In addition to specifying variables for an element, you can also select a personalization adapter to use with the parameters. You can select the username and password parameters as variables, then select the personalization username/password login adapter. The Key represents the value that a user needs to enter.

Users can use the My Info link in Portal Interface to personalize the values for these input fields so that they can always log in using their own user name and password. See “Using the user name/password adapter” on page 202 for more information.

Whether you access the parameter configuration windows during application creation or by clicking Params, you can edit the following parameter properties.

---

**Note** The parameter options display in a different order and on more than one screen when you click Params to access the Configure Parameters window. The order shown here is how the parameter options display during application creation.

---

- Position – use the up and down arrows to organize the parameters in the order you want them to appear in the application.
- Variable – select the parameter that you want to configure as a variable. This option is labeled “Configure” in the Configure Parameters wizard.
- Name – displays the parameter name used in the CGI script.

---

**Note** If you change the CGI parameter name, you must also change the names of any events associated with this application as there is a dependency between the name of the CGI parameter and the name of events associated with it.

---

- Display Name – enter an alternate name to use as the input field label.

---

**Note** You can assign two or more parameters the same display name to combine input parameters. When you are prompted to enter input for the parameter, that input is used to fill in all the parameters with that display name.

---

- Default Value – enter a default value to automatically populate the input field. The user can override this value by typing in a different value.
- Display Type – select the input field type: None, Text Field, Calendar Date, Password, Select, Text Area, Boolean, Grid, and Linked. Keep in mind the following guidelines:
  - When using the username/password personalization adapter, you must select Password (rather than Text Field) for this option.
  - When you select Select, Add and Clear buttons display next to the parameter. Click the Add button to add name:value pairs for default values.

- When you select Grid, Add and Clear buttons display next to the parameter. Click the Add button to search for the application that produces the tabular data for this input parameter.
- When you select Linked, Add and Clear buttons display next to the parameter. Click the Add button to search for the application to which to link. See “Linking parameters” on page 152 for more information.
- Personalize – personalization adapters retrieve values from external systems or databases and automatically submit them to the application. They must be registered with Mobile Web Studio before they appear in the Personalize drop-down. Once a Personalize adapter is selected, the adapter’s available methods display in the Key drop-down.  
Key – select a method to specify the parameter within the personalization adapter.
- Kind – select the kind of operation to perform, Search or Update.
- Required – specify whether the parameter is required.
- Enter Setup Description HTML – this option appears only in the Configure Parameters wizard. Enter HTML code in the box that displays.
- Advanced – click this button to launch the Advanced Parameter Options window, where you can select the JSP template user interface you want to assign to the parameter.

## Assigning JSP templates

When you click on the Advanced button in the Configure Parameters window, you see the Advanced window, where you can open the Find Template window to search for the JSP template user interface you want to assign to the parameter.

- 1 In the Advanced window, click Select next to the parameter you want to assign the JSP template to.
- 2 In the Find JSP Template window, select JSP from the Type drop-down list, and click Search.
- 3 In the Results pane, a list of available JSP templates displays. Highlight the JSP template you want to assign to the parameter, and click Select.
- 4 You return to the Configure Parameters window. Click OK.

After saving the changes, the JSP template ID is stored in the application XML as an attribute–`template_id`–for each `<Parameter>` element. For example:

```
<CGIParameters>
  <Url>
    <Parameter name="day" default_value="26"
auto_fill_adapter="" auto_fill_key="" type="TEXT"
kind="search" sequence="1" variable="true"
required="true" display_name="day" cgi_mapped="true"
template_id="211"/>
    <Parameter name="month" default_value="01"
auto_fill_adapter="" auto_fill_key="" type="TEXT"
kind="search" sequence="2" variable="true"
required="true" display_name="month" cgi_mapped="true"
template_id="221"/>
    <Parameter name="currency" default_value="USD"
auto_fill_adapter="" auto_fill_key="" type="TEXT"
kind="search" sequence="4" variable="true"
required="true" display_name="currency"
cgi_mapped="true" template_id="" />
    <Parameter name="year" default_value="2005"
auto_fill_adapter="" auto_fill_key="" type="TEXT"
kind="search" sequence="3" variable="true"
required="true" display_name="year" cgi_mapped="true"
template_id="" />
  </Url>
</CGIParameters>
```

The XML code above is a fragment of the application definition, which shows the CGI parameter definitions.

## @OP parameters

You can insert an @OP parameter into the SQL query of a database. This dynamically modifies the query string submitted to the database.

---

**Note** When creating a database application with the “like” clause, for example where xyz like, and using an @OP parameter, you must use % in the SQL statement. For example:

```
select * from gemp
where fname like '%@OP["fname"]="Ger_t"%'
```

---

❖ **Inserting @OP parameters**

- 1 Create a new database element:
  - Right-click the mouse in the Application Manager detail view and select Aggregated Application from the pop-up.
  - Click the down arrow beside the New icon on the Application Manager toolbar and select Aggregated Application from the drop-down menu.
- 2 When the Application Builder appears, right-click in either frame and select New Database Element from the pop-up. You can also click the down arrow next to the Add icon and select Database Element.
- 3 Complete the Database Element Definition:

---

**Note** This example uses an Adaptive Server database Enterprise.

---

- Username – enter the user name to access the database; for example:

```
sa
```

- Password – enter the password to access the database. Leave this field blank for this example.
- JDBC Connect URL – enter the connection string to access the database; for example:

```
jdbc:sybase:Tds:<ASE_server_name>:5000
```

- JDBC Driver – enter the JDBC driver used to access the database; for example:

```
com.sybase.jdbc3.jdbc.SybDriver
```

- SQL Query String – enter the SQL statement to retrieve the requested data from the database; for example:

```
select * from tiles where header like '@OP["Header"="%CNN%"]'
```

To determine other header values, use a wildcard search '%’.

- UI XSLT – enter the complete URL of the XSLT to be applied to the content returned by the database or click Select to select from existing XSL templates.
- 4 Select the Variable option next to the Header parameter and enter “%CNN%” into the Default Value input field.
  - 5 Click Next in the upper-right corner.

- 6 Enter an element name such as “DB Search” and click Finish.
- 7 In the Application Builder, click Save to display the Application Save window.
- 8 Complete the fields and click OK.
- 9 Click Finish in the upper-right corner. You should see a pop-up that displays “Application saved successfully.”
- 10 Click OK to close the window, then click Close in the upper-right corner.

## Labels

Application labels allow you to create custom field labels for data-capable elements. Application labels let you combine multiple data-capable elements into a single table, sort records by a particular field, and modify the column order. You can map labels to table columns or rows. A mapped element is created when you map a grid element that has one or more columns mapped to application labels.

See “Working with data-capable elements” on page 130.

### ❖ Creating application labels

- 1 In the Application Manager, select an application listing from the detail view for which you want to add labels and click Edit.
- 2 When the Application Builder displays, click Labels on the toolbar.
- 3 In the Application Labels window, click Import Label to automatically import the defined labels from the application definitions. You define labels in the Define window when creating an application. See “Defining record layout” on page 133.

Or, you can enter the text you want as a label for one of the columns, then click the plus sign to add that label. Continue this process until you have created all the labels you need.

Use the up and down arrows under Sequence to reorder the labels.

- 4 For the Sort Records By option, select:
  - Sort Records By – to sort the table records by one of the labels listed in the drop-down list.
  - Ascending/Descending – to sort the records in ascending or descending order.

- 5 Select the Map All check box to automatically map labels to existing column headers. Automatic mapping is only for headers that have a mapping label name.
- 6 Click OK to save your entries and close the window.

❖ **Assigning application labels**

You do not need to perform this procedure if you select the Map All check box. These are the manual steps for assigning application labels.

- 1 Click the name of a data-capable element in the Element List. This displays the element field labels below the element name and the associated Application Label drop-down lists.
- 2 Select an application label from the drop-down list associated with an element field. The element is added to a new mapped element within the Element List. The Mapped Element has a cell position drop-down list. You must assign a position to the Mapped Element for it to become visible in the application.
- 3 Select a location from the cell position drop-down list for the mapped element and preview the results in the application viewer pane of the Application Builder.
- 4 Click Save to save the application with the new labels.

The mapped elements and grid element columns along with the resulting labels previews in the application viewer.

## Data types

Application data types enable you to further define and customize the application's data fields. You can select existing data types—such as integer, string, e-mail, phone, address, calendar, task, or contact—or you can create a new data type. You can establish a primary key, and specify column width.

❖ **Modifying the data type**

- 1 In the Application Manager, select an application listing from the detail view for which you want to modify the data type, and click Edit.
- 2 When the Application Builder displays, click Data Types on the toolbar.
- 3 In the Data Types window, identify the label you want and select a Data Type from the drop-down list.

- 4 In Primary Key, select the box if this is to be the primary key for your application.
- 5 In Column Size, enter the maximum width of the column number in characters, such as 24.
- 6 Click OK to save your entries and close the window.
- 7 In Application Builder, save the application.

❖ **Creating a data type**

- 1 In the Application Manager, select an application listing from the detail view for which you want to add a data type, and click Edit.
- 2 When the Application Builder displays, click Data Types on the toolbar.
- 3 In the Data Types window, identify the label you want, and expose its drop-down list.
- 4 Scroll to the bottom of the list and select Add New.
- 5 In New Data Type, enter the name of the new data type.
- 6 In Primary Key, select the box if this is to be the primary key for your application.
- 7 In Column Size, enter the maximum width of the column number in characters, such as 24.
- 8 Click OK to save your entries and close the window.
- 9 Save the application.

## **Finding and replacing applications**

This section describes how to search for applications and how to search for and replace broken applications.

### **Performing an application search**

Find Application lets you search for an application based on search parameters. The Find Application window displays the available search parameters in the left pane and the search results in the right pane.

**❖ Performing an application search**

- 1 Click Find on the Application Manager toolbar.
- 2 Enter search parameters. Select Advanced Search to display additional search parameters. Click Clear to remove the current search parameters. If you do not enter parameters, the search displays all applications with an active state and an approved status.
- 3 Click Search. The Search Results pane displays applications matching the search criteria you specified. To sort the results, click the heading of the Search Results column you want to sort.

The search results display in sets of 41 records. If there is more than one set of result records, the record sets are numbered and you can navigate to a specific record set by clicking the record set number that displays below the search results.

- 4 Select an application from the search results and click either Open or Preview. Open displays the application in the Application Builder and Preview displays the application content in a new browser window.

## Finding and replacing applications

As applications change, you may need to replace existing applications with newer versions or different application implementations. Use the Find/Replace Application window to perform this update.

This functionality is particularly useful to replace broken applications in your production environment.

---

**Note** Using Find/Replace for any application with CGI parameters is limited to finding the same application with a different version. For example, if application A has CGI parameters, you can find or replace another application with the same name, application A, but the application must have a different version to find it using Find/Replace.

---

**❖ Enabling find and replace application functionality**

Unwired Accelerator lets you enable the find and replace applications functionality for Mobile Web Studio users with the PortalAdmin role, as described in the *global.properties.xml* portal configuration file:

```
<Property name="PortalAdministrationRole"  
value="PortalAdmin" description="The J2EE role
```

required to administer the Portal performing export/import and update operations." menugroup="-1" />

- 1 To grant a Mobile Web Studio user the PortalAdmin role, you must have the StudioAdmin role.
- 2 Log in to Mobile Web Studio and select Administer | Organizations from the left pane.
- 3 Select the organization in which the user resides to whom you want to give find and replace access.
- 4 Select Users in the Organization Manager menu.
- 5 Right-click the user in the detail view and select Edit Roles.
- 6 When the Edit User Roles window displays, grant the user the PortalAdmin role, then click OK.

### ❖ Finding and replacing applications

To find and replace applications should follow this procedure.

- 1 Log in to Mobile Web Studio as a user with the PortalAdmin role.
- 2 When the Mobile Web Studio main window displays, select Build | Applications from the left pane, then click Replace from the toolbar above the detail pane.
- 3 In the Find/Replace Application window, click Select to the right of “Find.”
- 4 In the Find Application window, click Search. A list of applications displays in the Search Results window.
- 5 Select the application you want to replace, and click Add.
- 6 You return to the Find / Replace Application window with the Name, Version, and ID fields filled in.
- 7 Click Select to the right of “Replace.”
- 8 In the Find Application window, click Search. A list of applications displays in the Search Results window.
- 9 Select the application you want to replace the previously selected application, and click Add.

- 10 You return to the Find / Replace Application window with the Name, Version, and ID fields filled in.

---

**Note** Before executing the find and replace operation, UA checks whether the applications you specified are using CGI parameters. If the applications use CGI parameters, both the find and replace application must have the same name.

---

- (From) Active Catalogs – select this option to find and replace the specified applications in all active catalogs. Active catalogs in Portal Interface display applications that users can select and add to their own personal pages. If any active catalogs are modified with a new application, Portal Interface users catalogs are immediately updated.
  - (From) Non-Active Catalogs – select this option to find and replace the specified applications in all non-active catalogs.
  - (From) Portal Pages – select this option to find and replace an application for all page types—guest, default, and catalog.
    - Update Portal User Pages – when you select the previous option (Portal Pages), select this option to have Portal Interface user page groups automatically updated for all changed portal pages.
  - Agent’s Rule – agent rules evaluate an application’s content and determine if the agent should continue. Select this option to find and replace the application specified in the agent’s rule. Agents can use one application in the processing rule. If a criteria is selected for the rule, no application is included in the agent and no update is needed. See Chapter 7, “Using Unwired Accelerator Alerts,” for more information.
  - Agent’s Action – select this option to find and replace the specified application in the agent’s delivery action. Agents can deliver or write applications to a location. These applications are listed in the agent’s actions.
- 11 To view a log of the changes that would occur as a result of your selections, click Preview.  
Click OK.
  - 12 Once you click OK, the Details button is enabled. Click Details to see a detailed log of the changes that would occur:
  - 13 Once you preview the changes, click Replace to permanently implement the changes.

You see an alert message similar to the one that displays when you click Preview.

# Advanced Application Creation

This chapter discusses advanced functionality that you can implement in the elements and applications that you develop in Mobile Web Studio.

Topic	Page
Overview	101
Server-side click-across	102
Creating click-across applications	103
Building secure applications	127
Working with data-capable elements	130
Using grid rules	135
Creating chart applications	141
Creating application input fields	143
Linking applications	144
Linking parameters	152
Handling JSESSIONID	154
Adding JSP-related capabilities	155
Generalizing parameters with @OP tags	157
Custom capture options	159
Business Objects capture strategy	168
Remedy capture strategy	176
Enabling the Split window	182
Enabling Answers Anywhere (natural language search)	183
Enabling applications and pages to support multiple languages	196

## Overview

Unwired Accelerator lets you build interactive business application applications to solve multi-staged business problems. These applications use information from one application for continued processing in another application. When these applications are combined to interact and create collaborative functionality, the applications become a virtual application.

With UA, you can combine multiple services to build a composite application that consists of functionality drawn from a variety of sources within a service oriented architecture.

## Server-side click-across

Unwired Accelerator click-across functionality lets you create events that start when a user clicks the content in an application table. For example, a user can click a grid value in one application to refresh another application's content on the page using the grid value as a parameter to be applied to the refreshed application. You can define an event that takes its input value from any cell in the grid, not just from the cell in which the event is defined. The source cell can even be in a hidden column.

---

**Note** When you define click-across events you can place the event on a subset. For example, when you define a click-across event, you can enter “2-” in the Row field, which means that this event will be placed in every record from record 2 to the last record.

---

Server-side click-across lets you replace an application's content that is the source of an event with another application's content. You can use server-side click-across to build drill-down applications that display data at a lower level derived from one or more applications built from a source different than the source of the event-generating top-level application.

To display information from an underlying data store, and build a drill-down application that enables users to select a particular item for more information, the item's key can be a generated identifier used to look up the details.

---

**Note** You may want to display a more meaningful description of the grid item. To support this, build a grid containing the generated identifier in one column, and the description in another column. By hiding the first column, and building an event on the second column, you can create a compact application that enables you to easily drill down to the details section.

---

Server-side click-across is implemented in the Define Events wizard, which lets you create both client-side and server-side click-across events.

See “Using grid rules” on page 135 for more details on hiding grid data.

The *Mobile Application Development Tutorial* contains sample procedures that illustrate how to use this functionality.

## Creating click-across applications

This section describes two advanced application implementations—click-across messaging applications that share information with other messaging applications and applications that combine to form comprehensive interactive business applications.

A click-across messaging application shares information between itself and other messaging applications on the same portal page. When you define applications as messaging applications, they act together to establish a new form of Web application. See “Building click-across applications” on page 103 for instructions.

---

**Note** Click-across applications are for use with Portal Interface only. If you are working with mobile applications, see “Server-side click-across” on page 102.

---

## Building click-across applications

You create applications in Mobile Web Studio from a variety of content sources, including Web, XML, database, JSP, and Web services elements. Portal Interface supports applications communicating with each other when they are placed on the same Portal Interface page or different pages.

---

**Note** UA also supports applications communicating with each other when they are placed on different pages within the same page group. See “Building composite applications” on page 114 for details.

---

Mobile Web Studio includes a wizard-driven graphical user interface that allows you to build messaging applications without programming.

Messaging applications use a “click-across” mechanism, which uses the publish-and-subscribe model. Applications that communicate through a publish and subscribe paradigm require the sending application (publisher) to publish messages without explicitly specifying recipients or having knowledge of intended recipients. Similarly, a receiving application (subscriber) must receive only those messages in which the subscriber has registered an interest.

An application publishes notifications, which are received by a listening application subscriber. The publishing entity does not have to be aware of who or what has subscribed to its broadcasts. This allows for a flexible system where an application publishes multiple notices and a subscriber subscribes to multiple notices. Additionally, publishers can also be subscribers and subscribers can be publishers. An application can listen for an event while broadcasting another event.

When a user clicks an enabled link in an application with a defined event, the application broadcasts the event and passes along some data, such as the text on which the user clicked or the URL associated with the link. The parent page passes the event notification to applications on the same or other pages that have registered as listeners for that event.

For example, some applications represent a series of navigation steps that are recorded during application creation and executed as a series of HTTP GETs and POSTs during playback. If an application contains CGI parameters from any GETs and POSTs, an application’s creator can override the CGI parameters to place input boxes above the application when it is played back. These input boxes allow the person viewing the application to enter their own data values in place of the CGI parameters.

If an application’s content contains tabular data, you can specify input parameters (commonly referred to as CGI parameters) during the application creation process.

An application’s creator can override the CGI parameters to place input boxes above the application when it is played back. These input boxes allow the person viewing the application to enter their own data values in place of the CGI parameters.

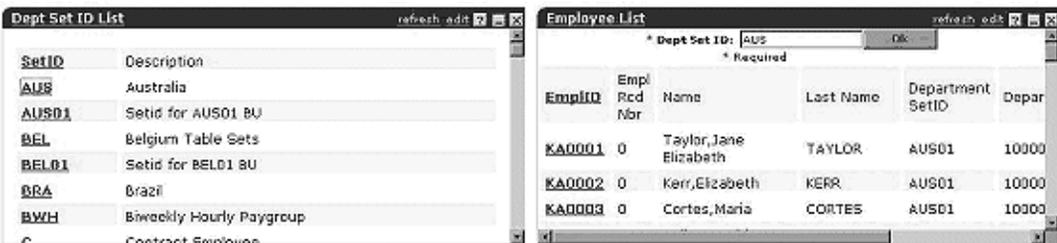
Figure 4-1 shows an application with the CGI parameter that corresponds to the Department Set ID. The value has been overridden and set to “AUS”.

Figure 4-1: Application input parameter



When used in conjunction with click-across applications, you can automatically populate data passed to the application from the input boxes from data in another application on the same, or different, page.

Figure 4-2: Applications exchanging data



The left application displays a list of Department Set IDs. The right application lists the employees for the given Department Set ID. When a user clicks the Department IDs in the left application, the selected ID is sent to the listening application on the right. Once the data is received, the listening application re-executes using the new data, and captures the resulting content.

**Note** Whether the listening application re-executes automatically depends on how you set the Auto Submit property in the Listener Editor when you define the application listener. If you selected Auto Submit when you define the listener, the listening application re-executes immediately. If Auto submit is not selected, the listening application does not re-execute instantly; click Refresh on the title bar to manually re-execute the listening application.

For example, if the user clicks the Belgium Department ID “BEL,” the string “BEL” is sent to the right application as input for the Department Set ID. The right application replays its capture sequence using the new query value of “BEL” and returns the result—the employee list for the Belgium department.

When developing click-across messaging applications, you are not limited to one parameter. As an application’s creator, you can override any number of parameters and set any number of those parameters to accept data from another application on the same page. Additionally, you can set any number of message events within an application, as explained later in this section.

## Creating click-across applications

---

**Note** This discussion of click-across applications uses the Web element application type for examples. However, you can create click-across messaging applications from any application type that has tabular data, a grid, or table structured output, and allows you to enter input parameters—JSP, XML, database, and Web services.

---

This section walks you through creating a click-across messaging application:

- 1 Creating a subscriber application
- 2 Creating a publisher application
- 3 Defining broadcast events
- 4 Assigning events to listeners
- 5 Adding applications to a page
- 6 Adding the click-across applications page to a page group
- 7 Testing the messaging applications

### ❖ Creating a subscriber application

- 1 Select Build | Applications from the Mobile Web Studio left pane, and click New.
- 2 When the Application Builder opens, click Add and select Web Element.
- 3 In the New Web Element window, in Location, enter:  
`http://www.marketwatch.com/tools`
- 4 Click Go.

- 5 In the Enter Symbol field, enter `SY` and click the arrow to submit your search query.
- 6 Click Next.
- 7 On the next page, click the stock name once. Click Next.
- 8 On the next page, move the mouse over the stock symbol `SY` and click to select. Click Next.
- 9 Click Select next to the second capture option, then click Next.
- 10 In the Configure Parameters window, click the Variable option next to the `symb` parameter and complete these options:
  - `symb` – select this option to create parameters for the stock quote.
  - Default Value – `SY`.
  - Type – Text Field
  - Required – select this option.

Click Next to see a preview of the captured data.

---

**Note** Verify that `SY` is the default value for the Symbol parameter. Parameters without default values may not work.

---

- 11 In the Element Name field, enter a name for the captured data; for example, `Stock Quote`. Click Finish.
- 12 In Application Builder, you see a working preview of the application.
- 13 To test overriding the CGI parameter, type a different stock symbol into the Symbol text box and click Update.
- 14 Click Save. You see the New Element Finish window. Complete these options:

**Content tab**

- Name – enter `Stock Quote`.
- Content type – select “text/html.”

Accept the defaults for the other options.

**Roles** Click Add All to move all roles from the Available Roles list to the Assigned Roles list.

**Presentation tab** Select these options:

- Iframe

---

**Note** Messaging applications must have the Iframe option selected.

---

- No Popup
- Editable

Leave the remaining options unselected.

Accept the defaults for the remaining options.

- 15 Click Finish to save the application.
- 16 Click OK in the confirmation pop-up.
- 17 Click Close.

#### ❖ **Creating a publisher application**

- 1 Select Build | Applications from the Mobile Web Studio left pane, and click New.
- 2 When the Application Builder opens, click Add and select Web Element.
- 3 In the New Web Element window, in the Location, enter:  
`http://finance.yahoo.com`
- 4 Click Go.
- 5 In the Yahoo Finance Web page, beneath the Market Summary heading, click the “Most Actives” link.
- 6 Click Next, then move the mouse around until the cursor is positioned over the Name column of the table. Click on the “Name” heading once to capture the table.
- 7 Click Select next to the top captured table option, then click Next.
- 8 On the Define window, click Next.
- 9 On the Filter window, click Next.
- 10 On the Configure Parameters window, click Next.
- 11 In the Preview window, give the element a name such as `Most Actives` and click Next. Then click Finish.
- 12 Click Save in Application Builder.
- 13 In the Finish window, verify your save options and settings:

**Content tab** •Name – enter a name for the application, for example Most Actives.

- Content Type – select “text/html.”

**Roles tab** click Add All to move all roles from the Available Roles list to the Assigned Roles list.

**Presentation tab** Select these options:

- Iframe

---

**Note** Messaging applications do not work unless Iframe is selected.

---

- No Popup
- Editable

Accept the defaults for the remaining options.

**Administration tab** Select Active.

Click Finish.

- 14 Click OK in the confirmation pop-up window.
- 15 Click Close to close Application Builder and return to the Mobile Web Studio main menu.

#### ❖ **Defining broadcast events**

---

**Note** Not all application types can have events defined; the application must have tabular data, a grid, or table structured output. If you try to create an event and nothing seems to happen, try using a different application type. Mobile Web Studio does not always issue a warning if the application is not capable of having an event defined.

---

- 1 Select Build | Applications from the Mobile Web Studio left pane.
- 2 On the Application Manager Status menu, click New to display new applications.
- 3 Locate the two newly saved applications (publisher and subscriber).

---

**Note** By default, an application has the “New” status until you right-click the application in the detail pane, select Approval Status from the pop-up, and select a different status.

---

- 4 Right-click the Most Actives application and select Define Events from the pop-up.

The Define Events wizard displays in a new window. Defining events is a three-step process. The first step shows you all grid, table, or list (structured) objects contained in the current application. You can define events only on structured (tables and lists) content.

- 5 In the first wizard window, click Select to choose the top table listed and click Next. The next window allows you to define or edit an event.

You can define an event for a row, a column, or any subset of rows and columns. You can also use the row and column number to identify a specific cell to define an event.

Enter the keyword “all” for Row and Column if you want to indicate all values in a row or column for an event.

Event Name is a logical name that must be unique across the entire portal and you must have separate names for distinct events. You can reuse the name for different cells to have the same event generated.

---

**Warning!** Defining multiple events on the same data causes invalid HTML. In other words, column 1 cannot have more than one event defined on it.

---

The differences between the types of data that can be sent in an event are:

- Cell – sends the text you click on as the event data.
  - Static – sends a constant value you specify in the text box to the right of the drop-down list as the event data.
  - HREF – if the text for which you are defining the event is a link, the underlying URL is sent as the event data.
- 6 Enter these values:
    - Row – all
    - Column – 1
    - Event Name – `stock_click` (note the underscore)
    - Select “Cell Value (this cell)” from the drop-down list
- a Select Client-side.
  - b Select:

- Notify Now – if you want the listener application to refresh immediately when you click the source application
  - Notify Across Page – if the listener application for this event is on a different page.
- 7 Click Add to add the event to your events list. After a short time, the first column should turn blue, which indicates there is an event defined for that column.
  - 8 Click Next to save the new event.
  - 9 When the Preview page displays, click Finish.
  - 10 Back in the Mobile Web Studio, right-click the Most Actives application and select Approval Status | Approved.

#### ❖ **Defining listeners**

This step lets you define a listener for the event to which an application is going to subscribe and listen. You can register an application to listen for one or more events. Each event to which an application listens must be assigned an application input parameter. You must remember the name of the event to which you want an application to listen.

- 1 On the Status Menu, click New, then right-click the Stock Quote application and select Define Listeners.
- 2 In the Event Name field, type `stock_click`, the name of the event you defined for the My Actives application. Verify that the Listener Param is set to “Symb.”
- 3 Select Auto Submit to have the publisher of the event call this listener and have the application re-execute immediately.
- 4 Click the plus sign to the right of Listener Param to add the listener event to the application. Once the plus turns into a check mark, click OK to save your changes. If you make an error, click the X icon to delete the Listener Param association, and repeat steps 1 and 2.
- 5 In the Mobile Web Studio, right-click Stock Quote application and select Approval Status | Approved.

#### ❖ **Adding messaging applications to a page**

To test the event you defined, you must add the new applications to a new page that can be viewed in Portal Interface.

- 1 In Mobile Web Studio, select Build | Pages from the left pane.

- 2 Click New on the Page Manager toolbar.
- 3 In the Page Builder window, click Add.
- 4 In the Search window that displays, click Search.

---

**Note** The application must have Approved status; otherwise, the applications do not display in the search results.

---

- 5 In the Results pane, hold down the Ctrl key and select the Most Actives and Stock Quote applications. When both are selected, click Add.
- 6 Click and drag the boxes in the right preview until the Most Actives application is on the left and the Stock Quote application is on the right.

The options for each application in the left pane are:

- Invisible – lets you hide a specific application from the end user. For example, you could set the display status of an application to invisible because the application is performing a server-side lookup operation; for example, transforming city names to postal codes.
  - Active – when this option is selected, which it is by default, the associated application is marked as active. Only applications marked active and approved display in Portal Interface.
- 7 Click Save As to save the page. Complete these fields:
    - Name – enter My Stock Quotes.
    - Type – select Default.
    - Active – this option is selected by default.
    - Roles – click Add All to assign all the available roles to the page.

Click OK to save the page.

- 8 After a dialog box tells you the page has been saved, click OK, then click Close to exit the Page Builder.
- 9 On the Page Manager Status menu, click New to locate the new page.
- 10 Right-click the My Stock Quotes page in the detail pane, and select Status | Approved.

❖ **Adding the click-across applications page to a page group**

Add the new page that contains the click-across messaging applications to a page group.

- 1 In Mobile Web Studio, select Build | Page Groups.
- 2 Click the New button on the Page Groups Manager toolbar.
- 3 When the Page Group Builder displays, click Add.
- 4 When the Search window appears, click Search.
- 5 In the Results pane, select the My Stock Quotes page and click Add.
- 6 In the Page Group Builder, click Save As.
- 7 When the Save Page Group window appears, complete the fields as follows:
  - Name – enter My Stock Quotes.
  - Type – select Default.

---

**Note** Pages must be of the same type as the page group to which you add them. For example, if you create a “default” type page group, you can only add pages to that page group that are also the “default” type.

---

- Navigation Style – accept the default.
  - Active – verify that this option is selected.
  - Assigned Roles – hold down the Ctrl key, select PortalAdmin and PortalUser, then click Add to move those roles to the Assigned Roles list.
- 8 Click OK to save the page group, and click OK again when a prompt verifies that the page group has been saved.
  - 9 Click Close to exit the Page Group Builder.
  - 10 Select New from the Page Group Manager Status menu in the Mobile Web Studio main window.
  - 11 Right-click the new page group, and select Status | Approved from the pop-up. Click OK when the page group saved confirmation displays.
  - 12 Select Approved from the Status menu.
  - 13 Right-click the My Stock Quotes page group select Update Users.
  - 14 When you see the message that the operation succeeded, click OK. Because the My Stock Quotes page group is the “default” type, all registered Portal Interface users can access this page group after the update operation.

15 Click Logout to exit Mobile Web Studio.

❖ **Testing the messaging applications**

1 Start Portal Interface by entering:

```
http://HOSTNAME.PORTALDOMAIN:4040/onepage/index.jsp
```

If you are using the UA Demo version, enter:

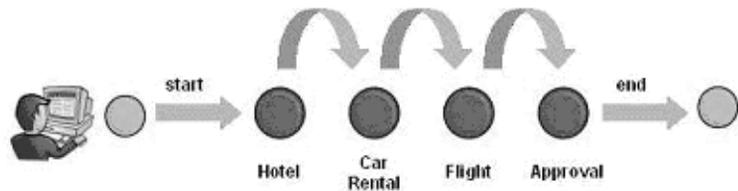
```
http://demo.sybase.com:4040/onepage/index.jsp
```

- 2 Log in using an existing account or create a new account using the Join Now link.
- 3 Select My Stock Quotes from the page group drop-down list on the far right of the Portal Interface window. You see the messaging applications.
- 4 To test the click-across messaging applications, select one of the stock symbols in the Most Actives application. The My Stock Quote Symbol input field updates with the stock symbol that you clicked on and updated stock information for the new symbol appears.

## Building composite applications

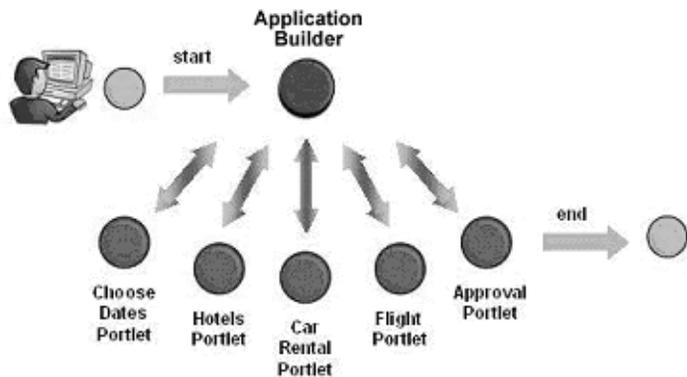
Composite applications bind existing applications into a common operation, sharing data between the applications and allowing them to operate as a virtual application. Application Manager and Composite Application Builder let you create composite applications in Mobile Web Studio. Composite applications can include individual applications on one page (a *application flow*) or serialized applications on multiple pages (a *page flow*), or for mobile applications, an application with tabs (a *tab flow*).

To arrange business travel, an employee must often interact with several Web sites that are independent of each other and maintain no shared context. For example, when arranging a trip, an employee must reenter travel dates into each individual Web site.

**Figure 4-3: Arranging travel at unrelated Web sites**

The employee follows a preferred order in which each travel element is arranged. For example, the employee reserves a hotel room, reserves a rental car, then reserves a flight. A composite application lets you combine these disparate tasks into a single application available at the portal. Each task in the process, such as reserving a hotel, is provided by an individual application.

The portal allows a user to personalize the presentation of the applications. A personalized presentation could include arranging commonly-used applications on one page and others on alternate pages. The Application Builder lets you enter the information relevant to each task.

**Figure 4-4: Arranging travel through a composite application**

Each application is separate from the other applications in the application, but the Application Builder displays them in a shared context.

## Shared context

To minimize the number of entries an employee has to make, a common view of individual tasks allows each task to share the same contextual view as the other tasks.

In this example, the travel dates are common to each application. The hotels application uses dates to check for the availability of rooms. The car rental application uses dates to check the availability of vehicles. The flights application uses the dates to check preferred airlines schedules. In each case, the user can override the dates if necessary.

In addition to the common dates, each application also has its own requirements. For example, the Hotel application requires the destination city, the Car Rental application requires the rental duration, and the Flights application requires the destination airport code. The user can enter each value directly into the application, automatically setting the context of the application request for each.

When constructing shared context, you identify the key fields that each of the component applications has in common, which allows the fields to be linked. For example, the application designer uses form processing in the Hotel application to examine the hotel reservation form. The designer is presented with a list of form fields with the relevant fields selected. The next step is to define events for the common relevant fields. After defining the events, the designer uses the Application Builder to assign events, as needed, to the fields of the Car Rental and Flight applications. The end result is a link between the common input fields of the forms in the application's component applications. When published to Portal Interface, the link between the form fields is used to automatically fill in the specified common fields of the other applications used in the application.

## Composite Application Builder

The Composite Application Builder lets you produce multi-application applications that are connected to one another using events or messaging. You lay out an application in an “application flow” or “page flow” for Portal Interface or a mobile application with tabs.

An application laid out with application flow displays all of its applications on one page. An application that uses page flow displays each row of applications on a different page.

### ❖ **Launching the composite application builder**

- 1 From the menu on the left select Build | Composite Apps.
- 2 In the Status menu, select New.
- 3 Click the down arrow to the right of the New button, and select either New Portal Application or New Mobile Application.

#### 4 The Composite Application Builder launches.

The left pane of the Composite Application Builder displays the applications and associated events that make up the application. The right panel displays a drag-and-drop workspace where you lay out and order the applications.

The top row buttons are:

- **Save** – displays the Application Properties window if the application has not been previously saved. If the application has been saved, this option saves the application without opening the properties window. Click the down arrow to the right of the Save option to select Save As and save the application under a different name.
- **Properties** – displays the Application Properties window where you can change the application’s flow type, assign roles, status, and specify whether or not the application is active.
- **Preview** – opens a new window that lets you preview the application. You must save the application before you can preview it.
- **Deploy** – sends the application to Portal Interface.
- **Template** – this button displays only for composite mobile applications.

The second row buttons are:

- **Add** – launches the Find dialog box, where you can select one or more applications to use in the application. The applications shown in the Results pane are approved applications. The applications you add display in the left pane of the Application Builder.
- **Post Process** – provides a list of post processing options that you can add to applications. See “Application post-processing options” on page 118.
- **Preview** – previews the application selected in the left pane application list.
- **Application Flow** – select one of these configurations in which to lay out application applications on one page:
  - 30% / 70% Layout
  - 50% / 50% Layout
  - 25% / 50% / 25% Layout
  - Full Layout

See “Selecting a page layout” on page 274 for more information on the layout options.

- Page Flow – select one of these configurations in which to lay out the application applications on different pages in the same page group.
    - 30% / 70% Layout
    - 50% / 50% Layout
    - 25% / 50% / 25% Layout
    - Full Layout
- See “Selecting a page layout” on page 274 for more information on the layout options.
- Tab flow – if you are building a composite mobile application, the Tab Flow button displays instead of the Page Flow button.

## Application post-processing options

The key to building virtual applications is the addition of postprocessing functions or filters to the application’s applications. Post-processing functions can either transform an application’s content (for example, render the content as a chart), or create events For example, you can transform an application’s content into a chart or you can create events that are triggered when a user accesses an application’s content. You can add more than one post-processing function to handle an application’s content and configure the order in which the postprocessing occurs. When you use multiple postprocessing functions, each function operates on the output of the previous function—similar to a Pipes and Filter design.

---

**Note** Pipes and Filters is a design architecture where the components are known as filters and the connectors are the pipes. Filters read input data, perform some computation or transformation on the data, and produce output data. Pipes serve as conduits for the data streams, passing outputs from one filter to the inputs of another filter.

---

The post-processing options are available by right-clicking an application in the Application Builder left pane:

- Define Events – launches an Event Definition wizard. See “Defining broadcast events” on page 109 and “Defining events” on page 120.
- Define Listeners – launches a Listener Definition wizard. See “Defining listeners” on page 120.

- Create Chart – lets you chart an application’s structured content. See “Creating charts” on page 121.
- Form Parser – lets you process form input and define which form parameters should be event default values or listeners for events from other sources. See “Adding form processing” on page 122.
- Create JSP Filter – lets you write custom code to process application content. See “Adding a JSP filter” on page 123.
- Text Processor – launches a text processor wizard that lets you define an event to occur when the user clicks on selected text. See “Processing text to trigger events” on page 124.
- Order Post Processing – lets you sequence an application’s postprocessing options at anytime. See “Ordering filters” on page 125.
- Invisible Status – lets you hide a specific application from the end user. See “Invisible status” on page 126.

All available post-processing modules are configured in the *uwp.xml* configuration file, which is located in *%SYBASE%\tomcat\webapps\onepage\fw\properties* if you are using Tomcat.

To add additional post-processing modules, use this interface:

```
</ApplicationWizards>
<PortletContentPostProcessorFilters>
  <FilterDef name="jspfilter"
    class="com.onepage.fw.uwp.apps.jspfilter.UWPJSPFilterApp"
    multiple_occurrence_allowed="true" use_app_object="true" desc="This is used
    to allow JSP code to operate on portlet content" />
  <FilterDef name="charts" class="com.onepage.fw.uwp.apps.charts.UWPChartsApp"
    multiple_occurrence_allowed="true" use_app_object="true" desc="This is used
    to allow JSP code to operate on portlet content" />
  <FilterDef name="formparser"
    class="com.onepage.fw.uwp.apps.formparser.UWPFormParserApp"
    multiple_occurrence_allowed="true" use_app_object="true" desc="This is used
    to allow JSP code to operate on portlet content" />
  <FilterDef name="clickacross"
    class="com.onepage.fw.uwp.apps.clickacross.UWPClickAcrossApp"
    multiple_occurrence_allowed="true" use_app_object="true" desc="This is used
    to allow JSP code to operate on portlet content" />
  <FilterDef name="clickacross-text"
    class="com.onepage.fw.uwp.apps.textprocessor.UWPTextProcessorApp"
    multiple_occurrence_allowed="true" use_app_object="true" desc="This is used
    to allow JSP code to operate on portlet content" />
  <FilterDef name="listener"
```

```
class="com.onepage.fw.uwp.apps.clickacross.UWPClickAcrossApp"
multiple_occurance_allowed="true" use_app_object="true" desc="This is used
to allow JSP code to operate on portlet content" />
</PortletContentPostProcessorFilters>
</Framework>
```

There is a method in the `UWPManager` `getPostProcessingFilters` that returns a list of configured filters.

## Defining events

Click the down arrow beside the Post Process button and select Define Events to launch the Event Definition wizard. When you define an event on an application, the application is saved under the same version, irrespective of what the versioning property is set to. See “Defining broadcast events” on page 109 for instructions on creating events.

Once an event has been defined on an application, all defined events are listed under the application name in a hierarchy.

## Event-naming convention

Events can have a three-part name:

```
[page_group_name] . [page_name] . [event_name]
```

However, a three-part name is not mandatory. You can also create a name with just `[event_name]`. The three-part name allows greater flexibility and registered listener name mapping.

The event name is matched with listeners that have their `[page_group_name]`, `[page_name]` registered along with the subscribed-to `[event_name]`. If an event name has only one part, all registered listeners are matched with that event name. If an event name includes a `[page_name]`, that event is matched to listeners with the same page name. The same rules apply to `[page_group_name]`.

---

**Note** Do not use spaces in event names.

---

## Defining listeners

You can add listeners only to applications with parameters. Applications with listeners must be saved with the `Iframe` attribute.

The Listener Editor lets you assign multiple parameters to event names. The application is saved under its current version number, regardless of whether the global versioning property is set to “on”.

❖ **Creating a listener**

- 1 Click the down arrow beside the Post Process button and select Define Listeners to launch the Listener Editor.
- 2 When the Listener Editor displays, complete these options:
  - List of Events – select the event to which you want this listener assigned.
  - Event Name – select an event from the List of Events drop-down list, or enter a name manually in this field.

---

**Note** Do not using spaces in event names.

---

- Listener Param – select the overridden CGI parameter that you want to associate with the selected event.
  - Auto Submit – when Auto Submit is enabled, a listener application re-executes when it receives an event. If Auto Submit is disabled, the application does not re-execute immediately.
- 3 Click the plus sign to add the association with the selected event or the X sign to remove any associations that have already been made.

Once you assign one or more parameters to event, parameter-to-event mapping is listed under the application name in the Application Builder in a hierarchy.

## Creating charts

Click the down arrow beside the Post Process button and select Create Chart to launch the Chart Definition wizard that is launched by right-clicking an application in the Application Manager.

When you define charts for an application, the application is saved as the same version, regardless of whether the global versioning property is turned “on.” Once you define a chart, the chart is listed under the application name in a hierarchy.

The *Mobile Application Development Tutorial* contains examples that demonstrate how to create charts.

## Adding form processing

Form processing allows you to preview a form, identify input fields, assign or create an event name, and assign default autofill values to the form's fields.

### ❖ Parsing a form for post-processing

- 1 Log in to Mobile Web Studio and select Build | Applications from the menu in the left pane.
- 2 In the detail pane, right-click the application to which you want to add post-processing, and select Edit from the pop-up.
- 3 When the Application Builder displays, right-click the application on which you want to parse a form and select Form Parser from the pop-up. The Form Parser window appears with a preview of the application.
- 4 Click Next. The Form Input Definitions window displays.
- 5 Click the Variable column for each form variable you want to be an event, default value, auto-filled, or listener for events from other sources.

Once you select a variable, fields and drop-down lists display for that variable.

- 6 Complete these options for each selected variable:
  - Event Name – this field lets you define an event on-the-fly. Type in the name of the event for which you want this field to be default value.

---

**Note** Do not using spaces in event names.

---

- Default Value – enter the event's default value.
  - Type – this column displays the field type. To change the field type, select a different type from the drop-down list.
  - Personalize – if the field can be filled in using an auto-fill adapter, select the adapter from the drop-down list.
  - Key – if the field has a key values, select the appropriate key from the drop-down list.
  - Notify Now – selection this option to send any event immediately.
  - Notify Across Page – select this option to send the event across pages if there is a listener on another page; otherwise, look for events on the current page. for this example, do not select this option.
- 7 Click Finish.

## Adding a JSP filter

JSP filters allow you to write custom code to post-process application content using programming. When you create JSP filters, you look at an application's content, enter the Java code to transform the content in any way you want, then save the code as a JSP template, which you can later select for post-processing.

### ❖ **Selecting a JSP filter**

- 1 Log in to Mobile Web Studio and select Build | Applications from the menu in the left pane.
- 2 In the detail pane, right-click the application to which you want to add post-processing and select Edit from the pop-up.
- 3 When the Application Builder displays, right-click the application to which you want to add a JSP Filter and select Create JSP Filter from the pop-up.

The JSP Editor appears with a preview of the application.

- 4 Complete these options:
  - JSP Template – enter the name of an existing JSP template you want to use or click Select to search for a template using the Find JSP Template dialog box.

Click Search to display existing approved JSP templates. When the list appears in the Results pane, select:

- New – to create a new JSP template.
- Edit – to modify the template selected in the Results pane.
- Select – to choose the template selected in the Results pane.
- Preview – to preview the template selected in the Results pane.

- Apply JSP Filter to XML Playback – select this option to apply the filter only when the application’s content is returned as XML. When this option is not selected, which is the default, the filter is applied only when the application’s content is returned as HTML.

---

**Note** When applications are played back, they return their content in either HTML or XML. Whether HTML or XML is returned is determined by the entity making the request.

For example, when Portal Interface requests that application A be played back, the output is requested as HTML. When an agent requests that sample application A be played back, the output is requested as XML. When you create a JSP filter and attach the filter to an application for postprocessing, the developer uses this option on a case-by-case basis to apply or not apply the JSP filter during application playback.

---

- 5 Click Finish to associate the JSP Filter with the selected application.

You can add more than one JSP filter as post-processing elements on an application and you can apply the same JSP filters to numerous applications. When the JSP filters have been added, the application is saved under the same version, regardless of how the global versioning property is set. Once you assign a JSP filter for application postprocessing, the filter is listed in the Application Builder under the application name.

## Processing text to trigger events

The Text Processor wizard is similar to the Event Definition wizard and allows you to create events that are triggered when a user clicks on the text you associate with the event.

### ❖ Post-processing text to trigger events

- 1 Right-click any application in the Application Builder Applications list, click Post Processing, then select Text Processor.
- 2 When the Text Processor wizard displays, you see a preview of the selected application.

Click Next. You see the window where you define click-across events.

- 3 Complete the options as follows:
  - Regular Expression – enter a regular expression to place an event on text that matches the expression.

- Event Name – the name of the event to generate when the matching text is clicked. The event must have already been created.

---

**Note** Do not using spaces in event names.

---

- Value drop-down list – select the type of data to send with the event:
    - Dynamic Value - sends the actual text you click on as the event data.
    - Static Value - sends a constant value as the event data. You specify this value in the text box to the right of the value drop-down list.
  - Matching Scheme – select the matching scheme to use when finding matches for your regular expressions. Select:
    - Match All Occurrences – to place the event on every match to your regular expression that is found.
    - Match First Occurrence Only – to place the event on only the first occurrence of a match to your regular expression.
    - Notify Now – send the event immediately. For this example, select this option.
    - Notify Across Page – send the event across pages if there is a listener on another page; otherwise, look for events on the current page.
- 4 When you are satisfied with your entries, click Next.
  - 5 When you see a final preview of the application, click Finish. You return to the Application Builder.

## Ordering filters

Click the down arrow beside the Post Process button and select Order Post Processing to launch a wizard that lets you order and re-order an application's post-processing filters.

## Invisible status

Click the down arrow beside the Post Process button and select Invisible Status to make the selected application Visible or Invisible. The default is Visible. When you make an application Invisible, the end user will not see it on the page.

For example, you could set the display status of an application to invisible because the application is performing a server-side look up operation that the end user does not need to see, such as transformation (from city name to postal code, currency conversion), look ups, polling queues or a table for availability of data.

## Application runtime entities

This section discusses the runtime entities that must be generated to execute the application.

### Application flow

If the application is laid out for application flow, the application's applications are positioned on one page at portal runtime.

Click Application Flow to lay out an application's applications on one page. Select one of the four page layouts that are available. Typically, you assign applications to the layout cells from left -to-right and top-to-bottom, which reflects the natural flow of information. Once the layout is done, click Preview to view how the applications will display on a Portal Interface page.

### Page flow

If the application is laid out for page flow, the application's applications are positioned on different pages at portal runtime. To lay out applications on a page, click Page Flow. You assign applications in the cells of the page-flow layout from top-to-bottom reflecting the natural flow of information. In Portal Interface, the pages show from left-to-right. Once the layout is done, click Preview to see how the pages will display in a Portal Interface personal page group.

---

**Note** To use Page Flow, the <page.group> property in the *global.properties.xml* file must be set to "on."

---

## Application properties

A new application's associated pages and page group status is always saved as Approved. Although you can set the status to a different state, you can deploy an application only when it has the Approved status.

When you click Properties in the Application Builder and are saving an application for the first time, you see the Application Properties window. Once you save the application and click Properties, the roles you have assigned appear in the Assigned Roles list and the system-assigned ID displays.

---

**Note** You can change an application's status only from the Application Properties window.

---

If you change the Type to Page Flow, an additional option—Page Naming—displays. The two naming options are Use Application Name and Use Application on Page. When the pages are created for Page Flow, the pages names are determined by this setting.

- Use Application Name – names each page in the format:  
`_[Application Name]_[Page number]`  
For example, `_StockApp_01`, `_StockApp_02`, and so on.
- Use Application on Page – uses the application name from the page in the format:

`_[Application Name]_`

---

**Warning!** Saving applications with the Page Flow type and the Page Naming option set to Use Application Name can overwrite a previous application's page if the names are the same.

---

## Building secure applications

Use the Secure Application Builder to create and edit Web elements captured from secure sites (URLs that begin with `https://`). Select New Secure Application from the New drop-down menu or the right-click menu in the Application Manager to launch the Secure Application Builder.

The primary type of secure implementation on the Internet is SSL version 3. The portal handles SSL3 using the Java Secure Sockets Extension (JSSE) implementation, which is included in Unwired Accelerator. UA does not work with SSL2 sites.

---

**Note** The portal handles secure applications in a catalog completely differently. See Chapter 8, “Building Catalogs,” for more information.

---

## Setting up Unwired Accelerator to create secure applications

This section describes the steps required to set up Secure Sockets Layer (SSL) and to turn on portal security parameters that enable the creation of secure applications.

### ❖ Setting up SSL for secure application creation

Perform this procedure only if you are using EAServer as your application server— it is not necessary if you are using Tomcat.

- 1 Copy *jnet.jar*, *jcet.jar*, and *jsse.jar* from  
%SYBASE%\EAServer\WebServices\java to  
%SYBASE%\EAServer\java\lib.
- 2 Modify %JAGUAR%\bin\serverstart.bat as follows:
  - a Add these lines at the beginning of Set EAS\_CLASSPATH\_P1:

```
set
EAS_CLASSPATH_P1=%EAS_CLASSPATH_P1%;%JAGUAR%\java\lib\jsse.jar
set
EAS_CLASSPATH_P1=%EAS_CLASSPATH_P1%;%JAGUAR%\java\lib\jnet.jar
set
EAS_CLASSPATH_P1=%EAS_CLASSPATH_P1%;%JAGUAR%\java\lib\jcet.jar
```

- b Add these lines at the beginning of Set EAS\_BOOTCLASSPATH\_P1:

```
set EAS_BOOTCLASSPATH_P1=%EAS_BOOTCLASSPATH_P1%;
%JAGUAR%\java\lib\jsse.jar
set EAS_BOOTCLASSPATH_P1=%EAS_BOOTCLASSPATH_P1%;
%JAGUAR%\java\lib\jnet.jar
set EAS_BOOTCLASSPATH_P1=%EAS_BOOTCLASSPATH_P1%;
%JAGUAR%\java\lib\jcet.jar
```

- 3 In a text editor, open *java.security*, which is located in `%SYBASE%\Shared\jdk1.3.1_06\jre\lib\security`, and add this line:

```
security.provider.2=com.sun.net.ssl.internal.ssl.Provider
```

#### ❖ Enabling portal security

The final step in setting up the portal to create secure applications is to enable the portal's security parameters in the *global.properties.xml* master configuration file.

- 1 In a text editor, open *global.properties.xml*, which is located in the `%SYBASE%\tomcat\onepage\config` subdirectory of your installation if you are using Tomcat. If you are using EAServer this file is located in `%JAGUAR%\EAServer\Repository\WebApplication\onepage\config`.
- 2 In the Global Properties group, locate the “secure” property and set the value to “on”:

```
<Property name="secure" value="on" description="(on/off). on if HTTPS is
enabled for the web-container. HTTPS will be used for login and
registration" menugroup="-1" />
```

- 3 Optionally, you can also set the `turnOffSecurePassword` property to “true” so you are not always prompted for another password.

```
<Property name="turnOffSecurePassword" value="@SECURE_PASSSWD_OFF@"
description="Turn on/off the usage of the secure password prompting by
PI when using http portlets created from Studio. Default is on"
menugroup="-1" />
```

- 4 Save your changes in *global.properties.xml* and exit the text editor.

## Creating secure applications

#### ❖ Adding a secure application

- 1 Select Build | Applications from the Mobile Web Studio left pane.
- 2 Right-click in the Application Manager detail view, and select New Secure Application to invoke the Secure Application Builder.
- 3 In the Application Builder, click the Add icon down arrow and select Web Element.
- 4 When the New Web Element window appears, enter the URL for a secure Web site (for example, *https://www.delta.com*), then click Next.

- 5 Continue with the usual creation of the Web application. See “Creating Web applications” on page 18.

---

**Note** Unlike other applications, when you save a secure application, it does not automatically appear in the Application Manager detail view when you select the New status due to protocol restraints. To view the application in the detail view, select a different status, then reselect New. The new secure application listing should be visible now.

---

If you try to share a page with secure content in Portal Interface, you see a message similar to this:

```
To protect your personal information, Unwired
Accelerator allows you to only send pages that do not
contain secure applications. Please share one of your
other pages, or remove any secure applications from this
page and try "Share Page" again.
```

Secure content applications cannot be shared in Portal Interface.

---

**Note** Sybase recommends that you do not mix insecure and secure applications on the same page.

---

## Working with data-capable elements

Data-capable elements conform to the Mobile Web Studio internal data format. The element types that can be data-capable are Web, XML, and database. The wizards for data-capable elements allow you to add split, define, and filter rules on the element content. The output of data-capable elements enables additional capabilities, such as assignment of application labels to element fields and the formatting of HTML output with cascading style sheet class definitions.

Database elements are data-capable by default. Web elements are data-capable when the visible text can be assigned a tabular structure. Data-capable options appear in the Web Element wizard as unformatted tables with a grey background. If the data-capable option does not appear, select an option and click Gridify. This forces the visible text in the element into a table that is then data-capable.

**❖ Creating data-capable Web element content**

- 1 Select Build | Applications from the Mobile Web Studio left pane.
- 2 Click the New icon on the Application Manager toolbar.
- 3 When the Application Builder displays, right-click in the left pane and select New Web Element.
- 4 Enter `www.yahoo.com` in the Location field and click Go.
- 5 Click Next.
- 6 Click any list.
- 7 When the captured options display, select the second Select option and click Gridify. You see your selection displayed as a table.

---

**Note** When you apply a UI XSL template to a Web application, you must select the “Enable Grid Rules” box, otherwise, the specified template is not applied.

---

XML elements are data-capable if the XML format conforms to the internal Mobile Web Studio Document Type Definition (DTD). If the XML does not conform to the Mobile Web Studio DTD, a content XSLT can be used to perform an XML to XML transformation that converts the source XML to the correct XML format. Data-capable XML elements have a default HTML presentation and do not require an UI XSLT. You can override the default HTML presentation by specifying an UI XSLT, alone or in combination with a content XSLT.

**❖ Creating data-capable XML element content**

- 1 Select Build | Applications from the Mobile Web Studio left pane.
- 2 Click the New icon on the Application Manager toolbar.
- 3 When the Application Builder displays, right-click in the left pane and select New XML Element.
- 4 When the XML Element Definition window appears, enter the following input as an example:
  - XML URL –

`http://p.moreover.com/cgi-local/page?c=Biotech%20news&o=xml`

- Content XSLT – click Select. When the Search window displays, click Search, select any XSL template you have created from the Results pane, and click Select.
  - UI XSLT – `pink_ui_xslt`
- 5 Click Next, then click Finish.
  - 6 When you return to the Application Builder, click the down arrow beside the Save button to save the new element. See “Saving applications” on page 76 for specific instructions.

## Data-capable post-processing options

When you create any data-capable element except an HTML element, options appear during application creation that allow you to isolate specific pieces of element content on which you can perform these post-processing options:

- Adding split rules – allows you to split columns or rows using commas, spaces, and so on. The `SplitEnabled` parameter must be set to “true” in *global.properties.xml*.
- Defining record layout – allows you to define the record layout and specify a record header.
- Applying newly created filter rules – allows you to add rules to filter the records so only the specified data displays.

## Adding split rules

Split rules allow you to add parameters by which to split the columns or rows in the table. The Add Split Rules options are:

- Split – split the table by All Rows, a specific Row No. (enter the row number in the text box after the Split drop-down list), All Columns, or a specific Column No. (enter the column number in the text box).
- Split by Delimiter – split the rows or columns by Line Feed, Space, Comma, or Other. When you select Other, specify the delimiter to use in the text box to the right of the delimiter drop-down list.

- Personalization – enabled when you select the Variable option. Personalization adapters retrieve values from external systems or databases and automatically submit them to the application. Personalization adapters must be registered with Mobile Web Studio before they appear in the Personalization drop-down. Once you select a Personalization adapter, the available methods within the adapter display in the drop-down. Select a method to specify the parameter within the Personalization adapter to submit to the application. See

To remove a split rule, select the rule in the Current Split Rules section and click Remove. Mobile Web Studio removes the highlighted rule. To remove multiple rules, press the Ctrl key while you select rules and click Remove.

## Defining record layout

Record layout allows you to define the format of grid elements. When you have captured an object that you have formatted as a grid (table), click Next from the Add Split Rule window.

The Define Record Layout options include:

- Record Layout – specify record orientation; column- or row-based.
- Records Contain Labels – select if the records do or will contain field labels.
- Labels Are Displayed in Record – displays when you select Records Contain Labels. Enter the record number in which the labels display.

Click Preview to see the result of defining the record layout.

- Transpose rows and columns – change the layout by transposing the placement of rows and columns.

## Adding filter rules

Click Next on the Define Record Layout window to the Add Filter Rules window, which allows you to create basic filters to include or exclude records and fields.

The Add Filter Rule options are:

- Include/Exclude – select the type of Filter Rule to create:
  - Include Records
  - Exclude Records

- Include Fields
- Exclude Fields
- Position/Condition (record rules only) – select:
  - Number – to specify the position of the record to filter. Enter the numbers or ranges of numbers for the filter rule.
  - Where – to specify a condition for the filter rule. Select the field where the condition should be met.
- Select the Boolean argument for the filter rule.
- Enter the value for the filter rule in the last input field.

Click Add to add the filter rule. The preview highlights any records or fields that match the filter rules.

- Position/Label (field rules only) – select:
  - Number – to specify the position of the field.
  - Where – to specify a condition for the filter rule.
  - Label – to include or exclude a field specifically by the contents of its label.

---

**Note** Editing filter rules can alter the structure of a table by adding, excluding, and moving columns or rows. Such changes can impact features that were dependent on the previous table structure, such as events and charting. After changing filter rules you should check the dependencies to verify the application still works as expected.

---

❖ **Applying newly created filter rules**

Continue to use the data-capable Web element example used in the previous section.

- 1 Select Include Records from the Include/Exclude drop-down.
- 2 Select Number from the Number/position drop-down.
- 3 Enter “1– 4” in the Number field. When you click Preview, records 1– 4 are highlighted in the preview.
- 4 Click Next. The element preview displays only those records that are defined by the filter rules.

- 5 Enter the name under which you want to save the element in the Element Name field, and click Finish.
- 6 When you return to the Application Builder, click the down arrow beside the Save button to save the new element. See “Saving applications” on page 76 for specific instructions.

## Using grid rules

Unwired Accelerator enables you to create user-defined rules that alter the appearance and content for application elements that generate table-based content (such as a database element or a Web element that is captured as a grid). For example, you can use rules to filter unwanted records or fields, and to display column header information.

Additional rules allow you to add new records or fields, and alter record and field content. You can take existing content, for example, from a remote Web site or a database, and alter the presentation to a different format.

---

**Note** Grid rules provide more options for altering either the layout or content of the original source data, without changing the source. However, if the source is external or provided from an application that cannot be changed, this functionality may not work.

---

Variations on include rules enable you to include records or fields, but keep them hidden from view. You can use hidden records or fields to define events, and use the content of the hidden records or fields in calculations or in a visible column.

Grid rules may be useful in situations such as these:

- Executive dashboard – create a concise financial report from general data. For example, use a Web page of financial report data that targets a wide audience as the source, create rules that eliminate unnecessary rows and columns, and add “bottom-line” information for an executive audience.
- Aggregate data – create aggregate data from various raw data sources. For example, use raw data accessible in a database as the source, and create rules that combine and present the data in a form that is more useful to a particular audience.

- Graphical presentation – create a chart or Flash presentation of data. For example, use an existing Web report as the source, and create rules that restructure the report for graphical presentation without manipulating the data. The grid rules provide flexibility for extracting data that can be used in the charting layer.

When defining grid rules, you should keep in mind that grid rules are processed by type, and in this sequence:

- Split rules
- Insert rules
- Edit rules
- Filter rules

## “include” options

When you capture or retrieve grid-based data, the program uses a default “include” logic that assumes you want to include all of the data you captured or retrieved. This logic remains, even when you specify to exclude records or fields, insert record or fields, or edit records.

However, when you select any “include” option from the first Add Filter Rule drop-down list, the default “include all” no longer applies, and you must explicitly specify the records to include.

Therefore, to hide records or fields from the display, first use Include Records to specify the records to include, then use Include Hidden \* to specify the record or field to hide.

---

**Warning!** If you select Include Hidden Records without explicitly specifying the records to include, the final preview of the portlet is blank. Because you have not explicitly stated which rows of data to include, the only data that is actually there is hidden, resulting in a blank display.

---

## Using grid rules

This section describes grid-rule functionality that is useful for creating mobile applications. Topics include:

- Insert rules – hidden records or fields

- Edit rules – images and values
- Using @OP tags with grid rules

### Insert rules – hidden records or fields

When inserting either records or fields, you can specify to insert a hidden record (or field). Such an insertion marks the record (or field) as hidden, so it does not display in the application. However, its contents can be used for event definitions.

### Edit rules – images and values

When you are editing records, there are two options for defining content of target cells:

- Specify an image to be inserted into the cell.
- Specify an explicit value, or a value based on a formula, that can be applied to a range of cells in the grid.

When selecting an image type, you can enter a path to the image relative to the application server hosting Mobile Web Studio. For example, for the *icon\_arrow.gif* under the *images* directory in the onepage Web application directory, enter: */onepage/images/icon\_arrow.gif*. You can also specify a full HTTP URL reference.

When selecting a value type, you can use a syntax that allows you to specify ranges of existing cells, and perform functions (nested if required) on these cells. The syntax is:

```
<command>(args [, ...])
```

where *<command>* is the operation to perform, and *args* are the arguments to use.

Commands include:

- *sum* – sums the arrays of cells specified.
- *avg* – calculates the average of arrays of cells.
- *int* – returns the integer portion of argument.
- *diff* – determines the difference between two arguments.
- *div* – divides two arguments.
- *concatenate* – concatenates all the specified cells.

- `concatenate2` – concatenates all the specified cells, inserting pad characters.
- `min` – determines minimum of specified arrays of cells.
- `max` – determines maximum of specified arrays of cells.

---

**Note** These commands are described in detail in the sections that follow.

---

Arguments include:

- `Scalar` – specifies a single-element value (either some literal text, or cell content).
- `Array` – specifies a range of cells, which can be one- or two-dimensional.

To specify single-cell content, use this syntax:

`$R<x>F<y>`

where `<x>` and `<y>` are indexes into the cells. For example, `$R4F5` specifies record 4, field 5.

You can use the special value 0 as an index to indicate the current value. For example, if you are editing record 6, field 9, `$R0F9` refers to record 6, field 9. This is especially useful for rules that edit a number of records or fields at once; it enables you to use a single rule to perform the same operation on a range of cells.

Array arguments use the single-cell syntax, combined so that a range of cells is specified. That is:

`$R<x>F<y> [ :R<x'>F<y'> ]`

If the second cell specifier is not present, the array consists of a single cell; if the second cell is specified, the array consists of all the cells in the indicated range. For example, `$R5F2:R6F3` indicates records R5F2, R6F2, R5F3, R6F3. Again, use 0 as an index to use the current record or field.

---

**Note** If a function is expecting an array argument, a scalar is accepted and treated as a single cell array. However, if a function is expecting a scalar argument, an array argument cannot be accepted.

---

### ***sum***

Description

This function sums all arguments, and returns a single result.

**Arguments** There can be one or more array arguments. Summation is carried across all the arrays specified (or, if scalars, by simply adding the scalar value to the current summation). Non-numeric values are ignored. The result is returned as a floating point value.

**Examples**

```
=sum($R3F4:R7F6)
=sum($R6F4, $R10F8)
=sum(5, $R6F7)
```

### **avg**

**Description** This function sums all arguments, divides the result by the number summed, and returns a single result.

**Arguments** There can be one or more array arguments. Summation is carried across all the arrays specified (or, if scalars, by simply adding the scalar value to the current summation). Non-numeric values are ignored, and not added to the count of items summed. The result is returned as a floating point value.

**Examples**

```
=avg($R3F4:R7F6)
=avg($R6F4, $R10F8)
=avg(5, $R6F7)
```

### **int**

**Description** This function returns the integer portion of a single scalar argument.

**Arguments** There can be only one scalar argument.

**Examples**

```
=int(2.5)
=int(=sum($R6F4:R10F8))
```

### **diff**

**Description** This function subtracts the second argument from the first argument.

**Arguments** There can be only two scalar arguments.

**Examples**

```
=diff($R3F4, $R7F6)
=diff($R3F4, $R7F6)
=diff(5, 3)
```

### ***div***

Description This function divides the first argument by the second.

Arguments There can be only two scalar arguments.

Examples

```
=div($R3F4, $R7F6)
=div($R6F4, 5)
=div(10, 2)
```

### ***concatenate***

Description This function concatenates all arguments, and returns a single result.

Arguments There can be one or more array arguments. Concatenation is carried across all the arrays specified (or, if scalars, by simply concatenating the scalar value to the current result).

Examples

```
=concatenate($R3F4:R7F6)
=concatenate($R6F4, $R10F8)
=concatenate(5, $R6F7)
```

### ***concatenate2***

Description This function concatenates the second and subsequent arguments, using the first argument as a pad, and returns a single result.

Arguments The first argument is scalar, and specifies the pad string to be used; there can be one or more subsequent array arguments. Concatenation is carried across all the arrays specified (or, if scalars, by simply concatenating the scalar value to the current result). The pad value is inserted between all concatenated values (but not appended or prefixed).

Examples

```
=concatenate2(abc, $R3F4:R7F6)
=concatenate(-, $R6F4, $R10F8)
```

### ***min***

Description This function determines the minimum value across all the arguments, and returns a single result.

Arguments There can be one or more array arguments. Determination is carried across all the arrays specified (or scalar values, if used). Non-numeric values are ignored. The result is returned as a floating point value.

Examples

```
=min($R3F4:R7F6)
```

```
=min($R6F4, $R10F8)
```

```
=min(5, $R6F7)
```

**max**

## Description

This function determines the maximum value across all the arguments, and returns a single result.

## Arguments

There can be one or more array arguments. Determination is carried across all the arrays specified (or scalar values, if used). Non-numeric values are ignored. The result is returned as a floating point value.

## Examples

```
=max($R3F4 :R7F6)
```

```
=max($R6F4, $R10F8)
```

```
=max(5, $R6F7)
```

**Using @OP tags with grid rules**

Another useful feature is the ability to use @OP tags when defining grid rules. At runtime, the @OP tag is replaced with a value passed to the application as an input parameter. @OP tags are useful for building SQL queries, because they enable an application to use queries with parameters. Several tutorials in the *Mobile Application Development Tutorial* use the @OP parameter tag with SQL queries.

You can use the @OP tags in grid rules enabling applications to use input parameters as a way of altering the runtime behavior of the rules associated with the application.

## Creating chart applications

Unwired Accelerator allows you to create graphical charts using data present in existing applications. When you chart an application, UA creates an image from the parameters you specify, then displays the data in a graph that is easy to read and analyze.

The charting application wizard lets you chart applications without programming. Charting applications lets you create graphical representations of tabular data. Data can display in one of three chart formats—bar, pie, or line. You can create a chart for applications containing any element type that has input parameters and a grid or table with structured output.

You can chart any number of variables, as long as the data is contained in a tabular form (a grid or table).

Unwired Accelerator allows you to create drill-down charts using the charting post-processing wizard. Drill-down charts use server-side click-across to link multiple charts together into a cohesive application.

To create one drill-down charting application from two chart applications, there must be a relationship between the category or series names of the first chart and the parameters used in defining the second chart's data source.

See the *Mobile Application Development Tutorial*, which contains a tutorial that illustrates the use of drill-down charts using server-side click-across. The tutorial contains procedures to create a table of data with the stock market's most active stocks and uses drill-down charting to display the quote page with details of a selected stock.

---

**Note** Currently, BlackBerry devices do not support CGI parameter charting. Also, for a charting portlet with a Client Side Click Across listener, the listener must be defined after defining the chart or the click-across link will not work.

---

### ❖ Removing a chart definition

Follow this procedure if you want to remove a chart definition.

- 1 From the left menu, select Build | Composite Apps.
- 2 From the Status menu, select New, then click New to deploy the Composite Application Builder.
- 3 Click Add to add the chart application you want to remove the chart definition for.
- 4 In the list of applications, there is a sub-tree of post-processing definitions on the application, including one for the chart and any events you placed on it.
- 5 Click the X icon to the right of the post-processing definitions to remove them from the application. In addition to the events, you must remove the chart itself.

- 6 Close the Composite Application Builder.

## Creating application input fields

The drop-down /select list functionality enables you to predefine input values for application input parameters. When an application requires that a user enter a value before receiving a result, the user can select one of several predefined values from a drop-down list.

You can convert the select lists and grid application input fields to drop-down lists.

### ❖ Defining application input fields

- 1 In the Application Builder, click Params to launch the Configure Parameters window.

---

**Note** The Params button is only activated if you are editing or creating an application containing CGI parameters.

---

- 2 The Configure Parameters window shows the list of active parameters. Click Next.
- 3 In Unique Configurable Parameters, from the Display Type drop-down list, select from:
  - None
  - Text Field
  - Calendar Date
  - Password
  - Select
  - Text Area
  - Boolean
  - Grid
  - Linked
- 4 If you select the Select or Grid or Linked display type, Add and Clear buttons appear. Click Add launch the Edit Default Value window.

Click Clear to clear the default values.

- 5 In the Edit Default Value window, define the Display Name and the value that gets passed in as the CGI parameter value when the Display Name is selected by the user. Click Close.
- 6 In the Configure Parameters window, click Save.

## Linking applications

Unwired Accelerator enables you to link applications. For complex transactions, you can link to multiple applications, such as insert, update, delete, verification, and login applications. You can order the operation sequence, and can order the menu level.

## Complex transactions

You can link applications to support complex transactions. These linked applications must be related and processed as such by both the server code and the client user interface. To accomplish this, each of the linked new components is associated with a level attribute.

This example illustrates the concept of level attributes:

A Customer Account application is associated with Contacts and Opportunities. Contacts and Opportunities do not exist without an Account. In terms of database entity-relationships, Account (ID) is a foreign key for Contact and Opportunity creation. Users are required to first create an Account, then create Contacts and Opportunities, which must then be associated with the Account created.

In this example, Account creation is a Level 1 operation and Contacts and Opportunities are Level 2 operations. Results (key) from Level 1 (new Account ID) are used in Level 2 operations (Account ID is a foreign key for Contact/Opportunity creation).

In SAP applications, when you link a new application, Mobile Web Studio shows it as a Level 1 application then silently queries the application to determine if it has input tables. If it does, hidden Level 2 items are automatically generated for each input table. When the insert application is synced to a mobile device, the mobile client automatically generates 2nd level inputs for each table. When the sync update is performed, the 2nd level data is gathered and passed into different rows in the SAP table.

---

**Note** In update applications all parameter names must match the column labels from the parent table. When syncing an application, UA looks at linked update applications and walks down their list of update parameters to match them with label names in the parent table. If it doesn't find a match for every update parameter, it ignores the update application and you do not get an Edit button.

---

#### ❖ **Linking applications**

Before linking applications, create the applications you require; for example, you may need to create insert, update, delete, and login applications first, and then create a master application that links to these applications.

- 1 In Application Builder, navigate to the Mobile tab.
- 2 Click Add, then find and select the application to which to link. The application displays on the tab.
- 3 Make these settings for the linked application:
  - **Display Name** – for the master application, the name that displays on the mobile device. Enter a name that is more meaningful or is abbreviated to fit the mobile device screen size. For example, change `so.company_add` to `Add Company`.
  - **Action Type** – select the type of
    - **Insert** – indicates an insert operation.
    - **Update** – indicates an update application.
    - **Delete** – indicates a delete application.
    - **Verification** – indicates a verification application.
    - **Login** – indicates a login application.
  - **Menu Level** – select the menu level for the application from the drop-down list.

- Sequence – use the up and down arrows to position the linked applications in a chained sequence.

---

**Note** To remove a linked application, click the X box to the left of the application name.

---

## Restrictions

UA supports only two levels for linked components. During a sync-update operation, each separate insert/update application that is linked is executed serially (level 1s first, then level 2s). The portal has no transaction coordinator, so each application execution happens independently in the auto-commit mode. If one application fails, the portal does not roll back changes that were made to other applications. The administrator must manually correct data in this case.

SAP applications that have level 2 implicit tables are executed all at once. Even though they are auto-commit, the 1st and 2nd level data are applied simultaneously, so there is no possibility for a partial update failure.

## MApp XML (spidered XML) and server code

When you link SAP applications for transaction support, the information from the linked new components is combined with the parameters defined for each of the linked new components to generate a new <insert> XML section that is generated and transferred to the clients (BlackBerry or PocketPc) for rendering.

For the BlackBerry J2ME client, the <insert> XML is added as a new type of child element to the Mapp spidered XML. The <insert> XML is generated and appended to the Mapp XML only if there is one or more new linked components.

Any mobile application created with Create XYZ RFC typically has scalar values and tables as inputs. The number of tables selected for input can be one or more. The changeable scalar values are marked as update parameters. The input table rows/columns are converted to parameters with the X.Y.0 format naming convention.

The naming convention of parameters is used to determine the <insert> XML content of level 1 and level 2 for a linked new SAP component. In the <insert> XML shown below, scalar values such as PO\_HEADER, Vendor, and PO\_Header.Doc\_DATE' are listed at level 1, because this information is critical and required before the creation of a sales order.

The PO\_ITEM\_SCHEDULES and PO\_ITEM are marked as table input items in the mobile application definition, so these are automatically converted into level 2 components. Each unique table (such as PO\_ITEM\_SCHEDULES and PO\_ITEM) in the input definition has its own level 2 in the <insert> XML.

```
<insert type="sap">
  <level id="1">
    <app name="PO" wid="211" rid="21">
      <Parameter type="TEXT"
display_name="PO_HEADER.VENDOR"
default_value="0000003910"/>
      <Parameter type="TEXT"
display_name="PO_HEADER.DOC_DATE"
default_value="20050223"/>
    </app>
  <level id="2">
    <app name="PO_ITEM_SCHEDULES">
      <Parameter type="TEXT"
display_name="PO_ITEM_SCHEDULES.QUANTITY"
default_value="5000.000"/>
      <Parameter type="TEXT"
display_name="PO_ITEM_SCHEDULES.PO_ITEM"
default_value="00010"/>
      <Parameter type="TEXT"
display_name="PO_ITEM_SCHEDULES.DELIV_DATE"
default_value="20050228"/>
    </app>
    <app name="PO_ITEM">
      <Parameter type="TEXT"
display_name="PO_ITEM.PO_NUMBER"
default_value="4100000001"/>
    </app>
  </level>
</insert>
```

```
                <Parameter type="TEXT"
display_name="PO_ITEM.PO_ITEM"
default_value="0000003910"/>
                <Parameter type="TEXT"
display_name="PO_ITEM.PUR_MAT"
default_value="3300-113"/>
                <Parameter type="TEXT"
display_name="PO_ITEM.MATERIAL"
default_value="3300-113"/>
                <Parameter type="TEXT"
display_name="PO_ITEM.NET_PRICE"
default_value="123.450"/>
        </app>
</level>
</level>
</insert>
```

---

**Note** For offline BlackBerry SAP applications, if you enter one or more rows of data for a table, but do not provide input values for every column of the table, default values for those columns are supplied from the sample application data. If you do not want these default values used, then set the default values to empty on the Parameter window in Mobile Web Studio.

---

## Creating update links (with conflict resolution)

When you create an application to display on a PDA or BlackBerry client, and you want to give users the ability to update the application's source content from that PDA, link the source application with an update application, which acts as a queue to store updates on a PDA until the PDA is re-synchronized with the local host computer on which Unwired Accelerator is installed.

First, you must create the update application, then you must create the primary application with a link to the update application (see the *Mobile Application Development Tutorial* for instructions on creating primary and update applications).

For conflict resolution to work correctly on your update, you must also create a verification application and link the update application to it. The verification application must return a grid data set using a select statement based on the values of one of CGI parameters used as the search parameter. It must also include a header row that defines the columns of the grid. The column names must match the update parameter in the update application.

---

**Note** If conflict resolution is not an issue for you, there is no need to create a verification application. All updates take effect regardless of whether or not the value to be updated is current.

---

## Update and conflict resolution

You can perform connected (online) or disconnected (offline) updates of records from PDA or BlackBerry client applications. The *Mobile Application Development Tutorial* includes instructions for performing an update.

Having multiple copies of the same data on multiple mobile devices on which multiple users can update and perform sync operations introduces the problem of one user's updates cancelling modifications made by other users. Unwired Accelerator provides concurrency control and conflict resolution to solve this problem.

### Updating records from PDA applications

When you make changes on record data from your PDA device while running applications in disconnected mode, you can check the status of your update operation. You can confirm whether the update operation succeeded, and if not, you can find out why it failed. The Mobile Portal Server maintains a log of all update requests it receives from the M-Business server during sync operations. Logs are created in the portaldatabase.

When you perform sync operations, update logs are sent to the PDA device. Click Log in the application pages to see a list of all sync log records. The List page of the logs shows only a subset of the fields from the log record. To see more details, click any column of the log record to display the detail page.

Use the Delete button on the detail page to delete log records. Delete log records periodically, otherwise, the list grows and increases your sync times. When you delete a log record, it is immediately removed from the device database and no longer appears in the log record list. However, the records are not deleted from the portal server until the next sync operation.

## Conflict resolution

Unwired Accelerator uses optimistic concurrency control. This allows users to make changes to records on their local devices, but at sync time, the mobile portal server rejects an update operation if the data submitted for modification has already been modified by another user in the enterprise. For example, if four users change the same record and perform the sync operation, only the first user's sync is executed, and the other three users' update requests are rejected.

During sync, the PDA devices return the original value (the value before the change) of the updated record columns, the new value for the columns, and all the search parameter values defined in the updated application, to the linked application of the mobile portal server. See "Creating update links (with conflict resolution)" on page 148.

## Validating data

You can access the data validation window by clicking on the Data Validation tab in either the Finish window when you save a new application, or the Properties Editor when you edit an application. The Data Validation tab allows you to verify regular expressions that define the format of user entered data in update applications that contain grid data.

For each data field, you can optionally enter a Detailed Instruction and Error Message. These fields are used by the client device when using a mobile device to enter or edit data in the application.

### ❖ Validating data

- 1 Log in to Mobile Web Studio and create a new application that uses grid data.
- 2 Click Save to save the new application.
- 3 In the Finish window, click Data Validation.
- 4 In the data validation window, enter:
  - Validation Expression – enter a regular expression defining the format of the user-entered data.
  - Test – click to test your expression. When you click Test, the Regular Expression Tester window appears showing the results of the test.
  - Common Expressions – this drop-down list contains common regular expressions you can select from.

- Detailed Instruction – optionally enter detailed instructions.
  - Error Message – optionally enter a custom error message.
- 5 If the results are valid after testing, click Save to close the Regular Expression Tester window. You are returned to the Data Validation tab, where the Validation Expression fields are populated with the tested regular expression.
  - 6 Click OK.
  - 7 Save the application.

## ContextID linkage across applications

When creating a set of applications that are linked through SSCA, you may need to maintain a “backend” session context. For example, you might have a number of applications that represent various screens on a Web application, which requires a user session to be maintained as the user moves from screen to screen. UA automatically maintains backend Web context for you through SSCA, provided you select “In Context” on the application properties. Then when you click a link in an application that uses SSCA, the application that is then loaded executes with the same context the previous one.

## Login application

When capturing applications, you can choose to attach a separate Login application to improve synchronization time to mobile devices. Create the Login application separately, and link it to the application.

Link two applications together through a “login” relationship. This is a new relationship in UA 8.0, in addition to the existing ones of “delete,” “insert,” “update,” and “verification.” If an application is related to another via a “login” relationship, the following applies:

- During Web Studio preview, when editing an application that has a linked “login” application, the login application is executed silently. As a result, any back-end Web server context is established prior to executing the main application.

- During off-line spidering (via the map-xml command), any linked “login” application is executed – but only once, to ensure that any required context is set. As a result, the login step is executed just once, rather than for each iteration through the spidering.
- During normal on-line playback, if there is a login application, the login application is executed at least once in the same context as the main application. Context is maintained through JSESSIONID session context objects in UA along with the ContextId parameter (see “ContextID linkage across applications” on page 151).

If the login application cannot be executed automatically (for example, if it requires input, then the main application does not execute either. In this situation, it is expected that the login application is executed as an application in its own right first, and then the main application is executed through application linkage (by passing context).

## Linking parameters

Unwired Accelerator enables you to set up chained relationships among field/parameter values. This makes it easier to create and update applications.

This is also useful when integrating Unwired Accelerator with Remedy, or another enterprise application or data source. For example, on the Remedy Administration GUI, you have many drop-down fields to choose from, and the values of some drop-down fields you select can affect choices in other fields. Linking parameters provides a way to imitate the Remedy Administration GUI, by linking a field to its data source. Additionally, on the mobile device, the drop-down options are available to the user.

Unwired Accelerator supports many data sources, including the following, for linked parameters:

- XML document
- Database table
- HTML document
- JSP output

However, these data inputs must be converted into a grid table in one specific format by utilizing any efficient transformation method (for example, using an XSLT template for an XML document) and querying the database table using proper SQL.

Unwired Accelerator provides support for the XML document data source, and a built-in XSLT template to do this transformation. A sample XSLT template is provided for XML document data source:

```
%SYBASE%\tomcat\webapps\onepage\portlets\samples\linkedparameters.xml
```

This example is generated using real data from Remedy application. Its schema is located at:

```
%SYBASE%\tomcat\webapps\onepage\schema\linkedparameters.xsd
```

Unwired Accelerator also provides a built-in XSLT template to generate grid table out of the XML document described above. To view the template, select Build | Templates | Approved in the left pane, and select Linked Parameters in the right pane. You can either click Edit to start the Template Editor, or click Preview to see the XML code.

See the *Mobile Application Development Tutorial* for detailed information and procedures for using linked parameters with Remedy mobile applications.

#### ❖ Linking parameters

- 1 In Configure Parameters tab, expose any parameters you need.
- 2 Under Display Type, select Linked from the drop-down list. The Add and Clear buttons appear.
- 3 Click Add, then find and select the application to which to link. This application is the data source. After you save and close the window, the linked application displays in the Component column.

---

**Note** You can select multiple parameters and select Add, to link the application to multiple parameters.

---

- 4 Repeat the procedure for any parameters you need.

---

**Note** To remove a linked parameter, click Clear.

---

- 5 Make any additional configuration parameter settings you require, and create or edit your application using regular procedures.
- 6 In the Configure Parameters window, click Save.

## Handling JSESSIONID

Many Web sites (especially those using J2EE application servers) use the JSESSIONID approach to managing sessions. This involves sending a cookie (JSESSIONID) during the login process, and the browser generating a URL request containing the string `';jsessionid=nnnnnnnn'` in return. The cookie value and the URL value must match.

Cookies are sent dynamically, so to prevent future playbacks from generating URL requests with the wrong `“;jsessionid=”` string embedded, when the playback engine sees that a URL has the string `“;jsessionid=”` as part of the URL, it automatically replaces the value of the JSESSIONID parameter with the value sent in the JSESSIONID cookie. This ensures that URLs sent in CCL playback are consistent with the actual JSESSIONID value.

### @OP cookie tags

In some cases, session values are included in URLs, but not through the `“;jsessionid=”` approach. A mechanism has been introduced that allows you to replace tagged sections of the CCL with the content of cookies in the current session cookie JAR. This works by allowing you to define runtime substitution of parameter values.

The @OP cookie tag allows you to specify runtime replacement of query parameters using the value of a specified cookie. You can do this by using this syntax:

```
@OPCOOKIE["<cookie name>"]
```

where `<cookie name>` is replaced by the name of the specified cookie. For example:

```
@OPCOOKIE["JSESSIONID"]
```

There is an alternative form, @OPCOOKIEENC, which URL encodes the cookie value when substituted. The syntax is otherwise the same.

## Adding JSP-related capabilities

Unwired Accelerator enables skilled users to enhance its capabilities quickly and easily without code-level changes using JSP. Unwired Accelerator sits within a JSP container (Tomcat, EAServer, or other suitable servlet container), so the framework for calling out to user-provided JSP is already in place. Unwired Accelerator provides hooks to allow users to insert a JSP component where needed.

Typically a JSP interacts with UA via request attributes, which are presented to the JSP when it is invoked; the JSP performs an operation, and returns results via the same or other well-defined attributes.

---

**Note** Because a JSP is invoked within the context of the servlet session, and in the same JVM as UA, there are issues about security, reliability, and so on. Sybase recommends these JSPs only be added by suitably authorized personnel, and only at development time. See the *Unwired Accelerator Administration Guide* for information about security.

---

Following is a summary of ways you can add JSP capabilities in Unwired Accelerator:

- Definition time
  - Web Services definition – this occurs when creating “advanced” web services only.
  - CGI Parameter definition – this occurs when creating “advanced” web services only.
- Execution time
  - CGI parameter pre-processing – when using server-side, click-across.
  - CCL execution
    - Data extraction/generation
      - Page Content processing – during page loads from remote web sites.
      - XSLT – when processing XML using XSLT. Currently only possible to set this up with ‘advanced’ web services unless you edit the CCL directly.
      - Web Services – during “advanced” web services only.
      - JSP - for JSP-elements.

- Feature extraction
  - Custom parsing – to perform custom feature extraction on HTML pages.
- SetupPage – allows developers to replace the standard input fields on a parameter-driven application with their own.
- Post-processing – provides developers with the ability to control any template processing. No current implementation.
- Mobile Device Template – this template is used to generate output targeted at a particular mobile device. These are JSPs that process (intermediate) output generated during application playback when we are routing the playback to a mobile device. When defining an application, you can associate different templates for different device types. If the template type is “JSP” then UA will generate content in a form that is compatible with mobile device templates.

In general, for template types of HTML/XSL, it is possible to get the template processor to invoke a JSP to generate the template content. For example, assume you have an XSL template; you have these three choices for defining the template body:

- Simply insert XSL content into the template body.
- Indicate a path to a JSP—such as */portlets/jsp/custom/some.jsp*—that generates XSL, into the template body
- Make sure the template body starts with `<%`, and then the template body is executed as a JSP that generates XSL. In this case the template body is the JSP content.

This pattern is also available for HTML templates. This means that you can get the template body output generated via a JSP, either by pointing at a JSP or by including the JSP content directly into the template body.

---

**Note** See Chapter 6, “Building Templates” for information about creating various types of templates.

---

## Generalizing parameters with @OP tags

Unwired Accelerator provides a way to generalize parameters through the use of the @OP special variable. An @OP tag indicates a parameter that should be altered at playback. You can indicate a default value for @OP tags on the CGI parameter screen.

Generally @OP tags are used to make a SQL query generic, but @OP tags can also be used in Web-based applications. Following are example uses for @OP parameters:

- @OP tags for connection pools – see “Creating SAP applications” on page 70.
- @OP tags in the SQL query of a database or in XML – see “@OP parameters” on page 92.
- @OP tags in grid rules – see “Using @OP tags with grid rules” on page 141.
- @OP cookies – see “@OP cookie tags” on page 154.
- @OP tags as part of an URL – see “Using @OP tags in URLs” on page 157.
- @OPCGIPARAM – a mechanism used to extract capture time variables in HTML code for use by down-stream CCL commands. An example would be generalizing `name:value` pairs. See “Using @OPCGIPARAM to define a default value” on page 158.

## Using @OP tags in URLs

You can use @OP tags as part of an URL to be altered at playback, and these are substituted just as they are for @OP tags in other situations, for example as part of SQL query strings. Common situations where the use of @OP tags might be required are when you need to parameterize parts of the URL path component – the section of the URL after the scheme/host/port, and before any query parameters. For example:

```
http://machine.sybase.com:4040/@OP["path"="whatever"]/index.html
```

It might also be necessary to use @OP tags to parameterize part of a query parameter. UA makes query parameters available for parameterization, but this only allows users to enter all the parameter values. If however, it is necessary to only replace a portion of a query parameter, then you need to use @OP tags. For example, assume you have a URL like:

```
http://machine.sybase.com:4040/path/index.html?complicatedparam=param1%3Dvalue1%26param2%3Dvalue2
```

In this case, you want to make the individual components of `complicatedparam` individually parameterizable. So you might change this to:

```
http://machine.sybase.com:4040/path/index.html?complicatedparam=param1%3D@OP["param1"="value1"]%26param2%3D@OP["param2"="value2"]
```

Then the user is presented with two parameters—`param1` and `param2`—which are substituted in the URL. Relying on “whole” parameter substitution would mean that the user would have had to enter the combined parameter.

---

**Note** The @OP tag must contain a suitable default value, as this is used to determine the initial navigation point.

---

## Using @OPCGIPARAM to define a default value

You can use @OPCGIPARAM on the CGI parameter page to define a parameter’s (default) value, using syntax similar to:

```
@OPCGIPARAM[sEntry]
```

Note there are no quotes around the text inside the []. This value is used during the execution of the application (or portlet), when the @OPCGIPARAM tag is replaced by a runtime value derived by looking up the value (in this example, “sEntry”) which has been determined earlier on in the application execution.

This is necessary sometimes during Web element execution, when we need to generate a HTTP post or get, that includes CGI parameters that vary from execution to execution. For example, a URL may have a CGI parameter that indicates some session key or cache lookup key. During original application capture, UA records the URLs with the values of the CGI parameters obtained at that time – but subsequent executions need values appropriate for that execution.

@OPCGIPARAM[] allows you to specify that the CGI parameter for the URL be modified appropriately at execution time. Note that the @OPCGIPARAM tag specifies a “key” or “tag” that indicates to UA that we lookup the OPCGIPARAM with this name.

UA determines the value for the @OPCGIPARAM tag in two ways: automatically, or by some explicit coding. During execution of a Web application, UA notes all CGI parameters that appear in URLs that are read from the Web server. These URLs could be the result of explicit requests encoded in the CCL used to describe this Web application, or the result of frame fetches or redirects that occur during loading. Quite often a URL is requested by a HTTP get, and a redirect to a session-sensitive URL occurs, with the session key encoded in the redirect URL. UA automatically notes and records these URLs and the CGI parameters.

For example, if a redirect occurs to: `http://www.company.com/somepath?cmd=somecmd&session=24155217677FFA65`, UA automatically records the OPCGIPARAMs “cmd” and “session”. Then if the next URL request in the CCL has session as a CGI parameter, using the OPCGIPARAM syntax, you can automatically use the value that occurred during the redirect.

It is also possible to determine OPCGIPARAMs programatically, via a “page content processor” JSP. This is a JSP that can be specified to analyze intermediate page content during the web execution. You specify such a JSP when determining a suitable “CCL fixup” step on the Select page immediately after capturing the target Web page. This JSP can return a set of OPCGIPARAM key/value pairs, which can then be accessed using the OPCGIPARAM syntax. This is used in Business Objects Capture strategy, described in “Business Objects capture strategy” on page 168.

## Custom capture options

Unwired Accelerator provides several custom capture options and strategies for the following:

- Custom – Web capture option that allows developers to create their own “feature extractor” JSP that processes HTML content. The JSP is used at capture time and run time.
- Crystal – Custom Web capture option that has been adapted for the unique requirements of Crystal Reports. It is intended only to support Crystal Reports generated by the Business Objects XI server, Release 1. For additional Business Object capture strategies, see “Business Objects capture strategy” on page 168.

- Custom XSL – Web capture option that allows developers to create an XSL file that can be used to transform HTML from the target Web site that has been converted by jTidy to XML.

The custom options provide alternative means to get around problems or issues that would otherwise prevent standard Web capture in Unwired Accelerator. The enhancements manipulate capture and playback mechanisms to overcome the problems.

Unlike other Unwired Accelerator features, custom capture options require you know basic Java, can create JSP and XSL/XSLT files, and understand how HTML pages work. Additional time is also needed to analyze the Web source, in order for you to create the JSP and XSL files to extract the features you want to capture.

## Custom Web capture option

The Custom Web capture option allows you to create your own “feature extractor” JSP that processes HTML content. The JSP returns in a list of possible formats during capture, and returns the specified HTML at run time (or playback).

For the Custom approach, the JSP file to be specified would be responsible for returning either a list of possible captures or a specified capture. The developer is responsible for writing the Java code that parses the HTML – although they could extend the HTML parsers that are used by UA for the standard feature extraction strategies.

## Creating the JSP file

Before you can create the JSP, get familiar with the Web page you plan to capture. You need to determine:

- What is the page content?
- How is the page rendered?
- How do you extract the feature you plan to capture?

Then, write the JSP, keeping in mind the following guidelines:

- The JSP is invoked either in Capture Mode or Run Time Mode. This is indicated by the request attribute “capture” which is a Boolean (“true” indicates “Capture Mode,” and “false” indicates “Run Time Mode”).

- When executing in Capture Mode, the JSP is responsible for returning a list of possible captures. This list is returned as a Java Vector object in the request attribute “allkeys.”
- Each element of the Vector is a capture “tag” or “name.” The JSP can use any naming convention it chooses, but if the tag or name starts with “grid,” it indicates to UA that the captured element is a grid, and will be subject to grid rules processing.
- At run time, UA presents the tag or name of the capture that is required in the request attribute “key” to the JSP. The JSP is then responsible for returning the HTML snippet for this requested tag. This is just returned as JSP output (see the example). In order to extract the correct HTML snippet, the JSP needs access to the HTML of the complete target page, and this is available through the request attribute “html”. Clearly the JSP can process this HTML in any fashion it chooses, and construct output HTML as it thinks fit. But remember, if the tag starts with the string “grid,” it must be an HTML table amenable to grid rules processing.
- Useful JSP request attributes:
  - capture – whether capture or playback
  - html – HTML content
  - url – URL for HTML
  - params – any parameters supplied to the JSP
  - key – key to match (run time only)
- Attributes JSP returns:
  - allkeys – list of all keys (capture time only)
  - HTML content in runtime
- Other JSP attributes the JSP can make use of:
  - conextID – ConextID value used to link applications together
  - sessionID – current Session ID
  - resourceID – the RID associated with playback
  - bindings – parameters used to play back the application

Following is a simple, JSP sample:

```
1: <%  
2: Boolean capture = (Boolean) request.getAttribute("capture");  
3: String html = (String) request.getAttribute("html");
```

```
4: Hashtable params = (Hashtable) request.getAttribute("params");
5:
6: if (capture.booleanValue())
7: {
8:     // capture time ...
9:     Vector allkeys = new Vector();
10:    // return 3 keys ...
11:    allkeys.addElement("grid:1");        // a grid key
12:    allkeys.addElement("grid:2");        // a grid key
13:    allkeys.addElement("nongrid:3");     // a non-grid key
14:    request.setAttribute("allkeys", allkeys);
15:    return;
16: }
17: else
18: {
19:     // run time ...
20:     String key = (String)request.getAttribute("key");
21:     String extractHtml = null;
22:     if (key.equals("grid:1"))
23:     {
24:         extractHtml = "<table><tr><th>Col 1</th><th>Col
25: 2</th></tr><tr><td>Col 1 val</td><td>Col 2 val</td></tr></table>";
26:     }
27:     else if (key.equals("grid:2"))
28:     {
29:         extractHtml = "<table><tr><th>Col 1</th><th>Col 2</th><th>Col
30: 3</th></tr><tr><td>Col 1 val</td><td>Col 2 val</td><td>Col 3
31: val</td></tr></table>";
32:     }
33:     else if (key.equals("nongrid:3"))
34:     {
35:         extractHtml = "<p>Some content ....</p>";
36:     }
37:     else
38:     {
39:         // shouldn't happen ...
40:         extractHtml = "<table><tr><th>Error</th></tr><tr><td>Bad key:
41: "+key+"</td></tr></table>";
42:     }
43: }
44: %>
```

## Creating the Custom Web application

Before you start, you should have analyzed the Web page, and created a JSP file to handle the capture and playback challenges you are overcoming. Typically the JSP files are placed in *SYBASE\tomcat\webapps\onepage\portlets\jsp\custom*.

### ❖ Using Custom Web capture

- 1 In Mobile Web Studio, open the Application Builder, and click Add. The New Web Element window displays.
- 2 In Address, enter the Web address, such as `http://localhost:4040/index.jsp`.
- 3 Select Custom from the drop-down list. The “JSP file” field displays.
- 4 In “JSP file”, enter the JSP file name.
- 5 In the Select window, choose the grid you want, typically the first one, and click Next.
- 6 On the Define and Filter windows, make changes, and click Next.
- 7 On the Parameters window, you will see the parameters that were used to reach the report you selected. You can click the select box next to them, and customize as needed.  
Click Next.
- 8 On Window Preview, enter a name for the element. Click Finish to save the element.
- 9 In Application Builder, follow the normal Save procedure to name your application, specify List/Detail fields, and mark the application for offline use, and so on.

## Crystal Reports Web capture option

The Crystal Reports Web capture option is a Web capture option that has been adapted for the unique requirements of Crystal Reports. It is intended to support Crystal Reports generated by the Business Objects XI server, release 1. (For additional capture strategies, see “Business Objects capture strategy” on page 168).

The crystal capture strategy is one instance of custom capture. It invokes *crystal-main.jsp*. This in turn includes *crystal-standard.jsp*, which performs the actual extraction. The two jsps exist to allow further customization of Crystal capture as explained below.

In some cases, the default Crystal capture fails because the HTML contains extra data. In this case, it is necessary to extract only the relevant parts of the entire HTML. This can be done by using Custom strategy and using a custom JSP that performs the extraction. This custom JSP needs to define a preprocess method that is passed to the entire HTML, and returns the extract. See *crystal-omit-tree-panel.jsp* in *SYBASE\tomcat\webapps\onepage\portlets\jsp\custom* for an example of this approach.

Unwired Accelerator provides the sample JSP for Crystal Reports, *crystal-standard.jsp*, which includes *crystal-main.jsp*, but allows users to modify it to do some additional pre-processing of the Crystal HTML before doing the main analysis. Both *crystal-main.jsp* and *crystal-standard.jsp* are located in *SYBASE\tomcat\webapps\onepage\portlets\jsp\custom*.

#### ❖ Using Crystal Web capture

- 1 In Mobile Web Studio, open the Application Builder, and click Add. The New Web Element window displays.
- 2 In Address, enter the Web address, such as `http://localhost:4040/index.jsp`.
- 3 Select Crystal from the drop-down list. The “JSP file” field displays.
- 4 In “JSP file”, enter the JSP file name, such as `crystal-standard.jsp`.
- 5 In the Select window, choose the grid, and click Next.
- 6 On the Define and Filter windows, make changes, and click Next.
- 7 On the Parameters window, you will see the parameters that were used to reach the report you selected. You can click the select box next to them, and customize as needed.  
  
Click Next.
- 8 On Window Preview, enter a name for the element. Click Finish to save the element.
- 9 In Application Builder, follow the normal Save procedure to name your application, specify List/Detail fields, and mark the application for offline use, and so on.

## Custom XSL Web capture option

The Custom XSL Web capture option allows developers to create an XSL file that can be used to transform HTML from the target Web site to XML using jTidy.

When selecting Custom XSL as a strategy, a View button appears that allows users to view the “tidied” HTML that will be presented to the XSL. This HTML (which is actually XML, after jTidy has processed it) is then shown in a pop-up window, using IE's standard XML browser. Using this window it is possible to expand/collapse XML nodes and traverse the complete XML (HTML) tree returned. This will make creating the XSL required to map this to an appropriate output easier.

Plenty of resources exist on the Web for writing XSL. Keep in mind the following guidelines:

- The XSL can be used to create either XML or HTML. If it creates XML, then the XML must conform to the basic “Onepage XML” format:
  - Sequential `<Record>` tags, each containing a sequence of `<Field>` tags.
  - Each `<Record>` tag must contain the same number of `<Field>` tags.
  - The `<Record>` tags must be enclosed in a single tag `<data>`.
  - If the output is in this form, then the output must contain an XML comment `<!-- onepage_xml=true -->`. This is obtained in the XSL document by using `<xsl:comment> onepage_xml=true </xsl:comment>`
- If the output XML or HTML is to be processed by grid rules, then the output must contain the XML comment `<!-- enable_grid_rules=true -->` (and this can be obtained by including `<xsl:comment> enable_grid_rules=true </xsl:comment>` in the XSL).
- Unwired Accelerator automatically transforms HTML with jTidy, then applies the XSL.
- Unlike Custom capture, you can only define a single capture possibility.
- If a JSP is used to generate the XSL, the following parameters are available to the JSP (as request attributes):
  - `url`
  - `html`

- params – key value pairs extracted from JSP
- sessionID – current session ID
- contextID
- resourceID
- bindings

The following Custom XSL sample returns fixed HTML content regardless of the input HTML.

```
1:<?xml version="1.0"?>
2:<!--
3:   This XSL shows an example that outputs fixed HTML content regardless of
4:   input HTML
5:   It also tells the CCL to enable grid rules
6: -->
7:<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
8:  <xsl:template match="/">
9:    <xsl:comment> enable_grid_rules=true </xsl:comment>
10:    <table class="ONEPAGE_FEGRID_CAPTURE_TABLE">
11:      <tr>
12:        <td>col1</td><td>Col2</td>
13:      </tr>
14:      <tr>
15:        <td>A</td><td>B</td>
16:      </tr>
17:    </table>
18:  </xsl:template>19:</xsl:stylesheet>
```

The following Custom XSL sample returns a grid layout.

```
1:<?xml version="1.0"?>
2:<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
3:  <xsl:template match="/">
4:    <data>
5:      <xsl:comment> enable_grid_rules=true </xsl:comment>
6:      <xsl:comment> onepage_xml=true </xsl:comment>
7:      <Record>
8:        <Field>F1</Field>
9:        <Field>F2</Field>
10:       <Field>F3</Field>
11:       <Field>F4</Field>
12:      </Record>
13:    </Record>
```

```
14:      <Field>R1</Field>
15:      <Field>R2</Field>
16:      <Field>R3</Field>
17:      <Field>R4</Field>
18:    </Record>
19:  </data>
20: </xsl:template>21:</xsl:stylesheet>
```

In both examples, notice the use of `<xsl:comment>` tags to indicate to UA how the output should be treated:

```
<xsl:comment> enable_grid_rules=true </xsl:comment>
<xsl:comment> onepage_xml=true </xsl:comment>
```

#### ❖ **Using Custom XSL Web capture**

- 1 In Mobile Web Studio, open the Application Builder, and click Add. The New Web Element window displays.
- 2 In Address, enter the Web address, such as `http://localhost:4040/index.jsp`.
- 3 Select Custom XSL from the drop-down list. The “XSL file” field displays.
- 4 In “XSL file”, enter the XSL file name.
- 5 In the Select window, choose the grid, and click Next.
- 6 On the Define and Filter windows, make changes, and click Next.
- 7 On the Parameters window, you will see the parameters that were used to reach the report you selected. You can click the select box next to them, and customize as needed.  
Click Next.
- 8 On Window Preview, enter a name for the element. Click Finish to save the element.
- 9 In Application Builder, follow the normal Save procedure to name your application, specify List/Detail fields, and mark the application for offline use, and so on.

## **Business Objects capture strategy**

This section describes capture strategies for capturing Business Objects XI Web content. The capture strategy uses JSP to re-write CCL output, thereby handling some known playback (runtime) problems.

Unwired Accelerator provides the following JSP templates for re-writing CCL

- Use Business Objects 6.5 Template
- Use Business Objects 6.5 Template (for use with separate login app)
- Add BO 6.5 Run-time Page Content Processor only
- Use Business Objects XI Template
- Use Business Objects XI Template (for use with separate login app)
- Add BO XI Run-time Page Content Processor only

To apply the JSP to fix up the CCL, choose the appropriate JSP on the Select page of the Web capture wizard. See “Capturing Business Objects” on page 168 for the general procedure of applying the template.

To learn more about the JSP used, see “Understanding Business Object JSP templates” on page 170 for information about each of the Business Objects templates, and see “Understanding Page Content Processor JSP” on page 173 for information about page content processor JSP.

## **Capturing Business Objects**

This section describes the basic procedure for capturing Business objects using Web capture, and applying one of the following JSP templates to re-write CCL during Web capture:

- Use Business Objects 6.5 Template
- Use Business Objects 6.5 Template (for use with separate login app)
- Add BO 6.5 Run-time Page Content Processor only
- Use Business Objects XI Template
- Use Business Objects XI Template (for use with separate login app)
- Add BO XI Run-time Page Content Processor only

❖ **Capturing Business Object XI content**

- 1 In Mobile Web Studio, open the Application Builder, and click Add. The New Web Element window displays.
- 2 In Address, enter the BusinessObjects InfoView URL, check “Advanced Navigation,” and click Go.
- 3 When the BusinessObjects login screen displays, enter your Business Objects XI user name and password, and click Log On.
- 4 Navigate the InfoView panels to get the report you plan to mobilize. You can drill down or set any filters needed; the filters will be included as parameter options.

See the *Mobile Application Development Tutorial* for the Business Objects tutorial that shows these details.

- 5 After you select the area to mobilize and the grid selection window displays, click the “Re-write CCL using a pre-defined template” box. A drop-down list and Apply button appears.
- 6 Select one of the objects from the drop-down list, depending on the BusinessObjects version you are using:
  - Use Business Objects 6.5 Template
  - Use Business Objects 6.5 Template (for use with separate login app)
  - Use Business Objects XI Template
  - Add BO 6.5 Run-time Page Content Processor only

Click Apply. The feature selection window re-displays. This allows you to verify that the CCL re-write was successful, and to choose the feature that best suits your requirements. The CCL re-write modifies the playback instructions, optimizing them according to the template chosen. The new CCL instructions, as updated by the selected template, is used for all future playbacks of this application. Each template modifies the CCL in the best way for the use for which it was designed, so it is important that the correct template is selected.

- 7 Select the grid you want, typically the first one. Click Next.
- 8 On the Define and Filter windows, make changes, and click Next.

- 9 On the Parameters window, you will see the set of Web pages and their associated parameters that were used to reach the specified report you selected. You can identify any parameters that reflect the filters or drill-downs you performed in the initial screen of the wizard. You can click the select box next to them, and customize as needed.

---

**Note** There may be multiple fields in the list that relate to the same filter. You must repeat this step for all those fields, and ensure the same Display Name is used for all of them.

---

Some of the parameters may have their default values changed to @OPCGIPARAM[...] values. This is done automatically during the CCL re-write, so that the playback CCL works in the correct way with the Business Objects Server. The correct @OPCGIPARAM substitution varies from template to template, and so it is important to use the correct re-write template.

Click Next.

- 10 On Window Preview, enter a name for the element. Click Finish to save the element.
- 11 In Application Builder, follow the normal Save procedure to name your application, specify List/Detail fields, and mark the application for offline use, and so on.

## Understanding Business Object JSP templates

This section provides information about each of the Business Object JSP templates available with Unwired Accelerator. Each template has a specific application, such as Business Objects XI or 6.5, but you can use them as a model for developing custom templates.

- Use Business Objects 6.5 Template
- Use Business Objects 6.5 Template (for use with separate login app)
- Add BO 6.5 Run-time Page Content Processor only
- Use Business Objects XI Template
- Use Business Objects XI Template (for use with separate login app)

- Add BO XI Run-time Page Content Processor only

---

**Note** The “Run-time only” templates are designed to re-write the CCL to add in the OPCGIPARAMs mentioned above. See “Understanding Page Content Processor JSP” on page 173 for information useful for ensuring OPCGIPARAM substitution works properly.

---

### Business Objects 6.5 template

This JSP processes CCL produced during Business Object capture and alters it to play back using HTTPClient rather than ActiveX. The CCL to be processed is presented on the request object via the “ccl” attribute, and is returned in the same attribute. The fixup works only on CCL generated for Business Objects 6.5. See the default template in:

```
SYBASE\tomcat\webapps\onepage\portlets\jsp\custom\fixupBO65CCL.jsp
```

### Business Objects 6.5 template (with separate login app)

This JSP processes CCL produced during Business Object capture and alters it to play back using HTTPClient rather than ActiveX. It also removes any login HTTP requests. The CCL to be processed is presented on the request object via the “ccl” attribute, and is returned in the same attribute. The fixup works only on CCL generated for Business Objects 6.5. See the default template in:

```
SYBASE\tomcat\webapps\onepage\portlets\jsp\custom\fixupBO65CCL-  
removelogin.jsp
```

### BO 6.5 Run-time Page Content Processor only

This JSP processes page content and returns any variables (OPCGIPARAMs) for future processing during CCL playback. It is designed to process Business Objects 6.5 HTTP requests, and determine the correct sEntry values.

The JSP takes a vector of page contents and processes each one in turn. When finished it places on the request object an attribute named “variables” that contains any OPCGIPARAM vars, which have been set as a result of the processing.

The request attribute “url” contains the URL that was requested that triggered the page loads. The request attribute “content” is a vector of content objects, for each page that was loaded as a result of the target being requested. This processing will only work for BO 6.5. See the default template in:

*SYBASE\tomcat\webapps\onepage\portlets\jsp\custom\handleBO65PageContent.jsp*

---

**Note** See “Using @OPCGIPARAM to define a default value” on page 158 for general information about @OPCGIPARAM, and see “Understanding Page Content Processor JSP” on page 173 for more detailed information, to help ensure OPCGIPARAM substitution works properly.

---

## Business Objects XI template

This JSP will process CCL produced during Business Object capture and alter it to play back using HTTPClient rather than ActiveX. The CCL to be processed is presented on the request object via the “ccl” attribute, and is returned in the same attribute. The fixup will only work on CCL generated for Business Objects XI. version 1 (version 2 is not supported). See the default template in:

*SYBASE\tomcat\webapps\onepage\portlets\jsp\custom\fixupBOXICCL.jsp*

## Business Objects XI template (for use with separate login app)

This JSP will process CCL produced during Business Object capture and alter it to play back using HTTPClient rather than ActiveX. It also removes any login HTTP requests. The CCL to be processed is presented on the request object via the “ccl” attribute, and is returned in the same attribute. The fixup will only work on CCL generated for Business Objects XI. version 1 (version 2 is not supported). See the default template in:

*SYBASE\tomcat\webapps\onepage\portlets\jsp\custom\fixupBOXICCL-removelogin.jsp*

## Business Objects XI run-time page content processor only

This JSP processes page content and returns any variables (OPCGIPARAMs) for future processing during CCL playback. It is designed to process Business Objects XI HTTP requests, and determine the correct sEntry/logonToken values.

The JSP takes a vector of page contents and processes each one in turn. When finished it places on the request object an attribute named “variables” that contains any OPCGIPARAM vars, which have been set as a result of the processing.

The request attribute “url” contains the URL that was requested that triggered the page loads. The request attribute “content” is a vector of content objects, for each page that was loaded as a result of the target being requested. This processing works only for Business Objects XI, version 1 (version 2 is not supported). See the default template in:

```
SYBASE\Tomcat\webapps\onepage\portlets\jsp\custom\handleBOXIPageContent.jsp
```

---

**Note** See “Using @OPCGIPARAM to define a default value” on page 158 for general information about @OPCGIPARAM, and see “Understanding Page Content Processor JSP” on page 173 for more detailed information, to help ensure OPCGIPARAM substitution works properly.

---

## Understanding Page Content Processor JSP

The example JSP that follows processes Business Object XI pages. The JSP processes page content and returns any variables (OPCGIPARAMs) for future processing during CCL playback. It is designed to process Business Objects XI HTTP requests and determine the correct sEntry/logonToken values.

It takes a vector of page contents and processes each one in turn. When finished it places on the request object an attribute named 'variables' that contains any OPCGIPARAM variables that have been set as a result of the processing.

The request attribute “url” contains the URL that was requested that triggered the page loads. The request attribute “content” is a vector of content objects, for each page that was loaded as a result of the target being requested.

---

**Note** See the notes following the code for more information.

---

```
<%--
```

```
This JSP processes page content and returns any variables (OPCGIPARAMs) for future processing during CCL playback. It is designed to process BO XI HTTP requests and determine the correct sEntry/logonToken values.
```

...

This processing will only work for BO XI.

```
--%>
<%@ page import="java.util.*" %>
<%@ page import="java.io.*" %>
%
boolean debug = false;

String targeturl = (String)request.getAttribute("url");
Vector content = (Vector)request.getAttribute("content");

if (debug) System.out.println("=====\n\ntarget url: " +
targeturl);

// we don't process any URL that isn't directed at the BO XI server
if (targeturl.indexOf("/businessobjects/enterprise11/") == -1)
    return;

Hashtable vars = new Hashtable();

for (int i = 0; i < content.size(); i++)
{
    // Each content item is a Properties object
    // The 'url' property contains the URL for the content
    // The 'html' property contains the HTML page content
    Properties pageInfo = (Properties)content.elementAt(i);

    String urlStr = pageInfo.getProperty("url");
    String html = pageInfo.getProperty("html");

    if (debug) System.out.println("JSP: url: " + urlStr);
    if (urlStr.indexOf("/InfoView/listing/reportList.do") != -1)
    {

        int idx = html.indexOf("return finalUrl + \"logonToken=");
        if (idx == -1)
        {
            if (debug) System.out.println("No logonToken ...");
            continue;
        }

        int endIdx = html.indexOf("\"", idx+30);
        String logonToken = html.substring(idx+30, endIdx);
```

```

        if (debug) System.out.println("JSP: logonToken: " + logonToken);

        Hashtable urlVars = new Hashtable();
        urlVars.put("logonToken", logonToken);
        vars.put(urlStr, urlVars);
    }

else if (urlStr.indexOf("/viewers/cdz_adv/viewCDZDocument.jsp") != -1)
{
    int idx = html.indexOf("writeBody(\"report.jsp?sEntry=");
    if (idx == -1)
    {
        if (debug) System.out.println("No strEntry ...");
        continue;
    }

    int endIdx = html.indexOf("&", idx+29);
    String strEntry = html.substring(idx+29, endIdx);

    if (debug) System.out.println("JSP: sEntry: " + strEntry);

    Hashtable urlVars = new Hashtable();
    urlVars.put("sEntry", strEntry);
    vars.put(urlStr, urlVars);
}
}

request.setAttribute("variables", vars);
%>

```

The main things to note are the input request attributes “url” and “content,” and the output request attribute “variables.”

- **Input request attributes**

“url” is a String used to tell the JSP what the URL requested during this step was.

“content” is a Vector of objects, describing all the HTML pages that were returned as a result of requesting this URL (there may be multiple pages). Each object in the Vector is a Java Properties object, containing a “url” and “html” property. The “url” property contains the actual URL of the content; “html” contains the HTML as a Java String. The JSP is responsible for parsing the HTML as it thinks fit, and extracting any values that will be useful as OPCGIPARAM key pairs.

- Output request attribute

On exit, the JSP should set the attribute “variables.” This is a Java Hashtable, that is keyed on urls – each URL would key another Hashtable, which contains the name/value pairs that have been determined for that URL.

When UA saves the name/value pairs, it does so under two keys: the simple name, and a key constructed from the URL the name was seen under and the name, as `<url>:<name>`. For example;

```
http://www.company.com/path:sEntry
```

In this example, the URL is `http://www.company.com/path`, and the OPCGIPARAM key is `sEntry`. This allows developers to decide whether OPCGIPARAMs are sensitive to URLs or generic across all URLs.

---

**Note** Note that the URL saved is always saved without any query parameters.

---

The Page Content Processor JSP is executed during web element runtime, such that after each HTTP get or post is recorded in the CCL, and submitted by UA, we call the JSP and present it with the returned HTML page content (as described above).

## Remedy capture strategy

This section describes strategies for creating Remedy applications. The strategy uses advanced web services, and WSDL files to access Remedy Java methods and the Remedy data source. You can create a master Remedy application, and link to insert, update, verification, and login applications, and you can create a web service application to access the Remedy data source.

You can create a JSP template to handle known capture problems, especially for older versions of Remedy. See “Adding JSP-related capabilities” on page 155 and Chapter 6, “Building Templates” for information.

The following section shows the basic procedure for using the Mobile Web Studio interface to create a Remedy application. See the *Mobile Application Development Tutorial* for detailed information.

**❖ Creating Remedy applications using Web services and WSDL files**

- 1 From Mobile Web Studio, select Build | Applications, and then click the New button.
- 2 On Application Builder, click the down arrow to the right of the Add button, and select Web Service Element.
- 3 On the Web Service Element Definition window, identify the WSDL file to use:
  - Service Type – select WSDL.
  - UDDI Key/UDDI Registry – do not select this option.
  - WSDL URL – enter the WSDL file location; for example:

```
http://hostname/arsys/WSDL/public/  
hostname/HelpDesk_Query_Service
```

---

**Note** Replace the *hostname* string with the host name of the mid-tier server specific to your own installation.

---

- Use Standardization XSLT – do not select this option when creating Remedy applications using a Web service.
  - Get Method – click Get Method. The system retrieves the WSDL XML and populates the Method drop-down list. The list of defined services is extracted from the WSDL file you specified in WSDL URL.
- 4 If you cannot access the WSDL URL you specified, WSDL files may be kept in a secure location and require a user name and password for access. If necessary, see the Remedy system administrator for access information. You can use the WSDL Authentication option to gain access.
    - Authentication – select the Authentication option. The User Name and Password fields appear.
    - User Name and Password – enter the user name and password provided by your Remedy system administrator.
    - Click Get Method. The system should retrieve the WSDL XML and populate the Method drop-down list.
  - 5 On the Web Service Element Definition window, identify the method to use:
    - Method – select a method from the drop-down list.

- Click Preview to see a list of the input parameters of the selected Web service method in the Preview pane
- 6 In the Web Service Element Definition window, make sure Enable Grid Rules is selected. This option is required to display the element output in grid format.

---

**Note** If you want the Web service application to be viewable on a mobile device, you must check the Enable Grid Rules box.

---

- 7 Optionally on the Web Service Element Definition window, establish the login method for accessing the Remedy datasource:
- Define Login Method – select this option if you selected a Web service method that requires a separate invocation to a login method to get valid session information for future transactions.
- When you select this option, three new fields appear:
- Login Method
  - Login Params
  - Login Profiles
- Login Method – if you selected a method where the Web Service requires session information in the SOAP header, select the login method from the drop-down list; for example, RemedyDISSoap:Login.
  - Login Profiles – select a login profile from the drop-down list. The Login Params option disappears.
  - Login Params – alternatively, enter the login parameters manually in the Login Params text area. You must specify the datasource (such as *labxp*), the user name (*sa*), and password for your environment.
- 8 Optionally on the Web Service Element Definition window, establish BASIC Authentication.
- Enable BASIC Authentication – select the Enable BASIC Authentication option. The User Name and Password fields appear.
  - User Name and Password – enter the user name and password. The user name and password are used internally by the Web services element for basic authentication of HTTP requests of SOAP calls.
- 9 Optionally on the Web Service Element Definition window, specify a custom JSP file:

- Advanced – select the Advanced option. The JSP Filename field displays to the right of the Method or Login Method fields.
  - JSP Filename – enter the name of a JSP file you previously created.
- 10 On the Web Service Element Definition window, search for or define an XSLT template. An XSLT template is used to transform XML format to the grid format required to build mobile applications.
- Content XSLT – search for or create a content XSLT template.
  - UI XSLT – search for a user interface XSLT template.

---

**Note** To create the Web service element and to see the Web service application on a mobile device, you must define Content XSLT. The Content XSLT template transforms Remedy business object XML to the XML format required by Unwired Accelerator to create the grid format needed for mobile applications. See Chapter 6, “Building Templates” for information about XSLT templates.

---

- 11 Click Create, located to the right of Content XSLT. The Web Services Output XSLT Template Customization window displays.

---

**Note** In UA Web services, UA uses cached information for the templates. If you make changes to the cached Content XSLT template, the changes do not take effect until the next Tomcat server restart.

---

- 12 On the Web Services Output XSLT Template Customization window, modify the Content XSLT for the business object:
- Name – enter a name for the template; for example, `HelpDeskList`.
  - Roles – select the roles for the runtime version of the mobile application. Use the buttons to move roles in and out of the Available and Selected columns. These are Unwired Accelerator roles and have no bearing on Remedy object security.
  - Select Output Parameters for Display – select the output parameters to include in the Web service element. Use the buttons to move output parameters in and out of the Available and Selected columns. Use the horizontal scroll bar to view the entire output parameter name.

- Update Output Display Name – for Output Parameters for Display, highlight an object in the Selected list, such as `FullName`. The Update Output Display Name field is populated with this value. You can change the value to something more meaningful or readable for the mobile application, such as `Full Name`.
- 13 If you do not have multiple records within the selected object, skip this step.
- If you have multiple records within the object (for example, multiple card codes, employees, service codes, and so forth), select the node level of the repeating record.
- Remedy business objects often contain multiple records at various nodes levels. Unwired Accelerator supports one node level per element, but enables you to specify the node level to use for a specific record. You select the node level to use from the Web Services Output XSLT Template Customization window. If you select more than one node level, the Web Service element will not work.
- Select Node Level That Contains One or More Child Records – click Select to the right of this field. The Web Services Output Node Level window displays.
  - Highlight a node level, such as `getListValues`, and click OK. The window closes and the value is populated in the Select Node Level That Contains One or More Child Record field.
- 14 Click Save to save the template, and OK to confirm then return to the Web Service Element Definition window.
- 15 On the Web Service Element Definition window, click Input, and type input for Remedy objects and parameters as needed:
- For objects, enter the object identifier; for example, for `_HEADER.AuthenticationInfo.userName`, you could enter `Demo`.

- For parameters, enter a default value; for example, for QueryListBy\_FullName\_Full\_Name, you could enter a.

---

**Note** If you want the Web service application output to be viewable on a mobile device, you must provide input parameters. In most cases, you can provide them on the Preview window to see the results more quickly, or you can provide them on the Configure Parameters window later in the process.

However, if you are creating a Web service element using the AddObject, UpdateObject, or RemoveObject methods, Sybase recommends you not enter input parameters in the Preview window. Otherwise a new record is created for every window in the mobile application wizard (Define, Filter, and so forth) since no invocation can be done, and no action can be performed to the database. Consequently, you end up with many user records, one for each window, rather than the one record you want. Instead, enter input parameters in the Configuration Parameters window for these methods.

---

- 16 Click Next. The Define window displays. The business object should display in a grid format, indicating the XML transformation from Remedy format to Unwired Accelerator format occurred successfully.
- 17 Use the Define, and Filter windows to define the mobile application in the same way you would define any other mobile application.
- 18 In Configure Parameters, under Variables, select the boxes for the parameters you need, and configure the CGI Parameters.

---

**Note** If you need to modify these parameters, you can do so after you save the application. Access Application Builder, select Params, and modify the parameters.

---

Click Next.

- 19 On the Preview window, enter an element name, and click Finish.
- 20 On the Application Builder window, save the mobile application and click Finish. Optionally, you can click Preview to preview the application, then approve the application.

❖ **Finalizing the Remedy application**

Once you have saved the mobile application, you can set up CGI parameters, including linked parameters, which enables you to set up chained relationships among field/parameter values. This makes it easier to create and update applications.

For example, on the Remedy Administration GUI, you have many drop-down fields to choose from, and the values of some drop-down fields you select can affect choices in other fields. Linking parameters provides a way to imitate the Remedy Administration GUI, by linking a field to its data source. Additionally, on the mobile device, the drop-down options are available to the user.

- 1 In Application Builder, select Params.
- 2 In Configure Parameters, click Next.
- 3 For the parameter, under Display Type, select Linked from the drop-down list. The Add and Clear buttons appear.
- 4 Click Add, then search for and select the application to which to link. This application is the data source. After you save and close the window, the linked application displays in the Component column.

---

**Note** You can select multiple parameters and select Add, to link the application to multiple parameters.

---

- 5 Repeat the procedure for any parameters you need; for example, any of the Remedy parameters your application needs.
- 6 Make any additional configuration parameter settings you need.

---

**Note** To remove a linked parameter, click Clear.

---

- 7 In the Configure Parameters window, click Save.

## Enabling the Split window

With Unwired Accelerator 8.0, the Split window is no longer included in the Application Builder capture wizard by default. The Split window enables you to add parameters for splitting rows and columns in a grid. Split rules can be defined for rows and columns; for delimiters; and for personalization adapters.

Split rules include:

- Split – split the table by all rows, a specific row number, all columns, or a specific column number. To specify a specific row or column, enter the row or column number in the text box.
- Split by Delimiter – split the rows or columns by Line Feed, Space, Comma, or Other. When you select Other, specify the delimiter to use in the text box to the right of the delimiter drop-down list.
- Personalization – enabled when you select the Variable option. Personalization adapters retrieve values from external systems or databases and automatically submit them to the application.

Personalization adapters must be registered with Mobile Web Studio. Once you select a Personalization adapter, the available methods within the adapter display in the drop-down list. Select a method to specify the parameter within the Personalization adapter to submit to the application.

See “Building applications with personalization adapters” on page 202 for more information about personalization adapters.

---

**Note** To enable the feature so that the Split window appears, you must set `SplitEnabled` to “true” in the *global.properties.xml* file. After you restart Unwired Accelerator, the Split window displays between the capture window and the Define window.

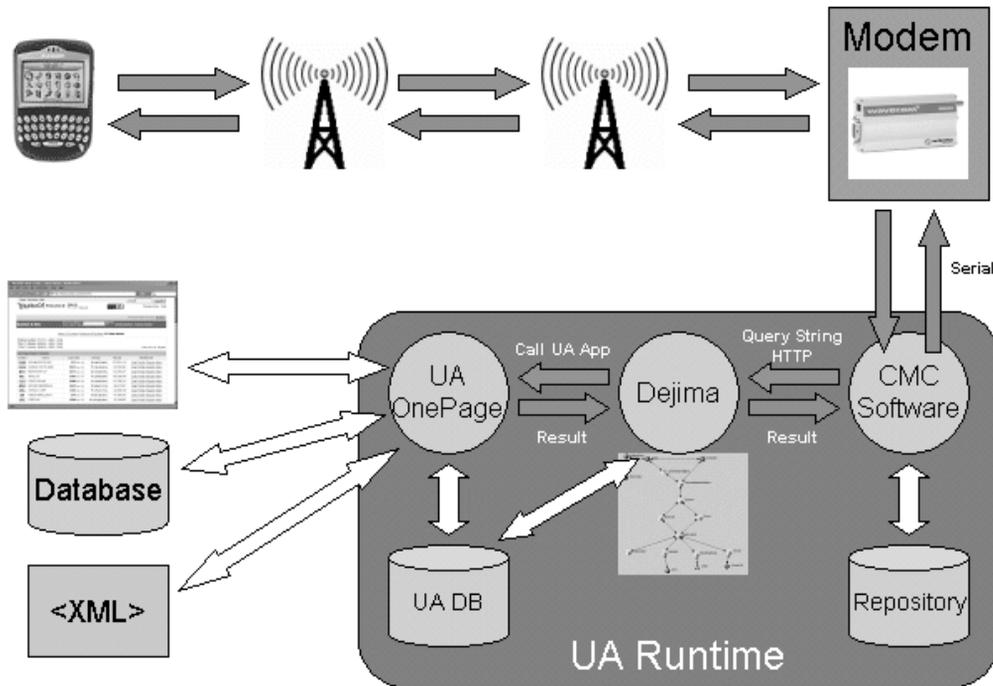
---

## Enabling Answers Anywhere (natural language search)

Answers Anywhere is a Java J2SE-based solution that allows you to design and run agent networks so that users can send simple text-based queries and receive text or HTML responses back. Applications in Approved status are used in query resolution. You can query the portal using:

- Desktop or mobile browsers
- BlackBerry devices
- E-mail
- SMS messaging on SMS-enabled mobile phones or PDAs

Figure 4-5: Answers Anywhere integration



See the *Unwired Accelerator Administration Guide* for information about configuring and managing Answers Anywhere integration with Unwired Accelerator.

The Answers Anywhere Web application contains:

- Web application SDK JAR files
- Agent Network *wired.opal* file, which is an XML file that stores the information about the UA agent network. This file is located in `%SYBASE%\tomcat\webapps\dejima\dbdsUnwired` if you are using Tomcat.
- Synonyms files, such as *application.synonym*, *call.synonym*, and *send.synonym*. These files contain synonyms for various agents and entities in the network. These files are located in:

`Tomcat SYBASE\tomcat\webapps\dejima  
\dbdsUnwired.`

- Library and properties folders, located in:  
Tomcat *SYBASE\tomcat\webapps\dejima*
- The *uadejima.properties* file, located in *SYBASE\tomcat\webapps\dejima\WEB-INF\classes* if you are using Tomcat. This file contains the property settings for configuring the e-mail server and user account the e-mail interface uses, as well as the location of the Agent Network's *wired.opal* file.
- Connection Cache – Answers Anywhere uses the portaldb connection cache.

See “Setting up Natural Language Search” in the *Mobile Applications Development Tutorial* for information about creating an Answers Anywhere application and establishing synonyms.

## Answers Anywhere Agent Network

The Answers Anywhere Agent Network is the mechanism that processes queries on behalf of UA. When a query is received by UA and passed to the Agent Network it is analyzed and broken into its constituent components in this way:

- 1 The Agent Network examines the command portion of the query to identify whether the query should be handled by the execute or send agent. This is done by comparing the command to the synonyms for each agent.
- 2 The Agent Network examines the *application\_name* portion of the query and attempts to locate the application in the UA database. To locate the application, the Agent Network inspects up to four parts of each application: the application name, application synonyms, field names and field synonyms. The field names and field synonyms only apply if the application contains grid structured data.
- 3 The Agent Network processes the remainder of the query for parameter and/or filter information.

While this sequence of events is taking place the Agent Network is also building a structured XML representation of the query, known as the Intent XML, which is returned to UA. If the Agent Network successfully understands and processes the query, the Intent XML contains all the information the UA needs to execute the desired application, including any parameter and data filtering information specified in the query. See “Intent XML” on page 190.

## Other language support

UA 8.0 Ask Anywhere integration supports simplified Chinese queries, as well as English queries. The Agent Network loaded determines what language is supported when using the Ask UA feature. UA 8.0 includes Agent Networks for English and simplified Chinese. To support simplified Chinese queries, you can change the Agent Network.

### ❖ Changing the Agent Network to Simplified Chinese

- 1 Open the *uadejima.properties* file located in `%SYBASE%\tomcat\webapps\dejima\WEB-INF\classes`.
- 2 Search for `default.opal=/dbdsUnwired/wired.opal`.
- 3 Change the `default.opal` parameter to:

```
default.opal=/dbdsUnwired_zh/wired.opal
```

## Language syntax

Answers Anywhere Agent Networks are usually designed for very specific applications. The Agent Network created for Unwired Accelerator is a very generic Agent network that is used for all Unwired Accelerator applications. Queries must conform to some basic syntax rules.

## Syntax

The basic language syntax is as follows, although you can mix the order of the elements:

*command application\_name parameters and/or record filters*

No matter how the question is constructed, you must supply at least two pieces of information: (1) the action or command, and (2) the UA application on which to apply the action. The optional parts are specifying parameters or recording filters which affects the content of an application that will be returned to the user.

Synonyms are used in commands, applications, parameters and fields. Synonyms are alternative words that the Answers Anywhere search mechanism uses to carry out the search. For example, if the field “auto” has “car,” “truck,” and “vehicle” defined as synonyms, a query like “find cars with make Acura and color blue” would include the field “auto” in its search for application data.

## Commands

The UA Answers Anywhere agent network uses two explicit commands—execute and send.

The execute agent processes queries that are requests for information that are returned to the user who created the query.

The send agent processes queries that are for collaborative efforts, such as sending the response content to a person other than the user who initiated the query. An example is “send John Doe’s phone number to jane.doe@domain.com.” In this case, the results of the query are sent to jane.doe@domain.com instead of to the user who initiated the query.

The synonyms defined for execute and send are in Table 4-1.

**Table 4-1: Synonyms for execute and send**

Command	Synonyms
execute	invoke, play, run, get, what, when, how, why, who, find
send	give

## Application Name

Application name is the name of the application in Unwired Accelerator. Note that this is not the element name in the application builder. You can also define synonyms for application names.

## Parameters and filters

Unwired Accelerator applications can have parameters to retrieve dynamic content and use filters to retrieve only the records that are of interest if the application contains grid/structured data.

The explicit word for parameter is “parameter.” If your application has more than one parameter, you need to specify the parameter display name or the display name’s synonyms for each; for example, “find cars with make Acura and color blue.”

The explicit word for filter is “filter.” If you use the explicit word for filtering, the default field name (column name) defined in the application will be used to apply the record filter. Otherwise, you can specify the field name or its synonyms and the value to filter out unwanted records. If the fields are assigned in the application, the field names can be used to identify the application. If that is the case, Unwired Accelerator will only return that field’s content in the structured data.

## Handling date queries

UA 8.0 allows you to specify dates as parameter values. For example, `get exchange rates for today`. In this example, `today` is converted to a date value, which is passed into the parameter `for` display name. For each parameter that is a date type, you must select the Calendar Date as the Display Type.

Date formats vary depending on the backend data source. You can select a JSP template for each parameter—the result of the JSP template is the value for the parameter. UA provides three default JSP templates:

- `UTCtoDayFormat` – for the date format `dd`
- `UTCtoMonthFormat` – for the date format of `mm`
- `UTCtoYearFormat` – for the date format `yyyy`

If you want a different date format for the parameter, for example `mmm` to represent the month as `Jul`, or `Aug`, or `Jun`, etc., then you must create your own JSP template. You can define the template in the Advanced Parameters window by clicking the Advanced button on the Configure Parameters window. See “Assigning JSP templates” on page 91.

If you want to query in the format `Jun 2005`, then you must write a JSP template that takes the input in that form. See “Creating application input fields” on page 143.

To create a JSP template that takes input in `mmmyyyy` form:

- 1 Create a JSP template called `UTCtoMMMYYYY.jsp`. Here is an example of the JSP code:

```
<%@ page language="java" %>
<%@ page import="java.util.Hashtable" %>
<%@ page import="java.util.Enumeration" %>
<%@ page import="java.util.Date" %>
<%@ page import="java.text.*" %>

<%
```

```
String value (String)request.getAttribute("RequestParamValue");
try
{
    if (value.length()>2)
    {
        Long stime = new Long (value);
        Date d = new Date(stime.longValue());
        out.println("Date is " + d);
        SimpleDateFormat sdf = new SimpleDateFormat ("MMM yyyy");
        value = sdf.format (d);
        out.println("Value is " + value);
    }
}
catch (Exception e) {
    out.println("Error in conversion !! " + e.getMessage());
    e.printStackTrace();
}
request.setAttribute("RequestParamValue", value);
%>
```

- 2 Save the file in  
*%SYBASE%\tomcat\webapps\onepage\portlets\templates\mobile* if you  
are using Tomcat .
- 3 In Application Builder, click Params to launch the Configure Parameters  
window.
- 4 Click Next.
- 5 In the Configure Parameters window, click Advanced. In the Advanced  
window, click Find next to the parameter you want to assign the JSP  
template to.
- 6 In the Find JSP Template window, click Search.
- 7 In the Results pane, select the UTCtoMMMYYYY template and click OK.
- 8 Save the application.

## Resolving ambiguity

Ambiguity is a state in which there is more than one possible meaning in your query. For example, if you want to query the portal for John Doe's employee information, and you enter `Show employee info for John Doe`, the network cannot determine which agent should handle the request because you did not enter "show" into the agent network as a synonym for either the `Execute` or `Send` agent.

Another example of a query that could cause ambiguity is if you have an application called "chart" and you assigned "stock" as an application synonym for chart. If you make the request `get stock`, the system sees two applications—`stock` and `chart`. When this happens the system executes the first application returned from the database.

To resolve conflicts such as the ones shown in the preceding examples, the Agent Network executes in this order:

- Exact match of application names
- Exact match of application synonyms
- Exact match of fields and field synonyms
- Substring match of application names
- Substring match of application synonyms
- Substring match of field and field synonyms

If there is ambiguity in your query and the agent network cannot resolve the query, you receive a list of applications from which you can select the one you want to use to execute the query.

If you are using multiple co-brands in your portal, Sybase recommends that you limit the number of applications you co-host within a single co-brand and use alternative co-brands to separate related applications to reduce the possibility of ambiguities.

When creating an Answers Anywhere application, click the "Check Conflicts" button to check for potential ambiguities with other applications in that particular co-brand. See "Enabling Answers Anywhere (natural language search)" on page 183.

## Intent XML

Intent XML is the information returned by Answers Anywhere. Here is an example of Intent XML:

```

<DejimaInterpretation>
  <Input><![CDATA[find info for wall]]></Input>
  <Understood><![CDATA[find info for wall]]></Understood>
  <System>
    <Execute explicit="true">
      <Application>
        <Header>
          <IdList match="info">
            <id>251</id>
          </IdList>
        </Header>
        <Parameter explicit="true"><![CDATA[wall]]></Parameter>
      </Application>
    </Execute>
  </System>
</DejimaInterpretation>

```

This example shows that the original query was “find info for wall.” Using the agent network, the Answers Anywhere runtime resolved this query using the Execute, Application, and Header agents. In the agent network for UA Answers Anywhere, the Header agent is connected to a data source that is linked to the portal database through the portaldb connection cache. The data source is used to determine whether there is an available application that matches the query. In this example, application 251 (the application ID) has been determined to be the relevant application.

Another feature of the Intent XML is the determination of parameters in the query string. The parameter XML node is present in the Intent XML whenever the Parameter agent in the agent network determines there is a parameter. Using this information, the parameter value can be passed along to the application as a parameter during application playback. Certain synonyms for parameters have already been specified in the agent network. These are:

- param
- arg

If multiple applications match the query, the Intent XML contains several id nodes as shown in the example below:

```

<DejimaInterpretation>
  <Input><![CDATA[get test]]></Input>
  <Understood><![CDATA[get test]]></Understood>
  <System>
    <Execute explicit="true">

```

```
<Application>
  <Header>
    <IdList match="test">
      <id>281</id>
      <id>211</id>
    </IdList>
  </Header>
</Application>
</Execute>
</System>
</DejimaInterpretation>
```

Both applications 211 and 281 matched the query. When at least one matching application is located, the matching value is placed in the match attribute of the IdList node. This match value is then tested against the field headers and field header synonyms to determine if a grid filter rule must be dynamically generated and executed against the application content. If the match attribute is not determined to be a field header or a field header synonym, the dynamic rule is not generated.

In the following example of a query that could cause conflict, there is an application called “chart” and “stock” is assigned as an application synonym for chart. When the request get stock is made, the system sees two applications—“stock” and “chart.” When this happens the system executes the first application returned from the database. This shows the intent XML for the query:

```
<DejimaInterpretation>
  <Input><![CDATA[get stock for SY]]></Input>
  <Understood><![CDATA[get stock for SY]]></Understood>
<System>
  <Execute explicit="true">
    <Application>
      <AppSyn>
        <IdList match="stock">
          <id>221:21:chart</id>
```

```
</IdList>
</AppSyn>
<Header>
  <IdList match="stock">
    <id>281:21:stocks</id>
    <id>211:21:stock</id>
  </IdList>
</Header>

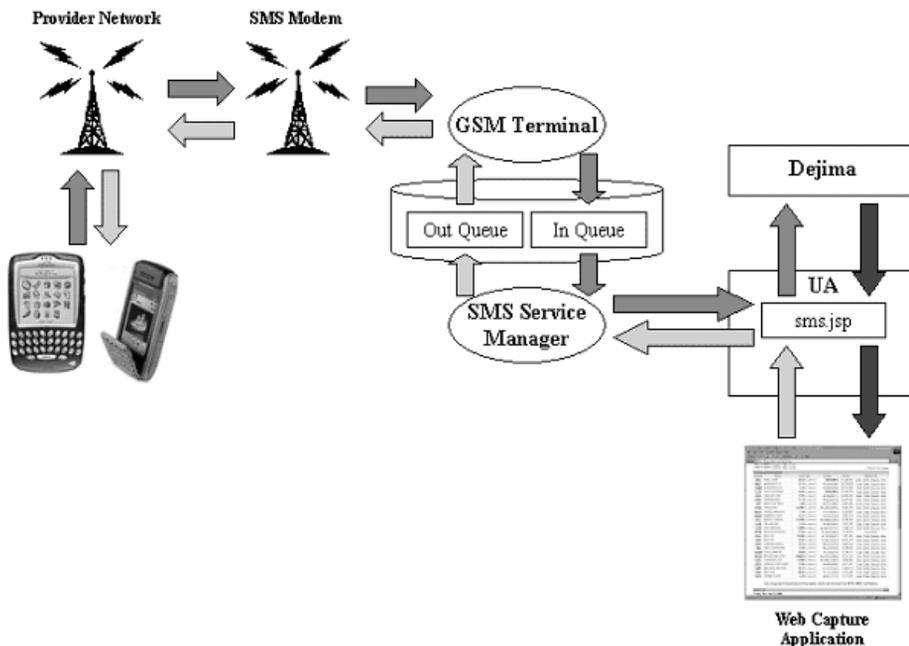
  <Parameter>
    <ParameterName>
      <IdList match="for">
        <id><![CDATA[2510:251]]></id>
        <id><![CDATA[2810:281]]></id>
        <id><![CDATA[2610:261]]></id>
      </IdList>
    </ParameterName>
    <Value><![CDATA[SY]]></Value>
  </Parameter>
</Application>
</Execute>
</System>
<Question>
  <AppSyn>
    <IdList match="stock">
      <id>221:21:chart</id>
    </IdList>
  </AppSyn>
  <Header>
    <IdList match="stock">
      <id>281:21:stocks</id>
```

```
<id>211:21:stock</id>  
</IdList>  
</Header>  
</Question>  
</DejimaInterpretation>
```

## Using SMS messaging

Users with SMS-enabled mobile phones or PDAs can send queries to a mobile phone or modem attached to a computer. Mobile modem control software on the computer then retrieves the message from the phone/modem and passes it to the Dejima components for processing. When the request is processed and a response is created, the modem control software sends the response back to the phone number that sent the request.

**Figure 4-6: SMS messaging with UA Answers Anywhere**



## CellularModemController (CMC)

The CellularModemController (CMC) retrieves SMS messages from the phone/modem and passes them to the Answers Anywhere components for processing. When the request has been processed and a response created, the CMC sends the response back to the phone number that sent the request.

See the *Unwired Accelerator Installation Guide* for information about configuring the CMC, and the *Unwired Accelerator Administration Guide* for information about configuring and managing the CMC.

---

**Note** If you set up multiple co-brands in your portal, you must also configure different instances of CMC with each CMC instance configured to access the appropriate co-brand (RID).

---

### Maintaining an access log

CMC enables you to maintain an access log that contains a subset of information for each SMS request received, such as:

- Requesting phone number
- Timestamp of request
- Request contents
- Response contents
- Response status

You can select which of these fields to record in the access log file. You can configure CMC to place the access log data in flat files or in a database file. See the *Unwired Accelerator Administration Guide* for information about configuring the access log.

### Security with e-mail and SMS based request handling

E-mail requests are serviced for all registered UA users whose “from” e-mail ID matches the registration e-mail ID in the system. SMS requests are serviced for all registered UA users whose SMS phone number matches the phone number of a user registered in the UA system.

Role-based access to the executed application applies once it is established that the user is a valid UA user. Security roles are defined when you build the application in Mobile Web Studio. Nonregistered users' requests are serviced only for applications that have the "everyone" or "guest" roles.

The administrator can also set up an explicit list of phone numbers for people who are allowed access (a "white list") to CMC services. In the same way, the administrator can set up an explicit list (a "black list") that blocks access from certain phone numbers. If the user's number is found on the allowed list, processing of the query completes, if the user's number is not found, a message is sent that indicates the user does not have permission to access that particular service.

See the *Unwired Accelerator Administration Guide* for information about configuring SMS features.

## Enabling applications and pages to support multiple languages

To enable UA applications and pages to support multiple languages simultaneously, use UTF-8 encoding. UTF-8 allows users to place applications from different languages on the same page and also allows application names in different languages.

UA applications are preconfigured with ISO encoding, which typically supports only one or more closely-related language sets.

To enable the portal for UTF-8 support:

- 1 Open *global.properties.xml* in WordPad on Windows. This file is located in `%SYBASE%\tomcat\webapps\onepage\config` if you are using Tomcat.
- 2 Search for the "use\_utf8" property and set the value to "true", which sets the portal to use UTF-8 globally. For example, change this:

```
<Property name="use_utf8" value="false" description="(true/false). true  
to enable global to set the Portal to use UTF-8" menugroup="-1"/>
```

to this:

```
<Property name="use_utf8" value="true" description="(true/false). true  
to enable global to set the Portal to use UTF-8" menugroup="-1"/>
```

- 3 Save *global.properties.xml* and close the text editor.

- 4 Set the charset property and emailCharset property to UTF-8 for each co-brand you have listed in *cobrands.xml*.

---

**Note** The *cobrands.xml* file contains a definition for each co-brand (portal). See the *Unwired Accelerator Administration Guide* for more information about co-brands.

---

- a Open *cobrands.xml* in MS WordPad.

The default *cobrands.xml* is located in  
`%SYBASE%\tomcat\webapps\onepage\config` if you are using Tomcat.

- b Set the charset property to “UTF-8”.
- c In *cobrands.xml*, set the e-mailCharset property to “ISO-xxxx-x”.

---

**Note** E-mail clients typically cannot interpret UTF-8 encoded e-mail messages. However, if your e-mail client can interpret UTF-8 encoded e-mail messages, you can set this value to UTF-8 also.

---

For example, change this:

```
<CobrandDef rid='1' name='Sybase'
  dir='/fw/cobrands/onepage-1'
  charset='ISO-8859-1'
  emailCharset='ISO-8859-1'
  emailerSetting='1' >
</CobrandDef>
```

to this:

```
<CobrandDef rid='1' name='Sybase'
  dir='/fw/cobrands/onepage-1'
  charset='UTF-8'
  emailCharset='ISO-8859-1'
  emailerSetting='1' >
</CobrandDef>
```

- 5 Set the correct character set in *styles.xml* for each co-brand you have created. The default *styles.xml* is located in  
`%SYBASE%\tomcat\webapps\onepage\fw\baseApps` if you are using Tomcat.

If you have more than one co-brand, you must change the character set in the *styles.xml* file for each one. The *styles.xml* for each co-brand is stored within that portal's co-brands subdirectory; for example, *%SYBASE%\tomcat\webapps\onepage\fw\cobrands\japanese-11*.

An example of this step is to change this:

```
<Styles
  charset="charset=iso-8859-1"
  charset_ie="charset=ISO-8859-1"
  charset_nn="charset=ISO-8859-1"
  customerCareEmail="customercare@Sybase.com"
  documentDomain="sybase.com" />
```

to this:

```
<Styles
  charset="charset=UTF-8"
  charset_ie="charset=UTF-8"
  charset_nn="charset=UTF-8"
  customerCareEmail="customercare@Sybase.com"
  documentDomain="sybase.com" />
```

## Web browsers and UTF-8

- UTF-8 encoding works successfully in Internet Explorer versions 5.0, 5.5, and 6.0. The Internet Explorer client machine must have the language font installed.
- If you are using Netscape and have set the server to use UTF-8 encoding, you must install UTF-8 fonts. MS Windows fonts typically do not work.

# Personalizing Application Content

Personalization allows you to configure application parameter input values to be filled in from adapters that extract values from other sources; for example, values that have been stored in the back-end database. When you configure an application input field to use a personalization adapter, the adapter is invoked at runtime to provide values for preconfigured key fields. The input fields of application instances that belong to different users can receive input values based on each user, providing customization of an application's content and its execution on a per-user basis.

Topic	Page
Overview	200
Creating new personalization adapter keys	201
Building applications with personalization adapters	202
Entering user-specific values in Mobile Web Studio	203
Importing and exporting personalization keys	204

---

**Note** Content personalization is also available to Portal Interface users when they select MyInfo. See the *Portal Interface User's Guide* for details.

---

## Overview

To use personalization adapters, first you create new key values for the adapter you want to use, then you create an application that provides input values using that key. Later, users can personalize the key by entering their own default value for the application's input field. For example, you have an application where you enter the postal code to see the weather for the associated area. It uses a database table adapter to provide the postal code input field. Each user can edit the key and enter their own personal postal code so the local weather displays each time he or she accesses the weather application in Portal Interface.

To use the personalization feature, select the following from the Mobile Web Studio main window:

- **Manage | Personalization** – select from the left pane to create, edit, manage, or delete personalization keys for each adapter type.
- **Build | Applications** – select from the left pane to create applications that pull input values from personalization keys.
- **Account Info | Personalize** – to manage and customize your personalization values.

## Adapter types

Unwired Accelerator provides these adapters:

- **User name/password** – the keys shown for this adapter are automatically created when a user logs in to Portal Interface or Mobile Web Studio. Users cannot create, edit, or delete keys. The key values are stored in memory and maintained only for the user's current login session. The values are reset each time a user logs in or changes their password. The key values can be used when a user logs in to Mobile Web Studio or Portal Interface, then wants to log in to an application during the same session that requires a login with the same user name and password.
- **Database table** – uses the back-end database to store keys and user values. You can create an input field with no default values, a single default value, or a drop-down list of default values from which a user can select.

## Creating new personalization adapter keys

Select Manage | Personalization from the Mobile Web Studio left pane.

The center pane lists personalization adapters extracted from the *uwp.xml* portal configuration file. The right pane lists the keys configured for the selected adapter.

The database table does not have default keys installed. To use this adapter, you must create a new key.

### ❖ Creating new keys

- 1 Log in to Mobile Web Studio using a user name with administrative privileges, such as `masuper/m8super`.
- 2 Select Manage | Personalization from the Mobile Web Studio left pane.
- 3 Select the adapter type for which you want to create a new key.
- 4 Click New on the toolbar.
- 5 When the Create New Key window displays, select or complete these options:
  - Name – enter a unique key name. The name used by applications along with the user ID to locate personalized input values for different users.
  - Type – select the type of input required for the value.
    - Text – the input field requires a text string.
    - Int – the input field requires whole numbers.
    - Select – the input field allows users to select a value from a drop-down list of predefined values.
    - Password – the input field requires a password. The value is shown as `*****` when typed in the user’s personalized section.
  - Permission – select whether the key can be edited or is read-only. Read-only is appropriate for any user name and password keys.
  - Description – enter a text description of the key.
- 6 Click OK to save your entries, and OK to confirm.

## Building applications with personalization adapters

When adding an application to a personal page, you can personalize the input field to always use a specified value. You can also create personalization keys so you can use personalization adapters with mobile devices.

You select which personalization adapter and key to use during the element definition process for Web, database, JSP, and Web services applications.

The remainder of this section describes how to use each adapter type.

### Using the user name/password adapter

---

**Warning!** This adapter does not work if you are using certificate-based authentication to access the portal (browser certificates, smart cards, biometric devices) because there is no user password. If your portal supports certificate-based user authentication, developers should use the security proxy adapter instead of the user name/password adapter.

---

Use the single sign-on adapter when you have a back-end application that shares the same user name and password as Enterprise Security, but is running on a separate machine, and is not just another application running within the same Web container.

For example, if your portal is configured to use the LDAP Authentication delegate instead of the Access Control Database (ACDB), and the LDAP delegate is pointed at your enterprise's LDAP server, you could use your PC user name/password to log in to the portal. Then you could use the same user name/password adapter key to create an application that gets you into another enterprise application that is also using the enterprise's LDAP server to authenticate users.

### Using the database table adapter

The default database personalization adapter uses the back-end database to store values.

The database adapter is built on top of two tables—`personalization_keys` holds the key values; `personalization_values` reference the `personalization_keys` and holds the value for each user.

**❖ Creating a database table adapter key**

- 1 Log in to Mobile Web Studio using a user name and password combination with administrative privileges, such as `masuper/m8super`.
- 2 Select Manage | Personalization.
- 3 From the Adapter menu, select Database Table and click New on the toolbar.

When the Create New Key window displays, complete these options:

- Name – a unique key name; for example, “Weather by postal code”.
  - Type – the type of input required for the value. Select “Int”.
  - Permission – whether the key can be edited or is read-only. Select “Editable”.
  - Description – a text description of the key; for example `Weather for selected postal code areas`.
- 4 Click OK.

## Entering user-specific values in Mobile Web Studio

This section explains how a user can personalize the values for any application that uses personalization adapter input fields.

**❖ Personalizing input values**

When you log in to Mobile Web Studio and have access to the newly created application, you can add your own personal default values for the input fields.

- 1 Log in to Mobile Web Studio.
- 2 Click Account Info at the top of the window.
- 3 When the Account Info window displays, click Personalize. You see the Personalization window.

Each line represents personalization adapter keys for any applications with personalization input fields which you can enter a default value. When you access applications in Portal Interface that use these adapter keys, your personal default value displays in the input field.

---

**Note** For users who are not allowed to access Mobile Web Studio, this information can also be entered, changed, or deleted in Portal Interface by clicking My Info and clicking Personalize.

---

To remove a key, select the X box to the right of the key's description. You cannot remove the Portal Login Name and Password keys.

To add a key, click Add.

- 4 In the Add Personalized Keys window, click Search to see a list of keys available for personalization.
- 5 From the Results pane, select the key you want to personalize and click OK.
- 6 You return to the Personalization window, where the key you selected in the previous step now appears. Click OK.
- 7 Click OK in the confirmation window.

## Importing and exporting personalization keys

When you use Mobile Web Studio to export applications that use personalization keys, any keys associated with the database adapter are exported along with the applications. When you subsequently import those applications to another portal, the database adapter keys are automatically created, and become available to personalize.

# Building Templates

Templates allow you to predefine the position and layout for applications that contain one or more elements. This chapter describes how to use the Mobile Web Studio Template Manager, which is the GUI administrative tool you use to change template information. It also describes how to apply online device templates to applications and preview the results.

Topic	Page
Overview	205
Creating and editing templates	213
Changing the default template	215
Customizing user XML element templates	216
Customizing online device templates	220

## Overview

In Unwired Accelerator, you create templates through the template manager. You can create templates for online help, HTML, JSP, XML, and XSL, using the Template Manager. You can also create online templates for mobile applications running on various device types.

In Application Builder you can apply multiple templates to a single application. Each template is associated with a particular viewer, such as a Web browser or a mobile device. You can also preview the application as it will appear on various mobile devices.

## Template Manager

Use the Template Manager to create and manage templates. Select Build | Templates from the Mobile Web Studio left pane to display the Template Manager.

The Template Manager interface includes:

- Template Manager Status menu – select a status to see templates for the selected workflow status: New, Pending, Approved, Rejected, Archived, Broken, or Deleted.
- Toolbar – displays New and Edit buttons that let you add, view, edit, or delete templates, and the Preview button that lets you preview a selected HTML template.
- Status bar – shows the current user and resource.
- Detail view – displays templates based on your selection from the Status menu, and the Filter By and Show Active Only options.
- Pop-up Template Menu – the options on this menu let you add and edit templates, and allow you to approve and activate specific templates. Only templates that are active and have an approved status can be applied to applications.

## Template Upload window

The Template Upload window is where you create and edit templates.

Click the New icon on the Template Manager toolbar, or right-click in the detail view and select New to access the Template Upload window.

In the Template Upload window, choose from:

- Save – save the template.
- Preview – preview the selected template before saving.
- Wizard – click the down arrow to select from these available templates:
  - Dashboard
  - BlackBerry Online
  - Nokia Online
  - Palm Online
  - PPC Online
  - Smartphone Online
  - WAP-HTML
  - WAP-WML
  - PDA Online

Mobile Web Studio allows you to create these types of templates by selecting from the Type drop-down list:

- HELP – online help template.
- HTML – HTML template.
- JSP – JavaServer Page (JSP) template.
- XML – XML template.
- XSL – Extensible Stylesheet Language (XSL) template. XSL allows you to apply one style document to multiple applications. XSL also lets you dictate the manner in which Web content prints, and lets you transfer XML documents across different applications.

See Template types for more information about these template types.

## Template types

This section describes the template types available.

Keep in mind that for template types of HTML/XSL, it is possible to get the template processor to invoke a JSP to generate the template content. For example, assume you have an XSL template; you have these three choices for defining the template body:

- Simply insert XSL content into the template body.
- Indicate a path to a JSP—such as `/portlets/jsp/custom/some.jsp`—that generates XSL, into the template body
- Make sure the template body starts with `<%`, and then the template body is executed as a JSP that generates XSL. In this case the template body is the JSP content.

This pattern is also available for HTML templates. This means that you can get the template body output generated via a JSP, either by pointing at a JSP or by including the JSP content directly into the template body. See “Adding JSP-related capabilities” on page 155 for more information about incorporating JSP code.

## HTML templates

All applications are wrapped in an HTML template to display in a Web browser. Other template types are available only when you are creating that type of application; that is, you can access JSP templates only when you are creating a JSP application.

When you create an HTML template that includes a table, the code must include the `<OPContent>` tag in each HTML table row description. The `<OPContent>` creates cells in which to add content. The `<OPContent>` tag uses this syntax:

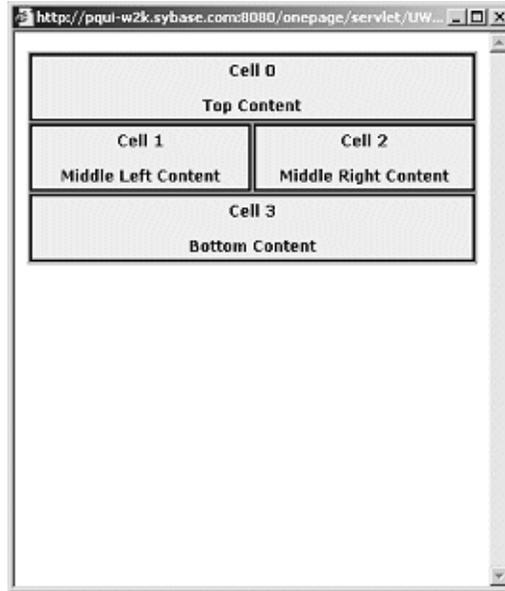
```
<OPContent id="id_number" name="position_descriptor_name"/>
```

- *id\_number* – a sequential number, starting with 0 (zero), which identifies table cells. The *id\_number* of the first `<OPContent>` tag must be `id=0` and subsequent `<OPContent>` tags must be sequential.
- *position\_descriptor\_name* – a descriptive name that specifies the cell's position within the table.

The *id\_number* and *position\_descriptor\_name* for each `<OPContent>` tag must be unique. Only content that would normally occur between the opening and closing `<body>` tags should be included in the HTML template.

This code represents a template that looks like Figure 6-1. The code between the `<STYLE>` uses the Cascading Style Sheet (CSS) classes that define the formatting of data-capable elements.

```
<STYLE TYPE="text/css">
.gridRowHeader{background:#DEDEB9;padding:4px;padding-bottom:0px;border:1px
solid #FFF;font-size:13px;font-family:verdana;color:#336699;font-weight:bold;}
.gridRowA{background:#EEF9FF;padding:4px;padding-bottom:0px;border:1px solid
#FFF;font-size:13px;fontfamily:verdana;}
.gridRowB{background:#FFFFFF;padding:4px;padding-bottom:0px;border:1px solid
#FFF;font-size:13px;font-family:verdana;}</STYLE>
<table width="100%" border="1" cellspacing="0" cellpadding="0">
<tr align="center" valign="middle">
<td height="10" colspan="2"><OPContent id="0" name="Top Content"/></td></tr>
<tr align="center" valign="middle">
<td height="10" width="50%"><OPContent id="1" name="Middle Left Content"/></td>
<td height="10" width="50%"><OPContent id="2" name="Middle Right Content"/>
</td></tr>
<tr align="center" valign="middle">
<td height="10" colspan="2"><OPContent id="3" name="Bottom Content"/>
</td></tr></table>
```

**Figure 6-1: HTML example template**

## HELP templates

Help templates are HTML files that you save to the database. They display when a user clicks the Help icon on an application's title bar in Portal Interface.

When you save a new element, one of the fields asks for the Help URL. You can either enter the URL of a Web page or select a Help template from the database.

The HTML code shown next represents a template that looks like Figure 6-2.

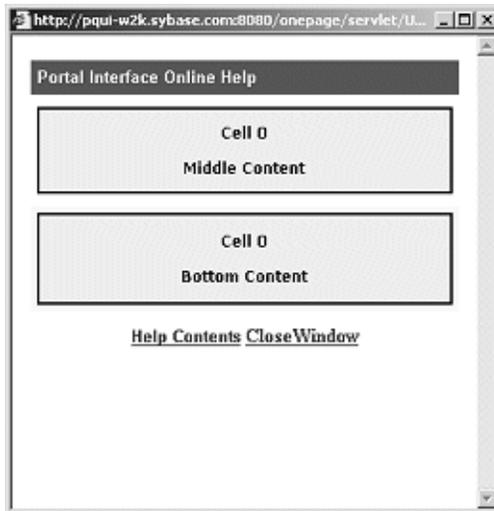
```
<script language="JavaScript">
<!--
function openWindow(URL)
{
window.open(URL,"PopUp","scrollbars=yes,resizable=yes,width=400,height=400";
}
function Winload()
{
}
//-->
</script>
<body bgcolor="#FFFFFF" text="#000000">
<table width="100%" border="0" cellspacing="5" cellpadding="5" height="190">
```

```

<tr>
<td colspan="2" height="27" bgcolor="#336666">
<p><font color="#FFFFFF" size="2" >
<b><font face="Arial, Helvetica, sans-serif">Portal Interface Online
Help</font></b></font></td></tr>
<tr height="16">
<td bgcolor="#FFFFFF" colspan="2" height="20">
<OPContent id="0" name="Middle Content"/>
<b><font face="Arial, Helvetica, sans-serif" size="2"></font></b>
<hr noshade width="100%" size="1"></td></tr>
<tr>
<td colspan="2" bgcolor="#FFFFCC"><OPContent id="0" name="Bottom Content"/>
<p><font face="Arial, Helvetica, sans-serif" size="2"></font></p></td></tr>
<tr bgcolor="#FFFFCC"><td width="100%" bgcolor="#FFFFFF">
<p align="center"><b>
<a href="JavaScript:openWindow('index.html')">
<font face="Arial, Helvetica, sans-serif" size="2">Help Contents</font></b>
<imgsrc="images/spacer.gif" width="122" height="6">
<a href="javascript:window.close();"><b>CloseWindow</b></a></font></td></tr>
</table>

```

**Figure 6-2: HELP template**



## JSP templates

You can create Web applications and deploy them into the portal as an application. Alternatively, you can use JSP files for small reusable pieces of functionality. The JSP approach requires you to create a *.jsp* file with embedded Java and HTML code. Servlets or other JSPs can invoke the JSP code by referring to the file.

JSP-based templates provide a way to manage JSP code across applications. You can create templates without managing individual files in the file system.

A JSP template has a name and a definition, which are saved and managed by Mobile Web Studio. Once you create a template, you can populate the template with applications. You can include those applications in other applications. When you change a template definition, those changes are applied universally to all the associated applications of that template.

The JSP code shown next represents a JSP template that displays the current server-side time and date.

```
<%@ page language="java" contentType="text/html;
charset=UTF-8" %>
<%@ page import="java.util.*"%>
<H1>Server Side Time rendered by JSP Template</H1>
The current time is
<%
java.util.Date date = new java.util.Date();
out.println(date);
%>
```

## XML templates

Mobile Web Studio allows you to create an XML template.

## XSL templates

Mobile Web Studio allows you to create an XSL template (XSLT) that you can use for XML applications and elements. Code the XSLs for data-capable elements to use the Unwired Accelerator internal document type definition (DTD).

---

**Warning!** The only XSL stylesheet supported to create XSL templates is:

```
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
```

```
version="1.0">
```

If you use a later stylesheet, a parsing error is generated because of the version of Xerces used.

---

The following XSL code is a sample of an XSL template for a data-capable user interface XSLT.

```
<?xml version="1.0"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"version="1.0">
  <xsl:template match="/">
    <TABLE WIDTH="100%" BORDER="1" BORDERCOLOR="#000000">
      <xsl:for-each select="//Record">
        <xsl:text> </xsl:text>
        <TR><xsl:for-each select="Field"><xsl:text></xsl:text>
          <TD>
            <xsl:for-each select="./text()">
              <xsl:value-of select="."/>
            </xsl:for-each>
          </TD>
        </xsl:for-each>
        <xsl:text> </xsl:text>
      </TR>
    </xsl:for-each>
  </TABLE>
</xsl:template>
</xsl:stylesheet>
```

---

**Note** When you apply a UI XSL template to a Web application, you must select the “Enable Grid Rules” box, otherwise, the specified template is not applied.

---

## Device Manager

Keep in mind you use the Device Manager to customize offline client templates for BlackBerry devices. Select Manage | Devices from the Mobile Web Studio left pane to display the Device Manager.

See the *Unwired Accelerator Administration Guide* for information about customizing the BlackBerry client device templates, and then creating a new BlackBerry offline client. The new BlackBerry offline client must be distributed to BlackBerry device users.

## Creating and editing templates

This section describes how to create, edit, and delete templates. There are also procedures for changing a template's status. See "Customizing online device templates" on page 220 for specific information about creating online device templates.

---

**Note** You can also create templates on the fly when you create and edit aggregated applications. Do not create templates in this way when you create secure applications—you may receive a JavaScript error. See Chapter 3, "Developing Applications," for details.

---

### ❖ Creating a template

- 1 Select Build | Templates from the Mobile Web Studio left pane.
- 2 Click New on the Template Manager toolbar to launch the Template Upload window.
- 3 If appropriate, click the down arrow next to Wizard to select from the list of templates.
- 4 Complete these fields:
  - Name – enter a name for the new template. For example, you might want to name a template "2-4 Basic" for an application with two columns and four elements.
  - Roles – assign the appropriate roles to the template. When you assign a role to a template, all users assigned that role can access the template in Mobile Web Studio.

---

**Note** Once you add an application to a page, the roles assigned to the page and applications supersede the roles assigned to the underlying template. A template's roles pertain only to roles that can access templates in Mobile Web Studio.

---

- Type – select from HELP, HTML, XML, JSP, or XSL.
- Status – select the template's status. You can only add applications and elements to approved templates.
- Code – enter either:

- The code that defines this template. Templates are stripped of header tags, so include only the content that normally resides between the opening `<BODY>` tag and closing `</BODY>` tag.
  - A filename that contains the code for the template; for example: */portlets/templates/mobile/mobile\_device\_template.jsp*.
- 5 If this is an HTML or HELP template, click Preview to see the template preview in a separate window.
  - 6 Click Save to save the new template definition.
  - 7 Click OK in the confirmation popup window.
  - 8 Click Close to exit the window.

### ❖ Editing a template

- 1 In the Template Manager, select the status of the template you want to modify.
- 2 Double-click the template listing in the detail view to open the Template Editor window.
- 3 Change the Status, Type, Assigned Roles, or Code, and activate or deactivate the template.
- 4 When you are satisfied with your changes, click Preview to review your modifications if this is an HTML or HELP template.
- 5 Click Save to save your changes under the same name, or click Save As to save the template under a different name.
- 6 Click Close to exit the Template Editor.

### ❖ Changing workflow status

If you are using the Mobile Web Studio's workflow feature, you can change a template's approval status as it moves through the workflow process.

- 1 Right-click a template in the Template Manager's detail view, select Approval Status from the pop-up Template Menu, then select the appropriate status; or
- 2 Select a template in the Template Manager's detail view, click Edit on the toolbar, select a new status from the Status drop-down list in the Template Editor window, then click Save.

**❖ Changing a template's active status**

You cannot add an application to a template unless the template is approved and active.

- 1 Right-click a template in the Template Manager's detail view, select Active Status from the pop-up Template Menu, then select the appropriate status; or
- 2 Select a template in the Template Manager's detail view, click Edit on the toolbar, select or deselect Active, then click Save.

**❖ Deleting a template**

When you delete a template, it remains in the database, and displays in the detail view with a status of "Deleted." To mark a template as deleted:

- 1 Right-click a template in the Template Manager's detail view, select Approval Status from the pop-up Template Menu, then select Deleted; or
- 2 Select a template in the Template Manager's detail view, click Edit on the toolbar, select Deleted from the Status drop-down list box in the Template Editor window, then click Save.

To change a template's deleted status, select Deleted from the Template Manager's Status menu, right-click the template in the detail view, select Approval Status, then select a new status.

## Changing the default template

Template1 is the default template automatically assigned to applications (see Chapter 3, "Developing Applications"). This system-assigned template is the value specified for the DefaultTemplateName property in the *global.properties.xml* portal configuration file, which is located in `%SYBASE%\tomcat\webapps\onepage\config` if you are using Tomcat, `%JAGUAR%\Repository\WebApplication\onepage\config` if you are using EAServer.

You can change the default to any template you want. For example, if you want every application to use the same header logo or footer, you can create a template with the logo, and enter that template's name in *global.properties.xml*.

---

**Note** When you change the template name in *global.properties.xml*, the new default template is used by all resource IDs in Mobile Web Studio.

---

❖ **Changing the default application template**

- 1 Create and save a new application template.
- 2 Use any text editor to open *global.properties.xml*.
- 3 Search for `DefaultTemplateName` and replace the value "Template1" with the name of the aggregated application template you created.
- 4 Save the file.
- 5 Restart the application server.

## Customizing user XML element templates

Unwired Accelerator allows you to customize XSL and XSLT templates on a per-user basis.

When you create XML applications, you specify an Extensible Style Language Transformation (XSLT) template for content transformation and a user interface (UI) template for presentation.

The XSLT template contains the language used in XSL style sheets to transform XML documents into other XML documents. The UI template specifies how to present the application; that is, shading, borders, and so on. You can customize these templates on a per-user basis for XML applications

❖ **Creating an XML sample application**

- 1 Log in to Mobile Web Studio.
- 2 Select **Build | Applications** from the Mobile Web Studio left pane and select **New** from the Application Manager Status menu.
- 3 Right-click in the detail pane and select **New Aggregated Application**.
- 4 When the Application Builder displays, click the down arrow beside the **Add** button and select **XML Element** from the menu.

You see the XML Element Definition window.

- 5 Enter the following sample input:

XML URL –

```
http://p.moreover.com/  
cgi-local/page?c=Biotech%20news&o=xml
```

Content XSLT – `moreover_content_xsl`

UI XSLT – `pink_ui_xslt`

- 6 Click Preview.

The content has a pink background, as specified in the UI XSLT. Any user who accesses an XML application that uses the specified UI XSLT sees a pink background.

- 7 Click Next until you reach the window where you enter the Element Name.
  - 8 Enter “XML\_sample” and click Finish to save the new element.
  - 9 When the Application Builder reappears, click Save.
  - 10 On the Finish window, enter an application name and select the J2EE roles:
    - Application Name – enter “XML\_sample.”
    - J2EE Roles – click Add All.
  - 11 Click Finish.
  - 12 When you see the prompt “Application successfully saved,” click OK, then click Close to exit the Application Builder.
  - 13 In the Application Manager, select New from the Status menu. You see the XML\_sample application.
  - 14 Approve and Activate the application by right-clicking the XML\_sample application and selecting:
    - Active Status – Active.
    - Approval Status – Approved.
- The application now displays under the Approved status.

❖ **Adding the XML\_sample application to a page**

You have created an XML\_sample application that uses the XML\_sample element and incorporates the pink\_ui\_xslt user interface. Now add the application to a page.

- 1 Select Build | Pages from the Mobile Web Studio main window and click New on the Page Builder menu bar.
- 2 When the Page Builder appears, click Add.
- 3 When the Search window appears, click Search.
- 4 Click XML\_sample in the Results pane and click Add.
- 5 In the Page Builder, click Save As.
- 6 In the Save Page window:
  - a Enter “XML\_sample” for Name.
  - b Click Add All for J2EE Roles.
  - c Click OK.
- 7 When the “Page saved” prompt appears, click OK, then click Close to exit the Page Builder.
- 8 At the Mobile Web Studio main window, select New from the Manager Status menu.
- 9 Right-click XML\_sample in the detail view and select Status | Approved from the pop-up. When the “Page saved” prompt appears, click OK.
- 10 Select Approved from the Page Manager Status menu.
- 11 Click XML\_sample in the detail view and click Update from the Page Manager menu bar.
- 12 In the confirmation pop-up, click OK.

Portal Interface now has the page available for users.

❖ **Creating a user-specific UI XSLT template**

- 1 Select Build | Templates and select Approved from the Template Manager Status menu.
- 2 Right-click the pink\_ui\_xslt template in the detail view and select Edit from the pop-up.
- 3 When the Template Editor appears, locate this line in the Code window:

```
<TD BGCOLOR="#ffccff">
```

- 4 Change the value from “#ffccff” to:

```
#ccffcc
```

- 5 Click Save As.

- 6 When the Save Template As dialog box displays, enter:

```
green_ui_xslt
```

Click OK.

- 7 When you see the prompt “Template saved,” click OK, then click Close to exit the Template Editor.

## Enabling different templates for the same element

To apply different templates to the same element, create the *xslinfo.properties* file and populate it with new template and user information.

This file should have the following content and structure:

```
<userid>.<application id>.ui=<xslt file>
<userid>.<application id>.content=<xsl file>
```

where:

- *userid* – is the login name of the user.
- *application id* – is the ID of the application for which you want to specify a user-specific UI or content XSLT. To determine an application’s ID:
  - a In Mobile Web Studio, select Build | Applications, then select Approved from the Application Manager Status Menu. Right-click the XML\_sample application in the detail view and select Properties.
  - b When the Application Properties window displays, write down the ID value, then click Close.
- *content* or *UI* file – enter the user-specific template for the application.

For example, to set up the file for the XML\_sample application, enter this information:

```
bob.231.ui=pink_ui_xslt
cindy.231.ui=green_ui_xslt
```

where *pink\_ui\_xslt* is the name of the presentation file for Bob, and *green\_ui\_xslt* is the name of the presentation file for Cindy.

Place the *xsltinfo.properties* file in the `%SYBASE%\tomcat\webapps\onepage\config` directory if you are using Tomcat. If you are using EAServer, place the file in the `%JAGUAR%\Repository\WebApplication\onepage\config`, then restart EAServer.

When each user that is specified in *xsltinfo.properties* adds the XML\_sample page to their portal view, Bob will see the XML\_sample application with a pink background and Cindy will see the XML\_sample application with a green background.

To create user-specific content, follow the same procedures in this section, but create a new content template and specify the two different content files in the *xsltinfo.properties* file; for example:

```
bob.231.content=moreover_content_xsl
cindy.231.content=new_content_xslt
```

You can also provide different presentation and different content for different users in this file.

```
bob.231.ui=pink_ui_xslt
bob.231.content=moreover_content_xsl
cindy.231.ui=green_ui_xslt
cindy.231.content=new_content_xslt
```

The first time you create *xslt.properties*, you must shut down and restart EAServer to initialize the user-specific settings. When you make additions, changes, or deletions to the file the portal is updated automatically with the new information; that is, you do not need to restart the application server.

## Customizing online device templates

You can customize the presentation of data on your mobile device by using the Mobile Application Template Customization window in Mobile Web Studio. See the *Unwired Accelerator Administration Guide* for additional information.

### ❖ Creating online device templates

You can create new online device templates or modify existing templates. You can create multiple version, such as several versions of Nokia Online device templates.

- 1 Select Build | Templates from the Mobile Web Studio left pane.

- 2 Click New on the Template Manager toolbar to launch the Template Upload window.
- 3 Click the down arrow next to Wizard to display the list of online templates.
  - Dashboard
  - BlackBerry Online
  - Nokia Online
  - Palm Online
  - PPC Online
  - Smartphone Online
  - WAP-HTML
  - WAP-WML
  - PDA Online
- 4 From Template editor, select Wizard. The `<device>` Online Application Template Customization window launches.
- 5 On the Mobile Application Template Customization window, modify these tabs (the options may vary slightly, depending on the device):
  - `<device>` device tab** Use the Device tab to name the device template, such as `newNokiaOnline`, and assign user roles.
  - Appearance tab** Use this tab to customize the properties of the presentation for your mobile device.e:
    - Default Properties
      - Font – select from the drop-down list.
      - Font size – enter the desired font size in pixels.
    - Header Properties – apply only to the header row
      - Background color – select the color box to the right to see a selection of colors.
      - Text color – select the color box to the right to see a selection of colors.
    - Row Properties – applies to all rows of data (not the header row)
      - Text color – select the color box to the right to see a selection of colors.

- Alternate row color – indicate whether the alternate rows should appear in another color.
- Alternate row color properties – applies to all other rows of data (not the header row), if alternate row color is enabled.
  - Even row background color – select the color box to the right to choose the even row background color.
  - Odd row background color – select the color box to the right to choose the odd row background color.
- Row color properties – select the color box to the right to choose the row color background if you did not choose alternating colors.
- Paging Properties
  - Number of rows in a page – enter number of rows
- CGI Parameter Properties
  - Parameter Input Field Size – enter number of characters
  - Button Text – enter button text, such as OK
- Advanced
  - Add Filter Rules

**Filter tab** This tab appears only if you selected Add Filter Rules in the Appearance tab. Select Enable Filter For List Data.

- Filter Value in Between String
- Filter Value in End of String

**Preview tab** Use this tab to preview the data presentation. The preview is not an exact representation of what you will see on your mobile device.

6 Click Finish to save your template.

#### ❖ Applying a device template to an application

- 1 In Application Builder, select the Device Type from the drop-down list—Portal, BlackBerry Online, Nokia Online, Palm Online, PPC Online, SmartPhone Online, WAP-HTML, WAP-WML, or PDA Online.
- 2 Click Template.
- 3 In the Find Template window, click Search. A list of available templates displays in the Results pane.

- 4 In the Results pane, select the desired template, such as `newNokiaOnline`, and click Select.
  - 5 In Application Builder, either preview the application template, or click Save and OK to confirm. The template is applied to the application, and dictates how the data is presented on the device type.
  - 6 Repeat the process to associate other device templates with the selected application.
  - 7 Click Close to exit Application Builder.
- ❖ **Previewing application with applied device template**
- 1 In Application Builder, after selecting the template, and click Preview. The application displays in a pop-up window. Any links and drill-down capability are enabled.
  - 2 Close the window when finished.



# Using Unwired Accelerator Alerts

This chapter describes how to use Unwired Accelerator's agent technology to automate the processing of application content.

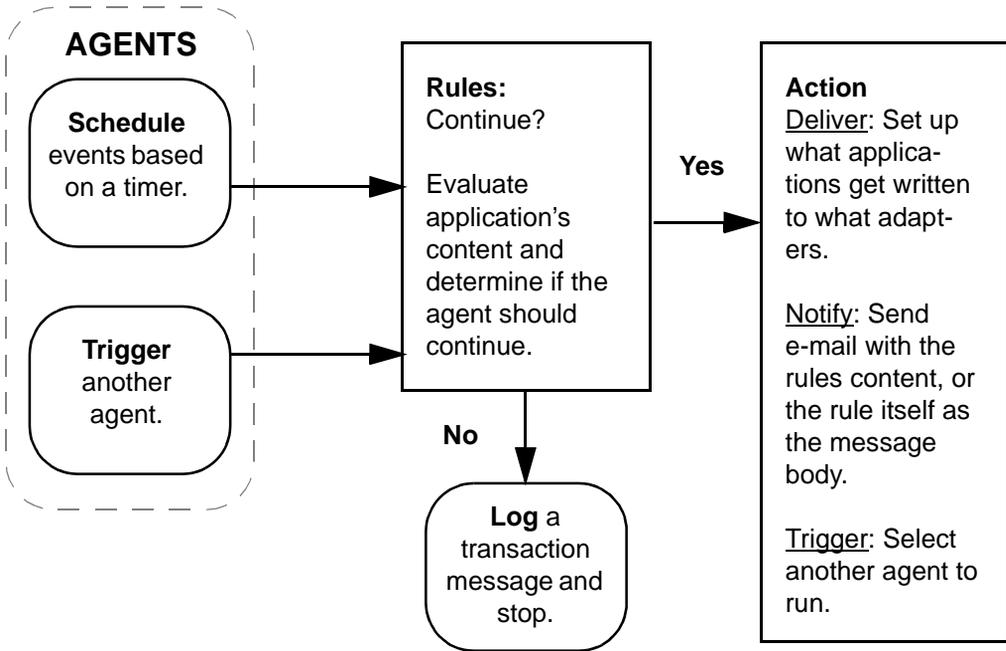
Topic	Page
Overview	225
Alerts	227
Using agents	241
Using adapters	255
Using agent servers	259
Viewing transaction logs	261

## Overview

Agents are software components that act on behalf of users or other agents to gather or process information in the background. Typically, an agent is given a small, well-defined task.

Unwired Accelerator agents allow you to schedule or externally trigger automatic processing of an application's content. You can also create agent rules that evaluate an application's content based on specified criteria. If an application's content matches the criteria, certain actions take place, such as writing the application's content to an e-mail message, to a database, or to disk, which is accomplished using an adapter. If the application's content does not match the criteria, a transaction message is written to a log.

Figure 7-1: Agents process flow



Agents are executed by:

- A timer triggering the agent’s execution based on the agent’s schedule
- One agent triggering another agent’s execution
- Selecting the Run option to execute any agent once

When an agent runs, the system checks the agent’s rules—if there are any—to determine if the agent should continue running. If the criteria specified in the rules has been met, the agent’s actions execute.

To write an application’s content to a file, an e-mail message, or a database table, you create an adapter. When the agent is triggered, the adapter interface writes the associated application’s content to a specified destination.

You can also assign an agent to a specific agent server to run that agent. Although you can create new agent servers, a default agent server is installed with Unwired Accelerator. If an agent is unassigned and there is more than one agent server, the first available agent server runs the agent. If only the default agent server exists, all agents run on that server.

## Alerts

Alerts allow you to create user notifications based on an application's conditions. For example, you can send an e-mail when the price of a specific stock reaches a specified value.

This section describes how to:

- Create alerts
- Establish alert rules and schedules
- View alert details
- View the generated alert in an e-mail application or on a mobile device

When you create an alert, the corresponding agent is seen in Mobile Web Studio. Alerts have these attributes:

- Rules – specify the application content that triggers an alert.
- Schedule – specifies when the agent should check the application for alert-producing content.

Alerts are saved based on user name. One alert (agent) definition is created per user, and is the unique ID for this user's agent.

## Requirements and restrictions

The following list describes requirements and known limitations of the alert functionality:

- Alerts require that you use Portal Interface with Internet Explorer 5.5 or 6.0. Netscape support is not currently available.
- You can define alerts only on applications created in Mobile Web Studio; that is, you cannot define alerts on personal applications that are created in Portal Interface.
- To define alerts on an application, you must select the Alert option on the Presentation tab when you save the application in Mobile Web Studio.
- Once the Alert option is selected and users set rules on that application in Portal Interface, removing (unselecting) the Alert option on the application in Mobile Web Studio has no effect. That is, you cannot roll back alerts that have been defined on an application by existing users.

- The Portal Interface alert capability uses the agent functionality of Mobile Web Studio. After an alert is defined on an application, an agent is created on the application by a user or by defining a schedule in the Alerts tab of MyInfo functionality. An agent is created in the background with the user's name. If an agent exists in Mobile Web Studio with the same name as a user, the existing Mobile Web Studio agent is overwritten.
- When Portal Interface users who have defined alert rules or schedules are deleted or disabled in Mobile Web Studio the administrator should delete or disable the operation to remove the alert agent associated with the user if one exists.

❖ **Creating alerts**

This example procedure shows you how to create an alert in Mobile Web Studio.

- 1 Use the sample database application (sampledb) included with the product.
  - a Log in to Mobile Web Studio.
  - b Select Applications from the Build menu on the left pane.
  - c Click New to access the Application Builder.
  - d When the Application Builder displays, click the down arrow to the right of the Add button and select Database Element.
  - e For this example, when the Mobile Web Studio – New Element window appears, select Connection Cache.
  - f Complete the remaining fields:
    - Conn Cache Name – name of the connection cache for the sample database; for this example:  
`sampledb`
    - SQL Query String – enter this SQL statement:  

```
select c.lname, p.name, si.quantity, si.id, si.prod_id
from customer c, sales_order s, sales_order_items si, product p
where c.id = s.cust_id and s.id = si.id
and si.prod_id = p.id
```
  - g Click Preview to see the data that corresponds to the SQL query you entered.
- 2 Click Next.
- 3 On the next two windows (Define, and Filter), click Next.

- 4 In the Window Preview, enter `Sales Update` for the Element Name, then click Finish.
- 5 In the Application Builder, click Save and complete these options on the Web Studio – New Application Finish window:
  - a Content tab – enter `Sales Update` for the application name.
  - b Roles tab – click Add All to add all roles to the Assigned Roles list.
  - c Presentation tab – select Alert, Display Within IFrame, and No Popup. Accept the default entries shown for the other options.
  - d Click Finish.
  - e In the confirmation pop-up window, click OK.
- 6 Click Close to exit the Application Builder.
- 7 Select New from the Application Manager Status menu.
- 8 Right-click the new Sales Update application in the right pane and select Approval Status | Approved.  
In the confirmation pop-up window, click OK.
- 9 Add the new application to a page.
  - a Select Pages from the Web Studio Build menu.
  - b Select New from the Page Builder Status menu, then click New.
  - c When the Page Builder displays, select the full layout icon (the far right layout icon), then click Add.
  - d When the Search window appears, click Search.
  - e Select the Sales Update application in the Results pane and click Add.
  - f In the Page Builder, click Save and complete these options:
    - Name – enter `Sales`.
    - Type – select Catalog from the drop-down list.
    - Active – verify that this option is selected.
    - Roles – click Add All to add all roles to the Assigned Roles list.Click OK.
- 10 When you see the confirmation that the page was saved, click OK.
- 11 Click Close to exit the Page Builder.

- 12 On the Mobile Web Studio main window, select New from the Page Builder Status menu, right-click the Sales page, and select Status | Approved.

When you see the message that the page was saved, click OK.

- 13 Make the new page and application available to Portal Interface users.
  - a Select Approved from the Page Builder Status menu.
  - b Right-click the Sales page in the detail pane and select Update Users.
  - c When you are prompted to reconfirm that you want to perform an update operation, click OK.
  - d When a message displays that the update operation succeeded, click OK.

- 14 Log out of Mobile Web Studio.

- 15 Alerts do not start until you set up rules for the alert in Portal Interface. To set up rules for the application you created in the previous steps:

- a Log in to Portal Interface.
- b Click Manage Pages.
- c In the Add Page Group window, enter `Information` as the page group name, and click Done.
- d Click Add Page.
- e In the Add Page window, add the pre-made catalog page “Sales,” and click Done.
- f In the Sales page, click the “!” icon in the Sales Update application.
- g In the Alert window:
  - Name – select Playback from the drop-down list.
  - Rule Data – enter the text you want to receive as a message in the alert. For this example: `Sales data update`.
  - Description – enter a brief description of the alert. For example: `Sales data has been updated`.
  - Under Rules:
    - Rule Relationship – select “and”.
    - Select Fields By – select “label”.
    - Execute Alert When – select “field3”, “is greater than”, 25 in the three columns, then click +.

Enter another rule using the same values for Rule Relationship and Select Fields By. Execute Alert When – select “field1”, “equals(text)”, enter Devlin, then click +.

- h Click Done.
  - i In the confirmation pop-up, click OK.
  - j Click Cancel to close the Alert window.
- 16 Set up a schedule for the alert:
- a Click MyInfo, and click the Alert tab.
  - b In the Alert window, select Trigger Alert Based on a Schedule.
  - c Select the Start Time, Recurrence, and Range.
  - d Select No End Date.
  - e Send Alerts To – enter the e-mail address where you want to receive the alert.
  - f Click Done.
  - g Click Start.
- 17 Log out of Portal Interface.
- 18 Log in to Mobile Web Studio to create a database update application, which can change the quantity of the Tee Shirts for Devlin in the `sampledb` and trigger an alert.
- a Select Applications from the Build menu on the left pane.
  - b Click New to access the Application Builder.
  - c When the Application Builder displays, click the down-arrow to the right of the Add button and select Database Element.
  - d For this example, when the Database Element Definition window appears, select Connection Cache.
  - e Complete the remaining fields:
    - Conn Cache Name – name of the connection cache for the sample database; for this example:  
`sampledb`
    - SQL Query String – enter this SQL statement:

```
update sales_order_items set quantity = @OP["quantity"]="13"]
```

```
where id = @OP["id"="-1"]  
and prod_id = @OP["prod_id"="-1"]
```

- 19 In the Split, Define, Filter windows, and Database Element windows click Next.
- 20 In the Location: Database Element window, check all three variables (prod\_id, id, and quantity).
- 21 In the Preview Window, in Element Name, enter:
  - Element Name – Devlin
  - id: – 2001
  - prod\_id – 300
  - quantity – 25Click Finish.
- 22 In the Application Builder, click Save and complete these options in the Finish window:
  - a On the Content tab, enter Devlin for the application name.
  - b On the Roles tab, click Add All to add all roles to the Assigned Roles list.
  - c On the Presentation tab, select Alert, Display Within IFrame, and No Popup. Accept the default entries shown for the other options.
  - d Click Finish.
  - e When you see the confirmation message that the application was saved successfully, click OK.

## Creating an SMS alert using Mobile Web Studio

This section shows how to create an alert for a Web element that is created from Mobile Web Studio instead of Portal Interface. The alert results are sent to a telephone or mobile device, so the data that is sent must be shortened as much as possible. This example illustrates how to send an alert to a Short Message Service (SMS) subscriber.

SMS is a service that lets cellular phone users send and receive short messages, generally 140 to 160 characters, depending on the cellular phone.

Unlike paging, but similar to e-mail messaging, short messages are stored at SMS centers, which lets you retrieve messages later if you are not available to receive them immediately. SMS messages travel to a cellular phone over the system's control channel, which is separate from the voice channel.

---

**Note** SMS was introduced in the Global System for Mobile Communications (GSM), a worldwide standard for digital wireless mobile phones. SMS is now supported by all other digital-based mobile communications systems.

---

The procedures are:

- “Creating an alert for SMS delivery” on page 233
- “Viewing the alert (telephone or mobile device)” on page 235

❖ **Creating an alert for SMS delivery**

This procedure describes how to create a stock quote alert for SMS delivery. Since 128 characters is the maximum size for short messages, you must shorten the data to be delivered from eight fields to the one pertinent field.

- 1 Log in to Mobile Web Studio.
- 2 Select Applications from the Build menu in the left pane, select New from the Application Manager Status menu, and click New to launch Application Builder.
- 3 Click the arrow to the right of the Add button and select Web Element. You see the New Web Element window.
- 4 In Location, enter a URL to capture grid content; for example: `http://finance.yahoo.com`, and click Go.
- 5 Scroll down the page, and click the “Most Actives” link under Top Stories.
- 6 In Capture Strategy, select Grid from the drop-down list, and click Next.
- 7 Click Select to select the grid that includes “Last Trade” and click Next.
- 8 In the Split and Define windows, click Next.
- 9 In the Filter section, set up a rule to exclude most records. Since the stock quote is sent to a telephone or mobile device, which is limited to 128 characters, you want to send only the pertinent data. In Filter:
  - a Create a rule that includes only the Symbol, Name, Last Trade, and Change fields.  
Click Add.

- b Create a rule to include the record with “MSFT” (Microsoft).  
Click Add.
  - c Click Next to bypass the Parameter window.
- 10 In Element Name, type `MS stock` and click Finish.
- 11 In Application Builder, click Save, and configure the application:
- a In the Content tab, enter `Quick Stocks` as the name of the application.
  - b In the Roles tab, click Add All to add all roles to the application.
  - c In the Presentation tab, click Alert, Display Within IFrame, and No Popup.

---

**Note** You can also use the Application properties window to add an alert, after you have saved the application.

---

- d In Window Preview, you can view the format.
  - e Click Finish.
  - f In the confirmation pop-up, click OK.
- 12 Click Close to close Application Builder. The new Web element displays in the right pane.
- 13 On the Mobile Web Studio main window, select New from the Application Builder Status menu, right-click the Quick Stocks application, and select Approval Status | Approved.
- When you see the message that the application was saved, click OK.
- 14 Make the new application available to Portal Interface users:
- a Select Approved from the Application Builder Status menu.
  - b Right-click the Quick Stocks application in the detail pane and select Update Users.
  - c When you are prompted to reconfirm that you want to perform an update operation, click OK.
  - d When a message displays that the update operation succeeded, click OK.

## Viewing the alert (telephone or mobile device)

When the data matches the rule you established, in this case the stock price crossing a certain threshold, the alert is sent to your telephone or mobile device. For example, you may receive a message similar to this:

```
Last Trade: 17.11
```

### ❖ Creating rules and schedules from Mobile Web Studio

Use this procedure to create a sample rule and schedule for an alert using the Agent Manager.

- 1 In Web Studio, select Agents from the Manager menu in the left pane, then click New to launch Agent Builder.
- 2 Select the Schedule tab and establish a schedule for the rule:
  - a Click the box next to “Trigger Agent based on a Schedule” to select it.
  - b For Time, use the option controls to specify the start time. For this example, use the current time
  - c For Recurrence, use the option controls to select the recurrence unit (such as seconds, minutes, hours) and interval. For the tutorial, select Minutes as the unit, and 5 as the interval. The alert schedule recurs every five minutes.
  - d For Range, enter today’s date in MM/DD/YY format as the start date. For convenience, use the calendar icon to select the month and date.  
  
For the end date, click “No end date.” Alternatively, you can click “End by” and specify an end date.
- 3 Select the Rule tab and establish a rule for the newly created application:
  - a Click the box next to “Execute Agent based on a “Rule” to select it.
  - b Click the Add icon to find the newly created Application.
  - c Click the Find icon once the Rule window launched.
  - d Click Search to find the SMS Test Application that you created in “Creating an alert for SMS delivery” on page 233.
  - e Select the application from the list to add this newly created Application, and click Add.
  - f In “Execute agent when,” select “field2” and “is greater than” from the drop-down lists, and enter a value in the text field that is slightly higher than the current stock quote.

- g Click the plus sign ( + ) to apply this rule.
      - h Click OK to finish.
- 4 Select the Action tab and set up a destination:
  - a Click Add to add the destination, and select Notify | SMS. The Action Details window displays.
  - b In SMS To, enter your telephone number.
  - c Select your carrier from the drop-down list.
  - d In Subject, enter a subject for this message.
  - e Select “Include Rule’s Content.” Additional options display.
  - f Click “Send the Portlet that met the Rule Criteria.”
  - g Click OK to finish and close the Action Details window.
- 5 In Agent Builder, click Save to save the agent. Supply a name for the agent, such as `GoingUp`, assign roles to all, and click OK.
- 6 In Agent Manager | Stopped, select the newly created agent and click Start to run the agent.

## Creating an e-mail alert using Mobile Web Studio and Portal Interface

For this example, create a Web element in Application Builder, and then create an alert for the Web element. You can enable alerts through the Application Builder Save window, or on the Application properties window in the Web Studio. Selecting the Alert option enables alerts for the application. Tasks include:

- “Creating an alert for e-mail delivery” on page 236
- “Creating a rule from Portal Interface” on page 237
- “Creating an alert schedule” on page 239
- “Viewing an alert created by a Portal Interface user” on page 240
- “Creating rules and schedules from Mobile Web Studio” on page 235

### ❖ Creating an alert for e-mail delivery

This section shows how to create a stock quote alert for e-mail delivery.

- 1 In Mobile Web Studio, select Applications from the Build menu in the left pane, select New from the Application Manager Status menu, and click the New button to launch Application Builder.
- 2 Click the arrow to the right of Add, and select Web Element. You see the New Web Element window.
- 3 Enter a URL to capture an application that contains tabular data, for example, `http://finance.yahoo.com`, and click Go.
- 4 Locate the content you want to capture, and from the Format drop-down list, select Grid to display the data in grid format.  
Click Next.
- 5 Select the presentation option, and click Next.
- 6 In the Define window, define the record layout and specify a record header. See “Defining record layout” on page 133. Click Next.
- 7 In the Filter window, specify filter rules to select only the relevant data from the element. See “Adding filter rules” on page 133. Click Next.
- 8 The Configure Parameters window displays the element’s CGI parameters. See “CGI parameters” on page 89 and “Using the database table adapter” on page 202. Click Next.
- 9 In the New Web Element window, enter a name for the application in the Element Name input field. Click Next.
- 10 Click Finish. The Application Builder window displays with the new application name.
- 11 Click Save, and configure the application:
- 12 Click Close to close Application Builder. The new Web element displays in the right pane.
- 13 Deploy the application to the Portal Interface.

❖ **Creating a rule from Portal Interface**

This section shows how to create a rule for the alert. This rule triggers an alert whenever the Last Trade value for Microsoft’s stock exceeds a set threshold. Use Portal Interface to add a rule to the Web element you created in “Creating an e-mail alert using Mobile Web Studio and Portal Interface” on page 236.

- 1 Log in to the Portal Interface.
- 2 Create a page group, a page, and add the “Stocks” application.

- 3 Click the Alert icon (!) on the application's title bar. The Alert definition window displays.

---

**Note** There are no default rules defined for an alert.

---

- 4 In the Alert definition window, define two rules for the Web element:
  - a In Name, select Playback from the drop-down list.
  - b Leave Rule Data and Description blank.
  - c Under Rules, define the rule relationship by clicking "and" to indicate multiple conditions.

You can also click "or" to indicate a choice between conditions.
  - d Define the condition for Select fields by clicking "label" to indicate field labels are used to meet the condition.

Other condition options include "position" to indicate window position, or "xpath" to indicate the same condition from every record.
  - e For the first rule, use "Execute alert when" to define the condition that triggers the alert:

Select "field 1" from the drop-down list.

Select "contains" from the drop-down list.

Enter MSFT (for Microsoft) in the value field, to establish a match for field 1.

Click the plus sign ( + ) to add the "Execute agent when" rule. The new rule displays under Rules and Rules Summary.
  - f For the second rule:

Select "field 3" from the drop-down list. Notice some of the other choices for establishing conditions that are available in the list.

Select "is greater than" from the drop-down list.

Enter 26 (to indicate a Last Trade stock price) in the value field, to establish a match for field 3.

Click the plus sign ( + ) to add the "Execute agent when" rule. The new rule displays under Rules and Rules Summary.
- 5 Click Done to save the Rule, and click OK in the confirmation window.
- 6 Click Cancel to close the Alert window.

## Creating the schedule (Portal Interface)

This section shows how to create a schedule for the alert. You will use Portal Interface to add the schedule for the rules you set up in “Creating a rule from Portal Interface” on page 237. Alerts do not start until you set up and save a schedule, and start the alert.

Every alert needs a destination to deliver the content. The destination can be:

- An e-mail address. The user’s registered e-mail address is the default destination. You can specify another e-mail address.
- An SMS subscriber. Default SMS carriers include ATT Wireless, Cingular, Nextel, Sprint, T-Mobile, and Verizon.

You can add additional SMS carriers to the *oem.xml*, using the syntax shown in this example:

```
<DeviceList>
  <DeviceClass name="sms" type="email">
    <DeviceDef name="ATT Wireless"
      address_value="mobile.att.net" />
    <DeviceDef name="Nextel"
      address_value="messaging.nextel.com" />
    <DeviceDef name="Sprint"
      address_value="messaging.sprintpcs.com" />
  </DeviceClass>
</DeviceList>
```

### ❖ Creating an alert schedule

- 1 From Portal Interface, click the My Info link. The icon looks like a lowercase “i.”
- 2 In MyInfo, select the Alert tab.
- 3 Set up the schedule parameters and the alert destination:
  - a Click the box next to Trigger Alert Based on a Schedule to select it.
  - b For Time, use the option controls to specify the start time. For the tutorial, use the current time.
  - c For Recurrence, use the option controls to select the recurrence unit (such as seconds, minutes, hours) and interval. For the tutorial, select Minutes as the unit, and 5 as the interval. This indicates that the alert schedule recurs every five minutes.
  - d Click Disable once to indicate that the e-mail or SMS notification should be sent only once, the first time the conditions are met.

- e For Range, enter the current date in MM/DD/YYYY format as the start date. For convenience, use the calendar icon to select the month and date.  
  
For the end date, click “No end date.” Alternatively, you could click “End by” and specify an end date.
  - f For Send Alerts to, click Email and enter your e-mail address. Alerts are sent to this e-mail address. You can also click SMS to send the alert to an SMS subscriber, as described in “Creating an SMS alert using Mobile Web Studio” on page 232.
- 4 Click Done to save the schedule.
  - 5 Click Start. The Start and Stop buttons act as a toggle, starting and stopping the schedule, and therefore, the alert.
  - 6 Click Cancel to close the Alert window.

❖ **Viewing an alert created by a Portal Interface user**

This section shows how to view the rules and schedule for the alert you created, on the Portal Interface, from the Mobile Web Studio.

- 1 Log in to Mobile Web Studio.

---

**Note** Use your portal login name.

---

- 2 Select Agents in the left pane.
- 3 Select Ready under Agent Manager. You see the alert you modified in Portal Interface. The alert has the same name as the Portal Interface account name.
- 4 Double-click the alert to display the Agent Builder window.
- 5 Select the Schedule tab to see the schedule for the newly created alert.
- 6 Select the Rule tab to see the rule list for the newly created alert.

---

**Note** You can disable a rule by clicking the box to the left of the Rule ID.

---

- 7 From the Rule tab, select a rule and click Edit to see the details of the rule. You can modify the rule from Mobile Web Studio, assuming you have correct permissions, even if it was created on Portal Interface. Click OK to close the edit window.
- 8 Click Close to close Agent Builder.

**❖ Viewing the alert in e-mail**

This section shows how to view the generated alert in your e-mail application. Wait to perform this step until after the first interval that you set in “Creating an alert schedule” on page 239.

- 1 Log in to your e-mail account, using the account that you specified in “Creating an alert schedule” on page 239.
- 2 View your inbox for the alert notice. Use the Date and Time, or Subject columns to identify the alert.
- 3 Double-click the alert to see details.

---

**Note** You may want to stop the alert to keep your e-mail account from filling up. To do so, use the Stop button.

---

## Using agents

The Mobile Web Studio left pane lists three agent-related components from which to choose:

- Agents – create and edit agents.
- Adapters – write an agent’s data to an adapter.
- Servers – monitor the transactions that have been processed and completed via transaction logging for the agent running on the agent server.

The list in the detail pane displays the agent’s name, what the agent does, the owner, the last time the agent ran, the result of the agent’s last execution, and the next time the agent is scheduled to run.

The Details column codes indicate that the agent:

- S – has a schedule.
- R – contains a rule.
- D – contains a delivery action.
- N – contains a notify action.
- T – contains a trigger action.
- E – uses error management.

## Changing an agent's status and running an agent once

Agents can exist in any of three states:

- Ready – the running state for a scheduled agent. When you create a schedule for an agent and change that agent's status to Ready, the agent automatically runs based on the schedule you specified.
- Stopped – the state for agents that do not run on a schedule. New agents are always created in a Stopped state. Select Run to run an agent with a Stopped status once.
- Deleted – agents that have been deleted. Until you dump the database, the agent record still exists in the portal database table.

### ❖ Modifying agent status and executing an agent once

- 1 Select the agent's current state from the Agent Manager Status menu.
- 2 Select the agent in the detail pane.
- 3 From the Agent Manager toolbar, select:
  - Run – to run any agent with any status once.
  - Start – to change an agent's status from Stopped or Deleted to Ready.
  - Stop – to change an agent's status from Ready to Stopped.

---

**Note** You can also right-click on an agent, select Status, and choose the status from the pop-up that appears.

---

To run an agent repeatedly, create a schedule for the agent and change the agent's status to Ready. See “Schedule tab” on page 243 for details.

### ❖ Creating new agents

When you create an agent, you create a schedule, create rules, and specify actions and error management.

- 1 Select Automate | Agents from the Mobile Web Studio left pane.
- 2 Select New above the detail pane. You see the Agent Builder window.

The Agent Builder has four tabs. For details, see:

- “Schedule tab” on page 243
- “Rules tab” on page 243
- “Actions tab” on page 249

- “Error Management tab” on page 253

## Schedule tab

The Schedules tab allows you to automate an agent’s recurring execution based on intervals that you specify. For example, you can create a schedule that runs an agent’s action every hour, once a day, once a week, or once a month.

### ❖ Setting up an agent to run on a schedule

- 1 Log in to Mobile Web Studio, select Automate | Agents from the left pane, then click New. The Agent Builder appears with the Schedule tab displayed.
- 2 Select “Trigger Agent Based on a Schedule.” When you select this option, the remainder of the schedule options display.
- 3 Enter the following information for the schedule options:
  - Time – the time of day (AM or PM) you want the agent to run.
  - Recurrence – the frequency at which you want the agent to run, entered in minutes, hours, days, weeks, or months.
  - Range – the date you want the agent to begin and end running, in month/day/year format. You can also click the small calendar to the right of the Start Date and End By to select the date. If you want the agent to run indefinitely, select No End Date.

## Rules tab

The Rules tab lets you specify rules that the application’s content must meet before the action associated with the agent can be executed. For example, you may want an agent to notify certain people of a stock price only when that price reaches a specified amount.

### ❖ Setting up agent rules

- 1 Log in to Mobile Web Studio, select Automate | Agents from the left pane, then click New.
- 2 When the Agent Builder appears, select the Rules tab.

- 3 Select Execute Agent Based on a Rule and click Add. The Rule Editor window displays where you select the source of the rule's criteria and create the actual rules.
- 4 Base Rule On – select:
  - Application – to base the rule on a selected application's content. See the next procedure, "Basing rules on an application's content."
  - Criteria – to base the rule on criteria that you specify. See the procedure, "Basing rules on criteria."

❖ **Basing rules on an application's content**

- 1 In the Rule Editor, select Application and click Find to select an application using the search dialog box. When you select the application and click Add, the application title appears in the Rule Editor's Name drop-down list and the application's content displays beneath Preview Source.
- 2 Select Playback to display the application as it previews. Select XML to display the application as XML.

The Rules section allows you to group rules together based on the Boolean operators "AND" or "OR." See "Using rule operators" on page 246 for a description of each operator.

Additional options let you select how to construct the rules based on:

- Label – uses column labels in the data. When table fields reorder, the rule still picks up the correct field.
- Position – an index on positions in the data.
- XPath – if you choose to view the content as XML, this option allows all matches to be done via XPath expressions.

---

**Note** XPath is a set of syntax rules for defining parts of an XML document.

---

When the application displays as XML, you see this code:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <ContentDoc>
  - <Element name="bonds">
    - <Record>
      <Field op_label="Maturity" op_position="1">Maturity</Field>
      <Field op_label="Yield" op_position="2">Yield</Field>
      <Field op_label="Yesterday" op_position="3">Yesterday</Field>
```

```

    <Field op_label="Last Week" op_position="4">Last<BR>Week</Field>
    <Field op_label="Last Month" op_position="5">Last<BR>Month
      </Field>
  </Record>
- <Record>
  <Field op_label="Maturity" op_position="1">3 Month</Field>
  <Field op_label="Yield" op_position="2">0.99</Field>
  <Field op_label="Yester- day" op_position="3">1.00</Field>
  <Field op_label="Last Week" op_position="4">0.97</Field>
  <Field op_label="Last Month" op_position="5">1.04</Field>
</Record>
...More XML records follow...

```

“Label” matches the XML `op_label` attribute and “Position” matches the XML `op_position` attribute.

XPath uses path expressions to locate nodes within XML documents. XML documents can be represented as a tree view of nodes.

Any XPath expression that uses partial paths starts at each “record” node and is applied. Unstructured data is in one record, so the pop-up selections for Label and Position contain only “content.”

---

**Note** XPath and the comparable options are valuable rule options. You can use the comparable options of “found” and “not found” to check if a specific node is found in an XML document regardless of the text content. This is useful to see if an application is broken, or has been changed in a way that requires attention.

To learn more about XPath, see the OASIS Specification at <http://www.w3.org/TR/xpath>.

Additional examples to help you understand how to use XPath are available on the W3 Schools Web site at <http://www.w3schools.com/xpath/default.asp> and on the ZVON Web site at <http://www.zvon.org/xxl/XPathTutorial/General/examples.html>.

---

- 3 At the bottom of the Rules panel, in the three unlabeled selection boxes after the term “Execute agent when,” select:
  - 1st box – Maturity
  - 2nd box – equals (text)
  - 3rd box – 3 months
- 4 Click the plus sign next to the rule to save the rule. Once you save the rule, the entry appears in the Rules panel.

- 5 To edit a rule, click the check mark. To delete a rule, click the X.  
When you edit an entry, the check mark changes to plus sign. Click the plus sign to save your changes.
- 6 When you finish, click OK to exit the Rule Editor and return to the Agent Builder. Click OK to save your entries.

## Using rule operators

The Rules section of the Rule Editor lets you create and group rules together based on the Boolean operators AND or OR that tell the system which keywords the application's content must include or exclude before the application's agent executes.

- AND or OR– use these operators to specify what terms must appear in the selected application's content. When you click the plus sign to save the rule, then click OK, the rule is summarized on the Rule tab. For example,

- If you set the rule as:

```
The rule will execute this agent when 'Maturity' equals (text)
'3 Month' AND label 'Yield' is greater than '0.5'
```

Then the bond application meets the rule, and the actions can be executed.

- If you set the rule as:

```
The rule will execute this agent when 'Maturity' equals (text)
'3 Month' AND label 'Yield' is greater than '1'
```

Then this bond application does not meet the rule, and the actions will not be executed.

- If you set the rule as:

```
The rule will execute this agent when 'Maturity' equals (text)
'3 Month AND 6 Month'
```

Then this bond application does not meet the rule, and the actions will not be executed.

- If you set the rule as:

```
The rule will execute this agent when 'Maturity' equals (text)
'3 Month' OR 'Maturity' equals (text)'6 Month'
```

Then this bond application meets the rule, and the actions can be executed.

The relational operators that appear in the 2nd rule box are:

- equals (number) – converts the application data into floating point numbers for the equals comparison. If the data is not numeric, the comparison fails.
- is greater than – converts the application data into floating point numbers for the greater than comparison. If the data is not numeric, the expression returns false. The application data must be greater than the value supplied in the rule’s text field.
- is less than – converts the application data into floating point numbers for the less than comparison. If the data is not numeric, the expression returns false. The application data must be less than the value supplied in the rule’s text field.
- equals (text) – compares the application data to the text supplied in the rule’s text field. If they are equal, the expression returns true.
- contains – returns true if the application data contains the text string supplied in the rule’s text field.
- starts with – returns true if the application data starts with the text string supplied in the rule’s text field.
- ends with – returns true if the application data ends with the text string supplied in the rule’s text field.
- matches RegExp – returns true if the application data matches the regular expression supplied in the rule’s text field. To understand the format of regular expressions, see the documentation on the Apache Web site at <http://jakarta.apache.org/regexp/apidocs/org/apache/regexp/RE.html>.
- contains data – checks the application to see if any data has a size greater than zero. The rule’s text field is disabled for this option.
- contains no data – is the opposite of “contains data.” Checks the application to see if any data has a size equal to zero. The rule’s text field is disabled for this option.
- is found – used with XPath. Returns true if the XPath expression resolves to a XML node or attribute.
- is not found – used with XPath. This operator is the opposite of “is found.” Returns true if the XPath expression does not resolve to a XML node or attribute.

**❖ Basing rules on criteria**

Basing a rule on criteria allows you to execute a rule on a selection of applications that meet user-specified parameters; for example, to execute the rule on applications owned only by a particular user.

This procedure describes how to set up an agent to find applications in which one-click content capture has failed and the applications are broken.

- 1 In the Rule Editor, select Criteria and click Select. You see the Add Content by Criteria window.
- 2 To complete the Criteria Parameters, select:
  - Category – find applications only in the selected category.
  - Subcategory – find applications that are only in the selected subcategory.
  - Status – find applications that only have the selected status.
  - Catalog – find applications that are only in the catalog you select.
  - Page – find applications that are only on the page you select.
  - Owner – find applications that only belong to the user whose name you enter.
  - Modified By – find applications that only belong to the user whose name you enter.
  - Roles – find applications that are assigned only those roles you select in the Available Roles list and move to the Assigned Roles list.
- 3 Click OK to save the criteria and return to the Rule Editor, which populates the Name drop-down list for Preview Source with all items that meet the selected criteria at that moment.
- 4 In the Rules section, select Xpath and enter this execution statement:

```
'//ContentDoc/Element/text( )' equals (text)
'ERROR'
```

and click OK.

The rule has been set up to check for a particular type of error by looking at the XML returned from the application when it fails. The XML for this application displays as:

```
<?xml version="1.0" encoding="UTF-8 ?>
-<ContentDoc>
<Element name="bonds2">ERROR</Element>
```

&lt;/ContentDoc&gt;

## Rule processing

The criteria you select allows a list of applications to be processed by the rule you create. The criteria is processed when the rule is executed.

Criteria-based rule processing uses these directives:

- The criteria you select are processed as “AND” options.
- When you select a catalog or a page, the portal processes all applications in that catalog or page by the rule.
- Selecting options other than catalog or page indicates that only the applications that match all the selected criteria are processed. For instance, only applications that have an Approved status are processed.
- The role selection takes advantage of the *global.properties.xml* property “RoleBaseDisplaySeeAllRoles.” In the example, this property is set to “superuser,” which means the example criteria returns all approved applications created by everyone.

## Actions tab

The Actions tab lets you set up actions for an agent to execute.

### ❖ Creating agent actions

- 1 Log in to Mobile Web Studio, select Automate | Agents from the left pane, then click New. The Agent Builder appears.
- 2 Select the Actions tab.
- 3 Click Add and select:
  - Deliver – to write an application’s content to an adapter that writes the content to a file system, an e-mail message, or a database table.
  - Notify – to send an e-mail message that contains the rules content or the actual rules to the users.
  - Trigger – to select another agent to execute when this agent runs.

### ❖ Creating a delivery action

The Deliver action lets you select applications that have content you want written to an adapter.

- 1 Select Add | Deliver from the Actions tab. You see the Action Details window.
- 2 Complete these fields:
  - Description – enter a description of this action.
  - Destination Viewer – specifies the format in which to write the content. Currently, only HTML is supported.
  - Destination Set – click Add to select an adapter into which an application’s content is written. See “Using adapters” on page 255 for details on defining adapters for agent use.

The content of each application in the Application Set is written to each defined adapter that you select for the Destination Set. To select multiple adapters, hold down the Ctrl key as you select each adapter, then click Add.

After you add an adapter to the Destination Set, select the adapter and click Properties to display a description of the adapter.

- Application Set – click Add to select the applications from which content is written to an adapter. There are four options:
  - Application – select an individual application.
  - Criteria – select an application based on criteria.
  - Rule Application – pass the triggered rule application to this action. This option is enabled only if a rule is set in the agent.
  - Matching Rule Record – pass the matching record from the rule as XML content to the destination set. This option is enabled only if a rule is set for the agent.

When you select Criteria, you see the Add Content by Criteria window.

- 3 Click OK to save the delivery action.

❖ **Creating a notify action**

This option lets you enter an e-mail address to send application content and an optional message without creating an e-mail adapter.

- 1 Select Add | Notify on the Actions tab.
- 2 Complete these options:
  - E-mail To – enter the complete e-mail address of the recipient.

- Subject – enter the text that should appear as the subject for the e-mail’s message.
- Prepend Subject with Rule’s Application Name – adds the name of the applications that were executed and passed through the Rules section to the beginning of the Subject line.

---

**Note** This option is for each application executed by a rule; that is, if a rule is a criteria, when an application triggered, a message gets sent for each application.

---

If, for example, you are searching for broken applications, you can use this option to be notified of the application that is no longer working.

- Include Message – lets you write a message that displays in the message body. When you select this option, a text window displays where you type in the message.
- Include Rule’s Content – select this option to display additional options that pertain to the executed rule. This option is enabled only if a rule is selected.
  - Send the Entire XML Rule Source – sends the XML content of the rule that was triggered.
  - Send Only the Record(s) From the Rule Source That Meet the Rule Criteria – sends the section of the rule application that was triggered. For example, the e-mail content might look similar to this:

```
Date: Mon, 21 Jul 2003 11:35:39 -0700 (PDT)
From: agentserver@sybase.com
Reply-To: agentserver@sybase.com
Subject: bonds portlet matched
X-Mailer: Sybase-EP
```

```
Rule executed on portlet=bonds(id=271)v0.1
```

```
Included Matching Records:
```

```
<?xml version="1.0" encoding="UTF-8"?>
<MatchingRecords><Match><Record> <Field op_label="Maturity"
op_position="1">3 Month </Field> <Field op_label="Yield"
op_position="2">0.79 </Field> <Field op_label="Yester- day"
op_position="3">0.78 </Field> <Field op_label="Last Week"
op_position="4">0.77 </Field> <Field op_label="Last Month"
op_position="5">0.71 </Field>
```

</Record></Match></MatchingRecords>

- Send the Application That Met the Rule Criteria – sends the rule application that triggered the rule. The application is sent in the default playback format, which was assigned during application creation and set the content-type; specifically, it could be an XML application.
- Send the Application as XML That Met the Rule Criteria – sends the rule application that triggered the rule as the exact XML that matched the criteria.

3 Click OK to save the action.

#### ❖ **Creating an agent action that triggers another agent**

Triggers allow one agent to process data and determine if another agent should be run.

1 Select Add | Trigger on the Actions tab. You see the Add Agent Search dialog box.

2 Complete these options:

- Name – enter the name of the agent you want to find. If you leave this field blank, the portal searches for all agents.
- Details – these optional selections match the codes in the Details column of the agent list in the Agent Manager detail pane. Choose from these options to narrow the search based on what is included in the agent.
  - Schedule – matches agents with an “S” details code.
  - Rules – matches agents with an “R” details code.
  - Content Delivery – matches agents with a “D” details code.
  - Send Notification – matches agents with an “N” details code.
  - Trigger Agent – matches agents with a “T” details code.
  - Error Management – matches agents with an “E” details code.

---

**Note** When you save an agent, the system assigns the agent a code based on the details contained in the agent.

---

- Status – select the status of the agents you are searching for—Ready, Stopped, or Deleted.

- Owner – enter the user name of the person who owns the applications for which you are searching.
- 3 Click Search. The agents that match the selected options display in the Results pane.
  - 4 In the Results pane, select the agent you want to trigger and click Add. You return to the Rule Editor.

## Error Management tab

The Error Management tab allows processing to take place or continue when an error occurs while the agent is running.

Click the plus sign next to each error type—two options display for each error type:

- Set Agent Status To – resets the agent’s state to the selected status when an error occurs. For example, you can use this option to change an agent’s state to Stopped when an error occurs to prevent the error from happening again.
- Send Message to Error E-mail – sends an error message to the e-mail address specified in the Error e-mail field at the top of the Error Management tab.

The Active column displays a check mark if one or both of the options are set for the given error check. The error check types are:

- Any Error Occurred – encompasses all errors when the agent is running.
- System Error Occurred – an unexpected system failure like “Disk Full” or “Database Connection Refused.”
- Content Retrieval Error Occurred – an agent’s rule or delivery action is processing an application and the portal playback engine fails to retrieve the application’s content.
- Rules Error Occurred – a rule uses a regular expression and the rule fails because the content is not in the expected format.
- Actions: Trigger Error Occurred – the agent that should get triggered does not exist.
- Actions: Notify Error Occurred – there is a problem sending an e-mail message to the Notify recipient.

- Actions: Delivery Error Occurred – a problem occurs delivering an application set to the destination set defined in the delivery action.

## Saving a new agent

When you finish creating a new agent’s schedule, actions, rules, and error management, you must save the agent.

- 1 When you save a new agent for the first time, click Save or Save As. You see the Agent Properties window.
- 2 Complete these fields:
  - Name – enter a name for the agent.
  - Description – enter a brief meaningful description of the agent.
  - Agent Server – values are “None” or the name of the assigned agent server.
    - None – the new agent runs on the first agent server that is recognized. Normally this is the default agent server.
    - Assigned agent – clicked Find and select an agent server. The new agent runs only on the selected agent server.

Click Clear to remove the assigned agent server and return the value to None.

  - Roles – select the available roles for who can edit the agent’s properties and view the agent’s log and click Add. Click Add All to assign all available roles.
- 3 Click OK to save your entries.

## Handling “permission denied” error messages

The roles assigned to an agent are independent of the roles assigned to applications that may run inside the agent. This means you can create an agent that includes applications that require role “A,” but the agent is assigned role “B.” If another user of role “B” edits this agent and tries to preview the application of role “A,” they receive a permission denied error message.

Permission denied errors can occur in these areas:

- Application preview in the Actions Delivery panel displays “Permission Denied” when the user is not in the same role as the application being previewed.
- Application preview in the Rules panel displays “Permission Denied” if the user does not have the rights to view the application.
- Role-based views of the rules criteria shows “Permission Denied” when selecting the application to which the user does not have access rights.

---

**Warning!** When you preview an application and receive the error “Permission Denied,” the system disables the JavaScript Rules panel that is directly linked to the parsing of the previewed application.

---

## Viewing or changing agent properties

To view an agent’s properties, select the agent in the Agent Manager detail pane, then click Properties in the Agent Builder. You see the Agent Properties window.

You can change the description, select an agent server, and change the agent’s assigned roles.

When you finish viewing or changing the agent’s properties, click OK. Click Cancel to exit the window without saving any changes you may have made.

## Using adapters

Adapters allow you to write an application’s content to a file, an e-mail message, or a database table. When you assign an adapter to an agent and the agent is triggered, the adapter writes the application’s content to the specified destination. Three default adapter types are installed with Unwired Accelerator—fs (file system), email, and db (database).

### ❖ Creating new adapters

- 1 Select Automate | Adapters from the Mobile Web Studio menu.
- 2 Select New from the toolbar, then select the type of adapter you want to create:

- File system (fs) – writes application content associated with an agent to a file system.
- E-mail – writes application content associated with an agent to an e-mail message.
- Database (db) – writes application content associated with an agent to a database table.

What you see next depends on the type of adapter you chose. For details, see:

- “Creating a file-system adapter” on page 256
- “Creating an e-mail adapter” on page 257
- “Creating a database adapter” on page 258

## Creating a file-system adapter

- 1 Click New on the Adapter Manager toolbar and select “fs.” You see the Adapter Builder.
- 2 Complete these fields:
  - Name – the program fills in this field after you save the new adapter and enter the name.
  - ID – the adapter ID is a sequential number maintained by the system.
  - Description – the system automatically enters “FileSystem Adapter” in this field, which you can change if you want.
  - Type – displays the adapter type you chose after you clicked New.
  - Viewer – specifies the format in which to write the content. Currently only HTML is supported.
  - Roles – select the available roles that can edit the adapter’s properties and view the adapter’s log and click Add. Click Add All to assign all available roles.
  - Parameters – for a file-system adapter, the parameter is the directory where you want the application’s content written.
    - Name – the parameter name, which is “directory.” You cannot edit this field.

- Display Name – change the display name or accept the default of “Storage Directory.”
  - Directory – enter the location where you want the content written. For example, *d:\temp\content\*. If the folder does not exist, the adapter creates the folder when it writes the application content.
- 3 Click Save As.
  - 4 In the Save As dialog box, enter an adapter name and click OK.
  - 5 Click Close to exit the Adapter Builder.

## Creating an e-mail adapter

- 1 Click New on the Adapter Manager toolbar and select “email.” You see the Adapter Builder.
- 2 Complete these options:
  - Name – the program fills in this field after you save the new adapter and enter the name.
  - ID – the adapter ID is a sequential number maintained by the system.
  - Description – the system automatically enters “Mail Adapter” in this field, which you can change if you want.
  - Type – displays the adapter type you chose after you clicked New.
  - Viewer – specifies the format in which to write the content. Currently, only HTML is supported.
  - Roles – select the available roles that can edit the adapter’s parameters and view the adapter log, and click Add. Click Add All to assign all available roles.
  - Parameters – for an e-mail adapter, the parameters are the values specific to writing the application’s content in an e-mail message; for example, the e-mail address, the subject line, and so on. You can change the display name or accept the default.
    - include\_portlet – enter “true” or “false.” Set to true to write the application contents in the e-mail message. Set to false to write something in the msg\_body field instead of including the application’s contents.

- `from_addr` – enter the sender’s e-mail address.
- `subject` – enter the text to appear as the message’s subject line.
- `msg_body` – enter the text you want added to the body of the message. Normally, if you set `include_application` to true, you would not enter anything for this parameter.

If you set the first parameter to true and include a message, the text you enter displays before the application’s content and the entire e-mail message is sent to a MIME type of “text/plain.”

If you send only application content, the content type is set to a MIME type of “html.”

- `to_addr` – enter the e-mail address of the recipient. To send the message to multiple recipients, enter the e-mail addresses as a comma-separated list.

- 3 Click Save As.
- 4 In the Save As dialog box, enter an adapter name and click OK.
- 5 Click Close to exit the Adapter Builder.

## Creating a database adapter

- 1 Click New on the Adapter Manager toolbar and select “db.” You see the Adapter Builder.
- 2 Complete these options:
  - `Name` – the program fills in this field after you save the new adapter and enter the adapter’s name.
  - `ID` – the adapter ID is a sequential number maintained by the system.
  - `Description` – the system automatically enters “Database Adapter” in this field, which you can change if you want.
  - `Type` – displays the adapter type you chose after you clicked New.
  - `Viewer` – specifies the format in which to write the content. Currently, only HTML is supported.
  - `Roles` – select the assigned roles associated with who can edit the adapter’s parameters and view the adapter log and click Add. Click Add All to assign all available roles.

- Parameters – the parameters for a database adapter are the database values that specific to writing the application’s content to a database. You can change the display name or accept the default.
    - Key\_column\_name – enter the table field into which the application ID is written.
    - Table\_name – enter the name of the database table. This table must already exist.
    - Value\_column\_name – enter the table field into which the application’s content is written.
    - User – enter the database user for the database connection.
    - Url – enter the JDBC connection URL used for the database connection.
    - Password – enter the password for the user connection to the database.
    - Driver – enter the database driver used for the JDBC connection.
- 3 Click Save As.
  - 4 In the Save As dialog box, enter an adapter name and click OK.
  - 5 Click Close to exit the Adapter Builder.

## Editing an adapter or viewing the log

To edit an adapter, select the adapter in the detail view and click Edit, or right-click the adapter and select Edit.

To view the adapter’s activity, select the adapter in the detail view and click View Log, or right-click the adapter and select View Log. See “Viewing transaction logs” on page 261 for more details.

## Using agent servers

Agents run on agent servers. When you install and start Mobile Web Studio, the default agent server is created by the portal. All agents run on the default agent server unless you assign them to a different agent server.

Every onepage deployed WAR file (the portal) has an agent server thread running when it starts up. If you set up a cluster of onepage Web applications and want an agent to run on only one of the server machines (for example, maybe the database connection or file system only exist on one machine), then you can assign that server an ID in the *global.properties.xml* portal configuration file. When you restart the server and Mobile Web Studio, the portal creates the agent server reference, which a user can then select for a specific agent.

You can also create the agent server reference using the following procedure. When you want the agent server to run, you must take the assigned ID and set it in *global.properties.xml* on the machine that would host the agent server; for example:

```
<Property name="AgentServerID" value="211"
description="Default Agent Server ID" menugroup="10"/>
```

#### ❖ **Creating new agent servers**

- 1 Select Automate | Servers.
- 2 Select the New icon from the toolbar. You see the Agent Server Editor.
- 3 Complete these fields:
  - Name – the program fills in this field after you save the new agent server and enter the name.
  - ID – the agent server ID is a sequential number maintained by the system. The default agent server has an ID of 1.
  - Description – enter a description of the agent server.
  - Status – select the agent server’s state —Ready, Stopped, Broken, or Deleted. Setting a server to a state other than Ready stops that agent server from retrieving agents to run. If an agent is assigned to an agent server with a state other than Ready, the agent does not run until the agent server state is reset to Ready.
  - Roles – select the available roles that can edit the agent server’s properties and view the agent server’s log and click Add. Click Add All to assign all available roles.
- 4 Click Save As to save the new agent server. A dialog box prompts you for the agent server’s name.
- 5 Enter a name for the agent server and click OK.
- 6 Click Close to exit the Agent Server Editor.

## Editing an agent server

To edit an agent server, select the server in the detail pane and click Edit, or right-click the agent server in the detail pane and select Edit.

## Viewing transaction logs

To view an agent server's activity:

- 1 From the Mobile Web Studio left pane, select Automate | Servers.
- 2 From the Status menu, select the state in which the agent server resides.
- 3 In the detail pane, select the server and click View Log, or right-click the agent server name and select View Log.

The log window displays information about what agents have been started and the actions that occurred during the agent's execution. The server log records all agent and adapter activity on all agent servers. Each agent and adapter has a log also and displays only the information pertinent to its type.

- Click Refresh to update the display.
- Click Clear Log to clear the content from the database. Clearing messages at the agent server level clears the messages from all agents and adapters that were running on that agent server. However, clearing messages from an agent or adapter log window clears only their references to the messages. The messages are still referenced in the agent server's log and must be cleared from the agent servers log for permanent deletion.

## Limiting the log size

To prevent sending too much data to the browser and slowing down performance, define the number of message you want to display in *global.properties.xml*. For example:

```
<Property name="MaxLogRows" value="150"
description="The maximum number of log rows to return
on a view log request (set to zero for all)."
```

To view all messages in a sizable log, use SQL to display the log rather than View Log as the text returned over HTTP can be too enormous for the browser.

Logging is kept in the `app_logs` table. The objects that reference each table row are stored in the `app_log_object_refs` table. For example,

```
select * from app_logs
```

displays all log messages.

---

**Note** The agents transaction logging can fill a database quite fast. If the View Log or other agent functionality seems slow, check the database logs to see if the database is out of space. You may see an error like:

```
Tasks are sleeping waiting for space to become available  
in the log segment for database tempdb
```

To solve this problem, increase the database size. See the *ASA Database Administration Guide* on the *Technical Library* CD that comes with the product. If you are using a database other than Adaptive Server Anywhere with UA, refer to that product's documentation for instructions on increasing the database size.

---

# Building Catalogs

Catalogs allow you to organize existing applications in groups. This chapter describes how to work with Mobile Web Studio and Portal Interface catalogs.

Topic	Page
Overview	263
Creating and editing catalogs	266
Updating Portal Interface with changes	270

## Overview

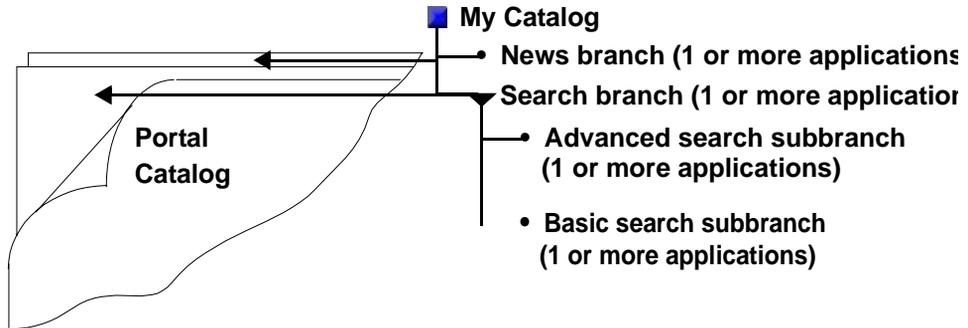
You create catalogs in Mobile Web Studio. A catalog contains applications grouped in user-specified branches. You can arrange applications into different branches and subbranches within a catalog for sorting and organization.

---

**Note** Catalogs are for use with Portal Interface and do not display on mobile devices.

---

Although you can create an unlimited number of catalogs, you can display only one catalog per resource in Portal Interface. The catalog that is downloaded to Portal Interface—the display catalog—is the only catalog in Mobile Web Studio with an approved status and marked as active.

**Figure 8-1: Catalog branch and subbranches illustration**

When portal users select Add Applications in Portal Interface, they can browse through the catalog and add applications of their choosing to their personal pages and page groups. Users cannot add applications to predefined or default pages or page groups.

A catalog can contain applications associated with any role. In Portal Interface, a user sees only the applications that have access roles assigned to the user.

## Catalog Manager

Use the Mobile Web Studio Catalog Manager to create new catalogs and perform management functions for existing catalogs. To access the Catalog Manager, select Build | Catalogs from the Mobile Web Studio left pane.

The Catalog Manager interface includes:

- Status menu – select a status to see catalogs with that status displayed in the Catalog Manager detail view.
- Toolbar – displays the New and Edit buttons that let you add, view, and edit catalogs. If you change the display catalog (that is, mark the current approved display catalog as inactive and mark another approved catalog as active), Update lets you update the portal with a new display catalog, which can be the only approved catalog that is marked active. In the detail view, you can also filter whether to display only your catalogs of a selected status, all catalogs of a selected status, or only the active catalog.
- Status bar – shows the current user and resource.
- Detail view – displays catalogs based on your selections from the Status menu, and the Filter By and Show Active Only options.

- Right-click Catalog Menu – right-click on the detail view to display the Catalog Menu. The options let you add and edit catalogs, view catalog properties, mark an approved catalog as active, and change a catalog’s approval status.

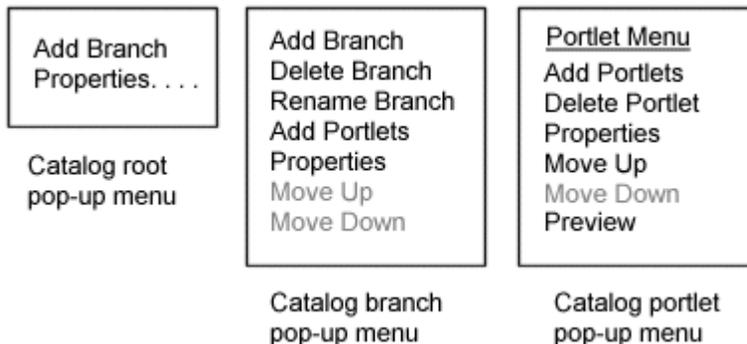
## Catalog Builder

To launch Catalog Builder, click New on the Catalog Manager toolbar or right-click in the detail view and select New from the pop-up.

The Catalog Builder interface includes:

- Catalog tree – displays the catalog branch and subbranch order. You can add, delete, rename, and move branches, subbranches, and applications, which display in the application detail view.
- Toolbar – save an existing catalog with the same name and assigned J2EE roles, or use Save As to save a new or existing catalog with a new name and new J2EE roles. You can also use the options on the toolbar to add branches and applications, edit or view application properties, and preview applications in a separate browser window.
- Application detail view – displays the applications of the branch selected in the catalog tree.
- Pop-up menus – right-click the top of the catalog tree, a branch in the tree, or on an application in the detail view to access the functionality of each menu.

**Figure 8-2: Catalog Builder pop-up menus**



## Creating and editing catalogs

---

**Note** When you add, modify, or delete portal objects—catalogs, pages, and pages groups—you must update Portal Interface with those changes. See “Updating Portal Interface with changes” on page 270.

---

To create a new catalog, do one of the following:

- Right-click in the Catalog Manager detail view and select New from the Catalog Menu. The Catalog Builder window opens.
- Click New on the Catalog Manager toolbar. The Catalog Builder launches.

The catalog tree shows only an untitled catalog. The application detail view is empty because no applications or branches have been created for the new catalog.

Use a branch to organize applications within a catalog. As you create new branches, keep these rules in mind:

- You must create a branch for a new catalog before you can add applications.
- A branch can contain other branches or applications.
- You can have both applications and branches beneath a single branch.

- You can create a new branch beneath the top level of a catalog, or you can create a branch beneath other branches.
- You cannot create branches beneath applications.

❖ **Adding a new branch**

- 1 In the Catalog Builder, click the catalog top level to add a branch, or an existing branch to add a subbranch. If this is a new catalog with no branches, there is only one level to select—Untitled Catalog.

---

**Note** When the Catalog Builder is launched for a new catalog, the catalog tree shows only an untitled catalog. The application detail view is empty because no applications or branches have been created off the new catalog.

---

- 2 Click the down arrow next to the Add icon and select Add Branch, or right-click the catalog's top level and select Add Branch from the pop-up menu.
- 3 Enter a name in the text field that appears. The default name is New Branch.
- 4 Click OK.

❖ **Adding an application to a catalog**

Applications can only be added to branches of a catalog. Only applications that are approved and marked as active can be added to a catalog.

The application list displays the applications that are assigned to a branch. You can add, delete, or move applications in the application list. The columns displayed in the Application List are Name, Version, Owner, and Date Modified.

- 1 In the catalog tree, select an existing branch to which you want to add an application.
- 2 Click Add on the Catalog Builder toolbar to launch the Find Application window. You can also:
  - Right-click the branch and select Add Applications from the pop-up menu; or
  - Select the branch, click the down arrow next to the Add button, and select Add Applications; or
  - Select the branch, right-click in the application detail view, and select Add Applications from the Application Menu.

- 3 Fill in the criteria by which you want to search for existing applications, then click Search. Applications that meet the search criteria display in the Search Results pane.

Select Advanced Search to display additional search criteria:

- Modified By – search for applications modified by a particular user.
  - Status – because only Approved applications can be added to a catalog, you cannot change this option.
  - Title URL – search for applications with a specific title URL.
- 4 In the Search Results pane, click a listing to select an application. To select several applications, hold down the Ctrl key and click each listing you want to select.
  - 5 Click Add to move the selected applications to the application list in the detail view of the Catalog Builder.

❖ **Saving a new catalog**

- 1 Once you are satisfied with a new catalog's contents, click the down arrow to the right of the Save button and select Save As.
- 2 Enter a unique name for the catalog.
- 3 Select Versioning to save the catalog as a new version.
- 4 Enter an optional description and notes.
- 5 Click a role in the Available Roles list to select that role, then click Add to add the role to the Assigned Roles list. Click Add All to move all of the available roles to the Assigned Roles list.

---

**Note** Only the users assigned the roles in the Assigned Roles list can access this catalog.

To allow every portal user to access the Add Application functionality, add the PortalUser role to the Assigned Roles list. Every portal user is guaranteed to have the PortalUser role.

See the *Enterprise Security Administration Guide* for information on setting up Enterprise Security to work with the portal.

---

- 6 When you are satisfied with your entries, click OK.
- 7 When the confirmation prompt confirms that the catalog was saved successfully, click OK.

- 8 Return to the Catalog Manager and select New from the Catalog Manager Status menu. You see the new catalog in the detail view. When you create a catalog in Mobile Web Studio, the catalog is automatically saved with a status of New.

❖ **Viewing or changing a catalog's properties**

- 1 To view or edit a catalog's properties, right-click a catalog in the Catalog Manager detail view and select Properties from the pop-up.

Change any of the following options:

- Status – to display a catalog in Portal Interface, change the status to approved.
- Shared – this option is reserved for a future release.
- Active – make this the active catalog that displays in Portal Interface. Only one catalog can be the display catalog and it must have an approved status and be marked active.
- Description/Notes – enter a note or description.

- 2 Click OK to save your changes.

❖ **Changing a catalog's status**

To display a catalog in Portal Interface, it must have a status of approved and be marked active.

- 1 In the Catalog Manager, select New from the Status menu.
- 2 In the detail view, right-click the catalog and select Approval Status | Approved from the pop-up.
- 3 Select Approved from the Status menu. The catalog displays in the approved detail view.
- 4 To make this catalog display in Portal Interface, right-click the catalog in the detail view and select Active | Yes.
- 5 To update Portal Interface with the new catalog contents, click Update on the Catalog Manager toolbar.

❖ **Moving an existing branch**

When a catalog displays in Portal Interface, the applications display in the order in which they appear beneath a branch in the catalog tree. Move branches or applications up and down according to how you want the branches and their associated applications to appear in Portal Interface.

- 1 In the Catalog Manager, select the status of the catalog you want to edit.
- 2 In the detail view, right-click the catalog you want to change and select Edit from the Catalog Menu. The catalog displays in the Catalog Builder.
- 3 In the Catalog tree, select the branch you want to move.
- 4 Right-click and select Move Up or Move Down from the pop-up.

❖ **Deleting an existing branch**

- 1 Select the branch you want to delete.
- 2 Right-click and select Delete Branch from the pop-up.

❖ **Renaming an existing branch**

- 1 Select the branch you want to rename.
- 2 Right-click and select Rename Branch from the pop-up.
- 3 Type the new name in the text field that appears.
- 4 Click OK.

❖ **Deleting an application from a catalog**

- 1 To remove an application, select the branch with that application in the catalog tree.
- 2 In the application list detail view, right-click the application to remove and select Delete Application from the pop-up.

## Updating Portal Interface with changes

When you add or change portal objects—catalogs, pages, and pages groups—you must update Portal Interface with those changes.

Each of these objects has an Update icon on the Manager toolbar; pages and page groups also have an Update Users option on the right-click menu that functions the same as the Update icon.

❖ **Updating changes to Portal Interface**

- 1 Select Build | Catalogs in the left pane.
- 2 When the Catalog Manager displays, select Update from the toolbar.
- 3 When a dialog box confirms that the update was successful, click OK.

# Building Pages

This chapter describes how to use the Page Manager to create, edit, and populate portal pages.

Topic	Page
Overview	271
Choosing the page type	273
Selecting a page layout	274
Creating and editing pages	277
Updating Portal Interface with changes	280

## Overview

A page is a collection of applications that displays in a portal environment, such as Portal Interface. You also use pages and page groups as containers for applications that you deploy to mobile devices. In Mobile Web Studio, you use the Page Manager and Page Builder to create and manage portal pages.

In Portal Interface, pages only exist as part of a page group, although you can update Portal Interface with both individual pages and page groups.

To create page groups and add a page to a page group, see Chapter 10, “Building Page Groups.”

## Page Manager

Use the Page Manager to perform management functions for portal pages. Depending on your access rights, you can create, and edit pages and populate pages with applications.

To display the Page Manager, select Build | Pages on the Mobile Web Studio left pane.

The Page Manager interface includes:

- View By drop-down list – displays pages in the detail view by Status or Type.
- Page Manager menu – the options vary according to your selection in the View By drop-down list.
  - Status – see pages for the selected workflow status: New, Pending, Approved, Deleted, Rejected, Broken, or Archived.
  - Type – see pages for Default, Catalog, or Guest pages.
- Toolbar – displays the New and Edit buttons that let you add, view, and edit pages. Update lets you update the portal with active, approved new or changed pages. You can also filter whether to display only your pages, all pages, or only active pages in the detail view.
- Status bar – shows the current user and resource. Activities that you perform are associated with the displayed resource.
- Detail view – displays pages based on your selections in the View By drop-down list, the Page Manager menu, the Filter By, and Show Active Only options.
- Pop-up Page Menu – these options let you add and edit pages, update pages in Portal Interface, view page properties, and change a page’s status.

## Page Builder

Use the Page Builder to create and edit portal pages. To access the Page Builder, select Build | Pages from the Mobile Web Studio left pane, then click the New icon on the Page Manager toolbar, or right-click in the detail view and select New from the pop-up.

To access the Page Builder for an existing page, select the page from the detail pane, then click Edit.

The Page Builder interface includes:

- Toolbar – click Save to save an existing page with the same name. Click Save As to save a new page or save an existing page under a different name.

- Layout – select any of the four layout options (30/70, 50/50, 25/50/25, or Full) to change the page’s column layout. See “Selecting a page layout” on page 274.

---

**Note** Unwired Accelerator allows you to design pages with three columns in Mobile Web Studio and Portal Interface. For Mobile Web Studio instructions, see “Selecting a page layout” on page 274. For Portal Interface instructions, see the *Portal Interface User’s Guide*.

---

- Buttons – Add, Preview, and Remove applications.
- Application list – the large box in the left pane lists the names of applications you have added to this page.
  - Invisible – lets you hide a specific application from the end user. For example, you could set the display status of an application to invisible because the application is performing a server-side lookup operation.
  - Active – shows whether each application is active.
- The right pane displays a representation of the page. As you add applications, smaller boxes that represent the applications display according to the layout you have selected. You can drag and drop content from one cell to another and right-click any cell to select a custom cell width. See “Selecting a page layout” on page 274.

## Choosing the page type

Mobile Web Studio supports several page types: default pages, catalog pages, guest pages, and personal pages.

---

**Note** Default pages, catalog pages, and guest pages are created in Mobile Web Studio. Personal pages are created in Portal Interface.

---

- Default – active and approved default pages are automatically added to new Portal Interface accounts. Portal users can use one or more default pages based on their roles.

- Catalog – predefined page available for users to add to their personal page groups by selecting Add Page in Portal Interface.

---

**Note** Do not confuse catalog pages with catalogs that group applications. The default display catalog allows users to add applications to their personal pages when they select Add Application in Portal Interface.

---

- Guest – guest pages display when you access the Portal Interface URL. These pages are visible to new users before they log in or register for an account.
- Personal pages – created by a user in Portal Interface. Personal pages do not display in Mobile Web Studio, although a user can share their personal pages with other Portal Interface users. See the Portal Interface online help for instructions.

## Selecting a page layout

Mobile Web Studio lets you control the layout of a page, offering a variety of column widths and the ability to customize individual cell widths.

---

**Note** Portal Interface users can change the layout only of their personal pages; they cannot change the layout of system or catalog pages created in Mobile Web Studio.

---

A portal page can have these layouts:

- 100 – the content occupies 100% of the Portal Interface browser window.
- 30/70 – 2 columns, with one column occupying 30% of the browser window, and the other column occupying 70%.
- 25/50/25 – 3 columns, with one column occupying 25% of the browser window, the center column occupying 50%, and the last column occupying 25%.

- 50/50 – 2 columns with each column occupying 50% of the browser window.

---

**Warning!** The page layout can display incorrectly if a user has not been assigned a role to view some of that page’s applications. The user may see only some of the pages applications and cannot change the layout. You may want to keep this in mind when choosing a layout for pages that contain applications that have been assigned different access permissions.

---

❖ **Changing a page’s layout**

- 1 To change a page’s layout, click a different layout choice. The selected layout displays in the page representation on the right side of the Page Builder.
- 2 In a 30/70, 25/50/25, or 50/50 layout, click the application box in the page representation and drag and drop the application from one side of the layout to the other.
- 3 Click Save.
- 4 Click Close to exit the Page Builder.

❖ **Creating a three-column page**

- 1 Log in to Mobile Web Studio, select Build | Pages, then click New.  
The default page layout of two equally wide columns displays.
- 2 Click the three-column Layout button.  
The page switches to a three-column layout, with a default column percentage width of 25/50/25.
- 3 Click Add to begin adding applications.

---

**Note** A three-column layout supports a maximum of 15 rows, which lets you add a maximum of 45 applications to a page.

---

- 4 After you finish adding applications, change the default column and application widths to suit your needs.

To change the width of a column, click the Column Width drop-down list above each column. To change the width of one application, right-click the application and select the width from the pop-up.

---

**Note** Once you set a custom cell width, the Column Width drop-down lists disappear. To redisplay the drop-down lists, click the three-column layout icon again.

---

Both column widths and application widths are specified in percentages of the page width. For example, the default width of 25% means that the column or application occupies 25 percent of the width of the page.

If you change the page layout from 3 columns to 2 columns on a page with 30 or fewer applications, or move to one column from 2 or 3 columns on a page with 15 or fewer applications, the applications are randomly rearranged.

For example, if your page layout is like this:

A	B	C
D	E	F
G	H	I

and you remove the third column, the resulting two-column page looks like:

A	B
D	E
G	H
C	F
I	

The total percentage of cell widths in a row must equal 100%.

You can also set a column width to zero. This allows you to simulate having a row with only one or two columns, or lets you create two-column layouts with column widths other than 50/50 or 30/70.

When a user logs in to Portal Interface, they see the applications laid out in the selected configuration.

---

## Creating and editing pages

---

**Note** When you add, modify, or delete portal objects— pages, and page groups—you must update Portal Interface with those changes. See “Updating Portal Interface with changes” on page 280.

---

Portal Interface users can add applications only to their own user-created pages and page groups.

---

**Note** Do not mix insecure and secure applications on the same page.

---

### ❖ Creating a page

- 1 Select Build | Pages from the Mobile Web Studio left pane.
- 2 Click New on the Page Manager toolbar.
- 3 Click the icon that represents the layout you want for this page.
- 4 Click Add to populate the new page with applications.
- 5 When the standard Search window displays, complete the Search criteria to display a list of active approved applications in the Results window. Leave the search criteria blank to see all active approved applications.
- 6 Click Search.
- 7 When the search results display, select an application in the Results pane. To select several applications, hold down the Ctrl key while you click the applications you want to add in the Results pane.
- 8 Click Preview to display the application in a separate browser window.
- 9 Click Add to add the selected applications to the page you are building. You return to the Page Builder, which now displays the name of the applications you added. You also see a representation of the applications in the selected layout page representation on the right side of the window.
- 10 Select the layout. For more information, see “Selecting a page layout” on page 274.
- 11 In the application list, select or deselect these options:
  - Invisible – lets you hide a specific application from the end user. For example, you could set the display status of an application to invisible because the application is performing a server-side lookup operation.

- Active – shows whether each application is active. Deselect Active to remove this application from the active list. The application will not display on the page until you reselect this option to check the box.
  - Select an application in the list and click Preview to view the application’s contents in a separate browser window.
  - Select an application in the list and click Remove to remove the application from the page.
- 12 When you are satisfied with your choices, click Save.
- 13 Complete or select these options:
- Name – enter a unique page name.
  - Type – select Default, Catalog, or Guest Page from the drop-down list.
  - Active – select this option to mark this page active. When you mark a page as active, give the page the Approved status, and update pages to Portal Interface, this page is available to portal users, depending on their access rights.
  - Roles – select roles from the Available Roles list and add them to the Assigned Roles list to define what roles can access this page. See the *Enterprise Security Administration Guide* for more information on security roles.

---

**Note** User accounts with the roles in the Assigned Roles list are the only user accounts that can access this page.

---

- 14 Complete or select these options. When you select any of these options, they apply to all applications on the selected page.
- Enable Page Bar – displays the Change Layout, Add Application, Create Application, Share Page, Delete Page, and Refresh Page options to the left of the page group drop-down list.
    - Enable Refresh Page – puts the “Refresh Page” option on the page bar.
  - Enable Applications Title Bar – adds a title bar to the applications on this page.
    - Enable Applications Title Bar’s Widget – adds widgets (graphic icons) that let the user maximize or minimize an application.
  - Enable Applications Border – places a border around the applications.

- Applications Border Color – accept the default color or enter a different hexadecimal code for the border color you want. You can find a list of Web-safe hexadecimal color codes at <http://html-color-codes.com/> or at <http://www.visibone.com/colorlab/>.
- Enable Applications Scroller – places a scroll bar on the application if the content is larger than the specified application size.

---

**Note** Although Portal Interface users can change the layout of their personal pages, they cannot change the page properties listed.

---

- Vertical Spacing – the percentage of empty page space to put between the application columns.
  - Horizontal Spacing – the percent of empty page space above and below each application row. Horizontal spacing does not add space to the top of the page.
  - Left Margin Spacing – the percent of empty page space to place on the left side of the page.
  - Right Margin Spacing – the percent of empty page space to place on the right side of the page.
- 15 Click OK to save your entries.
  - 16 Click OK again to save your entries on the Save Page window.
  - 17 Click Close to exit the Page Builder.

❖ **Editing a page**

- 1 Double-click a page listing in the detail view, or
- 2 Right-click a page listing in the detail view and select Edit, or
- 3 Click to select a page in the detail view and click Edit on the toolbar.

❖ **Removing applications from a page**

- 1 In the Page Builder, select the application you want to remove.
- 2 Click Remove to remove the selected application.
- 3 Click Save.
- 4 Click Close to exit the Page Builder.

❖ **Viewing or editing page properties**

- 1 To view or edit a page's properties, right-click the page listing in the Page Manager detail view and select Properties from the pop-up.
- 2 Edit any of the fields.
- 3 Click Properties to view and edit additional page parameters.
- 4 Click OK to save your changes.
- 5 In the confirmation pop-up, click OK to save the changes from the first window of properties.

❖ **Deleting a page**

- 1 Select the page's current status from the Page Manager Status menu to display the page listing in the detail view.
- 2 Right-click the page listing in the detail view, select Status, then select Deleted from the Page Menu.

---

**Note** The page listing moves to the Deleted status in the detail view. The page is not removed from the database.

---

- 3 To change the page's status once it has been deleted, select Deleted from the Page Manager Status menu.

## Updating Portal Interface with changes

When you add, or modify portal objects—catalogs, pages, and pages groups—you must update Portal Interface with those changes.

Each of these objects has an Update icon on the Manager toolbar. Pages and page groups also have an Update Users option on the right-click menu you access in the detail view.

❖ **Updating page changes to Portal Interface**

- 1 Select Build | Pages in the left pane.
- 2 When the Page Manager displays, right-click in the detail view and select Update Users.
- 3 When a dialog box confirms that the update was successful, click OK.

# Building Page Groups

This chapter discusses how to create page groups.

Topic	Page
Overview	281
Choosing a navigation style	283
Page groups, page numbers, and <IFRAME>	289
Creating and editing page groups	289
Updating Portal Interface with changes	293

## Overview

A page group is a set of pages that are grouped together. You can assign different roles to different page groups to restrict access to only the page groups relevant to specific users.

Page groups created by developers in Mobile Web Studio are called *system page groups* (SPG), as opposed to page groups created by Portal Interface users, which are *personal page groups*. SPGs represent default or guest page groups.

For a system page group to get updated to Portal Interface, the “page.group” property must be set to “on” in *global.properties.xml*. When the page.group property is set to “off,” system page groups are still accessible in Portal Interface, but the Mobile Web Studio update operation is disabled. See the *Unwired Accelerator Administration Guide* for more information on this file.

---

**Warning!** Do not set the page.group property to “off,” then back to “on.” These actions corrupt the Unwired Accelerator onepage resource.

---

## Page Group Manager

The Unwired Accelerator Page Group Manager allows you to create a page group in Mobile Web Studio, then deploy it to Portal Interface. You can also use page groups as “containers” for deploying mobile applications to mobile devices such as BlackBerry.

The Page Group Manager interface includes:

- View By drop-down list – displays page groups in the detail view by Status or Type.
- Page Group menu – the options vary according to your selection in the View By drop-down list.
  - Status – display page groups for the selected workflow status: New, Pending, Approved, Deleted, Rejected, Broken, or Archived.
  - Type – display page groups for Default or Guest page groups.
- Toolbar – displays the New and Edit buttons that let you add, view, edit, or delete page groups. Update lets you update the portal with active, approved new or changed page groups. You can also filter whether to display only your page groups, all page groups, or only active page groups in the detail view.
- Status bar – shows the current user and resource. Activities that you perform are associated with the displayed resource.
- Detail view – displays page groups based on your selections in the View By drop-down list, the Page Group Manager menu, the Filter By, and Show Active Only options. You can filter whether to display only your page groups, all page groups, or only active page groups.
- Pop-up Page Group Menu – these options let you add and edit page groups, update page groups to Portal Interface, view page group properties, and change a page group’s status.

## Page Group Builder

Use the Page Group Builder to create and edit portal page groups. To access the Page Group Builder, select Build Page Groups from the Mobile Web Studio left pane, then click the New icon on the Page Group Manager toolbar or right-click in the detail view and select New from the pop-up.

To access the Page Group Builder for an existing page group, select the page group from the detail pane, then click Edit.

The Page Group Builder interface includes:

- **Toolbar** – click **Save** to save an existing page group with the same name. Click **Save As** to save a new page group or save an existing page group under a different name.
- **Buttons** – **Add** and **Remove** pages.
- **Page list** – the large box in the left pane lists the names of pages you have added to this page group.
  - **Active** – shows whether each page is active.
- The right pane displays a representation of the page group. As you add pages, additional boxes that represent each page display according to the order in which each page is added. The box at the top represents the page that displays first in Portal Interface when that page group is selected. You can drag and drop page boxes up and down to change the order in which the pages display.

## Choosing a navigation style

Unwired Accelerator provides *navigation styles*, which let you specify how users pages display based on the user's operating system and browser. You specify a navigation style when you create a page group.

Unwired Accelerator refers to the combination of a particular browser and operating system as a *user agent*. A user agent consists of only one combination of a browser and operating system; for example, Internet Explorer on Windows CE (a typical PocketPC configuration). A navigation style specifies one or more agents for use with a page group.

See for instructions on setting up

Every page group has the navigation style "Default." You can add others by editing the *navstyles.xml* file located in `%SYBASE%\tomcat\webapps\onepage\fw\properties` if you are using Tomcat as your application server. If you are using EAServer, it is located in `%JAGUAR%\Repository\WebApplication\onepage\fw\properties`.

## Navigation styles

Unwired Accelerator includes several navigation styles that allow you to format application content to display properly on a mobile device. These styles display in Mobile Web Studio only when you save a new page group or edit an existing page group; that is, the navigation style options are not available when you save or edit applications or pages.

---

**Note** Navigation styles also display in Portal Interface when you click the Manage Pages icon, and then click Edit next to a page group, to display the Save Page Group dialog box.

---

The *navstyles.xml* configuration file is located in *%SYBASE%\tomcat\onepage\fw\properties*. The default navigation styles provided in UA 8.0 are:

- Full HTML and frame support (desktop browsers). This is the default.
- PocketPC/PDA HTML with frames like Pocket IE on PocketPC 2003.
- MbusinessClient/PDA HTML without frames, like the M-Business Anywhere client.
- Phone-WML – phones with Wireless Markup Language browser.
- Phone-HTML – phones with HTML browser.

## Content style

Content style (device type) determines the application playback presentation. Each content style has a template associated with it. The default template for each of the content styles is defined in *oem.xml*. The Device Type drop-down list in the Application Builder displays the list of content styles. On application playback, the UWP applies the template associated to the content style and the content style is mapped to a UserAgent.

---

**Note** For Wireless Access Point (WAP) Phones, you can define a custom template for application playback by implementing an interface.

---

## UserAgentMapping

*UserAgentMapping.xml*, located in `%SYBASE%\tomcat\webapps\onepage\fw\properties`, maps the UserAgents to the navigation and content styles. Several UserAgent mappings are defined by default. For example:

```
<UserAgentMapping>
  <UserAgent clientpattern="Mozilla">
    <NavStyle name="Default"/>
    <Content name="portal" type="text/html"/>
  </UserAgent>
<UserAgent clientpattern="MSIE" platform="PPC">
  <NavStyle name="PocketPC/PDA"/>
  <Content name="portal" type="text/html"/>
  <ResolutionHeader name="UA-pixels"/>
</UserAgent>
```

The `clientpattern` and `platform` attribute values are from the client's User-Agent HTTP header. The User-Agent header usually contains the platform and browser information. In device detection, Unwired Accelerator matches the platform substring against the User-Agent header first.

The name attribute value in the `NavStyle` element is from *navstyles.xml*. This specifies the Navigation Style the UserAgent supports. The name attribute value in `Content` element is from *oem.xml*. This specifies the Content Style that application playback should use for the UserAgent.

### ❖ Defining a custom template for Content Style

- 1 Implement the interface

*com.onepage.fw.uwp.interfaces.uwp.UWPMobileTemplate:*

```
String getTemplate
```

This returns the template string name, or if there are any errors, `NULL`. The template string name should match one of the names listed in the Template Manager in Mobile Web Studio.

- 2 Compile it. All the required JARs are in *WEB-INF\lib*.

Check the XML data by going to:

```
http://host.domain.com/onepage/servlet/UWPServlet?wid=<wid>&content=text/xml
```

<wid> is the application ID you see in the application properties in Application Builder.

- 3 Move the implementation in the *WEB-INF\classes* directory.
- 4 Set “WAPCustomTemplate:<class package>” in the name attribute of the Content element. For example:

```
<Content name="WAPCustomTemplate:com.sybase.template.MyTemplate"
```

Defining a custom template example

In this example, the OpenWave Basic template is returned. The OpenWave Basic template is located in *%SYBASE%\tomcat\webapps\onpage\portlets\templates\mobile\openwave.xml*.

```
package com.sybase.ep.utils;

import com.onpage.fw.uwp.interfaces.uwp.*;
import javax.servlet.http.*;
import java.io.*;

/**
 * Sample custom template for openwave browsers
 */
public class OpenWaveTemplate implements
UWPMobileTemplate
{
    public String getTemplate (UWPRequest request)
    {
        HttpServletRequest req =
(HttpServletResponse)request.GetServletRequest();
        String agent = req.getHeader("User-Agent");
        if (agent.indexOf ("UP.Browser") >= 0) {
            return "OpenWave Basic";
        }
        else {
            return null;
        }
    }
}
```

For this example, the custom template is used on OpenWave browsers, so you must modify the content in *UserAgentMapping.xml* to say:

```
<UserAgent clientpattern="UP.Browser" platform="UP.Browser">
    <NavStyle name="Phone-WML"/>
    <Content name="WAPCustomTemplate:com.sybase.ep.utils.OpenWaveTemplate"
type="text/vnd.wap.wml"/>
</UserAgent>
```

## Sample *navstyles.xml* file

The code fragment below shows a typical *navstyles.xml* file. The example is from the FWPageServer Web application.

```
<?xml version="1.0"?>
<NavigationStylesConfigDoc>
<!--Default supports Netscape, IE browsers - Mozilla user agents -->
<!--Will be this the generic one for all devices/clients-->
  <NavStyle name="Default" clientpattern="Mozilla">
    <AppDefinition name="FWTopMenu">
      <Target default="topmenu_mop_ie" alternate="topmenu_mop_ie" />
    </AppDefinition>
    <AppDefinition name="FWPageServer">
      <Target default="main_IE" alternate="main_IE" />
    </AppDefinition>
  </NavStyle>
<!-- support IE browsers on PocketPC devices running on Windows CE -->
  <NavStyle name="PocketPC" clientpattern="MSIE" platform="Windows CE">
    <AppDefinition name="FWGuestPage">
      <Target default="guest" alternate="ppclogin" />
    </AppDefinition>
    <AppDefinition name="FWLogin">
      <Target default="default" alternate="default_ppc" />
    </AppDefinition>
    <AppDefinition name="FWMyOnePage">
      <Target default="myonepage_ie" alternate="myonepage_ppc" />
    </AppDefinition>
    <AppDefinition name="FWTopMenu">
      <Target default="topmenu_mop_ie" alternate="topmenu_ppc_ie" />
    </AppDefinition>
    <AppDefinition name="FWPageServer">
      <Target default="main_IE" alternate="main_ppc_ie" />
      <Target default="emptypage" alternate="main_ppc_ie" />
    </AppDefinition>
    <AppDefinition name="FWTileServer">
      <Target default="standard_tile" alternate="standard_tile_ppc_ie" />
    </AppDefinition>
  </NavStyle>
</NavigationStylesConfigDoc>
```

The `NavStyle` element specifies the navigation style name. The `AppDefinition` element specifies the Model View Controller (MVC) application. The default attribute in the `Target` element specifies the default logical name returned by the MVC application command, and the `alternate` attribute specifies the alternate logical name to use for the style. These logical names must exist in the applications View XML file (see “Sample \*View.xml file” on page 288). For

example, the guest page for the PocketPC style shown in the preceding code has the default logical name of guest and the alternate logical name of ppclogin.

## Sample \*View.xml file

The code fragment below shows a typical \*View.xml file. This example uses `%JAGUAR%\Repository\WebApplication\onpage\fw\baseApps\fwpageserver\FWPageServerView.xml` for the FWPageServer Web application.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  User Interface Definition of FWPageServer
-->
<AppView xmlns="x-schema:http://localhost/fw/schema/framework-view.xml">
  <AppName>FWPageServer</AppName>
  <AppPage>
    <Target logicalname='main_IE'>
      <Resource realname='/fwdisplaystaticfile/htmlHeader.jsp'/>
      <Resource realname='/fwdisplaystaticfile/style/css.jsp'/>
      <Resource realname='/fwpageserver/pageserver_top.jsp'/>
      <Resource realname='/fwpageserver/pageserver_body_ie_div.jsp'/>
      <Resource realname='/fwpageserver/pageserver_body_ie_iframes.jsp'/>
      <Resource
        realname='/fwpageserver/pageserver_body_ie_mixed_divIFrame.jsp'/>
      <Resource realname='/fwpageserver/pageserver_runscript_ie.jsp'/>
      <Resource realname='/fwpageserver/pageserver_tabsetDropDown.jsp'/>
      <Resource realname='/fwdisplaystaticfile/footer.jsp'/>
    </Target>
  </AppPage>
  <AppPage>
    <Target logicalname='pocketpc_main_IE'>
      <Resource realname='/fwdisplaystaticfile/htmlHeader.jsp'/>
      <Resource realname='/fwdisplaystaticfile/style/css.jsp'/>
      <Resource realname='/fwpageserver/pageserver_top.jsp'/>
      <Resource realname='/fwpageserver/pageserver_body_ie_mac.jsp'/>
      <Resource realname='/fwpageserver/pageserver_runscript_ie_mac.jsp'/>
      <Resource realname='/fwpageserver/pageserver_tabsetDropDown.jsp'/>
      <Resource realname='/fwdisplaystaticfile/footer.jsp'/>
    </Target>
  </AppPage>
</AppView>
```

## Page groups, page numbers, and <IFRAME>

When you build a page group, each page you add is wrapped in an <IFRAME> tag, which allows the page to display on a page group tab and not pop-up in a separate browser window.

---

**Note** <IFRAME> creates a frame that sits in the middle of a regular nonframed Web page. <IFRAME> works like <IMG>, only instead of putting a picture on the page, it puts another Web page on the page

Versions of Netscape prior to 6.0 do not support the <IFRAME> tag.

---

The number of IFRAMEs (pages) contained in a page group is saved in the portal database column of the page group (tabsets) table. When multiple page groups exist with different values for the number of pages, the portal uses the maximum number of pages specified and uses that number for that log in session. If the maximum number of pages is more than the number of pages in a particular page group, some pages are left unused. However, if the maximum number of pages is less than the number of pages in a particular page group, it is possible that more than one page will share an IFRAME and switching between those pages will cause pages to reload.

## Creating and editing page groups

---

**Note** When you add, modify, or delete portal objects—catalogs, pages, and pages groups—you must update Portal Interface with those changes. See “Updating Portal Interface with changes” on page 293.

---

Portal Interface users can add applications only to their own user-created pages and page groups.

### ❖ Creating a page group

- 1 In Mobile Web Studio Select Build | Page Groups from the left pane.

The Page Group Manager opens. The Status menu allows you to navigate page groups by their status. To see all page groups of a specific status, deselect the Show Active Only check box.

- 2 Click New. The Page Group Builder opens.

- 3 Click Add to begin adding pages to the page group. To eliminate a page from the page group, select the page and click Remove.

A new Search window opens. You can search for pages by page name, page owner, and date of last modification. To locate a page, enter search criteria and click Search.

- 4 Select the page you want to add, then click Add.

---

**Note** You can select multiple pages by holding down the Ctrl key while selecting page names.

---

The Search window closes, and the pages you selected appear in the Page Group Builder.

- 5 Click Save As.
- 6 When the Save Page Group window appears, complete these fields:
  - Name – enter a name for the page group.
  - Resource – displays the resource associated with the user.
  - Type – select the page group type. Page groups have two types: Default and Guest. Default is for registered portal users; Guest is for guests.

---

**Note** You can add pages only to page groups of an identical type; for example, you can add guest pages only to a Guest page group.

Guest and Default pages cannot be updated in the Page Manager when the *global.properties.xml* page.group property is set to “on,” and must be updated by changing the guest and default page groups.

---

- Active – indicate whether the page group is active. Selecting the Active check box makes the page group active for the roles you select, after you approve the page group.
- Specific Number of Frames – specify the number of frames that should be used to render the pages of the page group. The default value is the number of pages that you added to the page group. For more information, see “Page groups, page numbers, and <IFRAME>” on page 289.

- Initial on Load – select this option to load all the applications on the different <IFRAMES> the first time a user logs in. This causes the listeners to register and receive events immediately. See “Page groups, page numbers, and <IFRAME>” on page 289 for more information.
- Navigation Style – select a navigation style. Navigation styles lets you assign a custom navigation type to the page group, such as navigation settings for a PocketPC. For more information about navigation settings, see “Choosing a navigation style” on page 283.
- Roles – assign roles to the page group. When you assign a user role to a page group, all users assigned that role can see the page group.

---

**Note** If a page in a page group is not assigned a role that is assigned to the group, the page does not display to users without the assigned role.

---

7 Click OK to save the page group.

8 Select Status | New in the Page Group Manager.

Right-click the page group you created and select Status | Approved to make the page group available.

#### ❖ Editing page group contents and properties

You can add or remove pages from a page group, or modify a group’s properties after you create it.

- 1 In Mobile Web Studio, select Build | Page Groups from the left pane.
- 2 When the Page Group Manager appears, navigate the Status menu to find the page group whose contents you want to edit, select the page group in the detail view and click Edit.
- 3 When the Page Group Builder displays, click Add to begin adding pages to the page group. A Search window opens. You can search for pages by page name, page owner, and date of last modification. To locate a page, enter search criteria and click Search.
- 4 Select the page you want to add, then click Add.

---

**Note** You can select multiple pages by holding down the Ctrl key while selecting page names.

---

The Search window closes, and the pages you selected appear in the Page Group Builder.

- 5 To eliminate a page from the page group, select the page in the Page Group Builder and click Remove. Click Save to save your changes.
- 6 To edit a page group's properties, right-click the page group in the detail view and select Properties. The Page Group Properties window appears. Follow the instructions in "Creating a page group" on page 289 to change the page group properties. Click OK when to finish and save your changes.

❖ **Disabling a page group**

To disable a page group and prevent it from showing in Portal Interface:

- 1 In Mobile Web Studio, select Build | Page Groups.
- 2 When the Page Group Manager appears, navigate the Status menu to find the page group that you want to disable.
- 3 Right-click the page group in the detail view and select Properties.
- 4 When the Page Group Properties window displays, deselect the Active option, then click OK.
- 5 When the confirmation that the page group was saved displays, click OK.

❖ **Marking a page group as deleted**

- 1 Select the page group's current status from the Page Group Manager Status menu to display the listing in the detail view.
- 2 Right-click the listing in the detail view, select Status from the pop-up, then select Deleted.

---

**Note** The page group listing moves to the Deleted status in the detail view; however the page is not removed from the database.

---

- 3 To change the page group's status once it has been marked as deleted, select Deleted from the Page Group Manager Status menu.
- 4 Right-click the page group listing in the detail view, select Status from the pop-up, then select any status that displays to recover the page group.  
The page group moves to the detail view of the status you selected.

## Updating Portal Interface with changes

When you add, modify, disable, or delete portal objects—catalogs, pages, and pages groups—you must update Portal Interface with those changes.

Each of these objects has an Update icon on the Manager toolbar and an Update Users option on the right-click menu you access in the detail view.

### ❖ Updating page changes to Portal Interface

- 1 In Mobile Web Studio, select Build | Page Groups in the left pane.
- 2 When the Page Group Manager displays, right-click in the detail view and select Update Users.
- 3 When a dialog box confirms that the update was successful, click OK.



# Deploying Mobile Applications

This chapter describes how to deploy mobile applications to mobile devices and the Portal Interface.

Topic	Page
Overview	295
Deploying to BlackBerry and Symbian/J2ME clients	296
Deploying to online devices	304
Deploying to Portal Interface	306

## Overview

Unwired Accelerator provides the ability to easily deploy mobile applications to mobile devices. You can deploy mobile applications to a variety of devices, including PDAs, BlackBerry, Symbian/J2ME, and SmartPhones.

After you create and approve an application, you need to decide how to deploy it. Following are a few guidelines to help in the process:

- Keep in mind whether devices are used in online or offline mode.
- Keep in mind that some UA features are not available on all devices, or in all modes. For example, the insert/update/delete features, and the linked applications and linked parameters features are not supported in online (browser) mode, only in offline mode; the e-mail and phone data types are only available in online mode.
- Online mode – to make the application available in online mode, create a page and page group combination for the application. The page and page group creates a container for the application, making it available to browsers running in online mode. For example, BlackBerry, Symbian/J2ME, Pocket PC, Portal Interface, and so forth. Typically, create a page/page group for every application.

You can see approved pages and page groups by selecting Build | Pages or Build | Page Groups. Ideally, use the application name for the page and page group names, so you can keep track of which applications are deployed.

- Offline mode (BlackBerry and Symbian/J2ME) – to make an application available to BlackBerry and Symbian/J2ME clients in offline mode, you must select the “Make available for disconnected mobile devices” option on the Mobile tab (see “Mobile tab” on page 79 for information).

You can see the applications assigned to offline mode by selecting Build | Applications in the left pane, selecting Approved, and checking the Offline column. A check in the Offline column indicates offline mode.

- Offline mode (PDAs) – to make an application available to PDAs in offline mode, you must deploy to Mobilink. To do so, select the approved application, right-click, select Deploy to MobiLink, and click OK to confirm when prompted.

You can see a list of applications deployed to MobiLink by selecting Manage | MobiLink in the left pane.

- M-Business (connected and disconnected) – to make an application available to M-Business clients, you must deploy to M-Business through public or group channels. The mobile applications are stored in the M-Business Anywhere AGDB database, and deployed to mobile devices when the user logs on, either in connected or disconnected mode depending on the mobile device capabilities.

---

**Note** To deploy an application to M-Business Anywhere, the application can contain only one element; multiple-element applications are not compatible as mobile applications.

---

## Deploying to BlackBerry and Symbian/J2ME clients

This section describes how to set up applications to be accessible on BlackBerry and Symbian/J2ME clients, in offline mode and online modes. To make a mobile application available for BlackBerry and Symbian/J2ME devices, you must set the “Make available for disconnected mobile devices” property.

See “Using the BlackBerry user interface” on page 299, the *Quick Start Guide*, and the *Mobile Application Development Tutorial* for information about using mobile applications on BlackBerry devices.

See “Using the Symbian/J2ME user interface” on page 302, the *Quick Start Guide*, and the *Mobile Application Development Tutorial* for information about using mobile applications on Symbian/J2ME devices.

❖ **Setting up applications (offline mode)**

- 1 In Mobile Web Studio, select Applications from the Build menu in the left pane, select Approved under Application Manager, and select an existing application.
- 2 Click Edit.
- 3 In the Application Builder, select Properties. The Properties Editor displays.
- 4 In Properties Editor, click the Mobile tab, and select “Make available for disconnected mobile devices,” which makes this application viewable on a BlackBerry or Symbian/J2ME device during synchronization.

---

**Note** You need to select “Make available for disconnected mobile devices” only for the master application, not its linked applications.

You can set the “Make available for disconnected mobile devices” property when you first create the application, or after it has been created and approved.

---

- 5 Click OK and save the application.
- 6 In the Application Builder, click Preview. The application displays in the Preview panel.
- 7 Click Close to exit Application builder.

❖ **Setting up applications (online mode)**

- 1 In Mobile Web Studio, create a page for the application:
  - Select Pages in the left pane, and click the New button.
  - Click Add.
  - Click Search.
  - Select the application, and click Add.
  - Click Save.

- For Name, enter the application name, and select one or more roles, or click Add All to select all roles.
  - Click OK to save, and OK to confirm.
  - Click Close to exit the Page Builder window.
  - Approve the page (click New under Page Manager, right-click the page, select Status | Approved, and click OK to confirm).
- 2 Create a page group using the page:
- Select Page Groups in the left pane, and click the New button.
  - Click Add.
  - Click Search.
  - Select the page, and click Add.
  - Click Save.
  - Enter the application name for the page group name, and select one or more Navigation Styles and Roles.
  - Click OK to save, OK to confirm.
  - Click Close to exit the Page Group Builder window.
  - Approve the Page Group (click New under Page Group Manager, right-click the page group; select Status | Approved; and click OK to confirm).
- 3 Deploy the page group:
- Select Page Groups, and Approved.
  - Select the page group in the detail pane.
  - Click the Update button, and click OK twice to confirm.
- 4 If you have not already done so, set up a Portal Interface user account for the page group:
- a Open a second Internet Explorer window, and enter the following in the Location field:
- ```
http://hostname.domain.com:port/onepage/mpindex.jsp
```
- For example:
- ```
http://labxp.sybase.com:4040/onepage/mpindex.jsp
```

- b Click Join Now, and set up the account using your initials for the member name and password, and PortalUser for the role.

## Using the BlackBerry user interface

See the *Unwired Accelerator Installation Guide* for instructions on installing the UA Client application onto your BlackBerry device. Once installed, the UA client icon displays on the screen of your device.

### BlackBerry device trackwheel menu

The UA client uses the standard BlackBerry device navigation model. The UA applications trackwheel has these menu items:

- AskUA – allows you to perform a search.
- Refresh All Apps – returns the list of available applications, from which you can select an application to sync to the device.
- Auto Refresh – refreshes all push registered applications in the background.
- Search All – allows you to perform a string search in the selected application.
- Profiles – allows you to configure and save information for connecting to the UA server.
- Settings – allows you to make UA settings such as: configure or disable the screen saver, lock the UA client, and set up a valid e-mail address to register for e-mail push notifications, or SMS phone number to register for SMS push notifications.
- Delete All Apps– deletes all applications or AskUA results.
- About UA – shows a screen with UA client information, copyright and version information, and technical information for RAM and Flash.
- Close – closes the UA client.

When you display an application, these menu options are available:

- Details – shows the details of an application.
- Click Thru – allows you to click through links in an application.

- *DeleteApplicationName* – this menu option is appended with the name of the application that is linked to it. This option enables you to remove a multi-level record.
- *InsertApplicationName* – this menu option is appended with the name of the application that is linked to it. This option enables you to insert a multi-level record.
- Search – allows you to search for any text in columns and rows in persistent storage. The search results display in the Grid/List view.

---

**Note** Search can be performed only on the grid level that you are viewing from. If you drill down, you can search on that level only.

---

- Sort on Column – allows you to sort columns in an application.
- *EditApplicationName* – this menu option is appended with the name of the application that linked to it. This option enables you to edit a multi-level record.
- Home – returns to the list of applications.

Applications typically are comprised of a List View and a Detail View. The list view includes the grid columns that the application developer included in the List/Detail tab (see “List/Detail tab” on page 82) and are usually the “key” to the record. You click on the magnifying glass icon to “drill down” to the Detail View, which typically includes more grid columns.

Some additional menu options you may see, include:

- Send Update – in Application List view or application view, you might encounter this menu option when there are pending edits/updates/inserts to be sent to UA server.
- Logs – In Application List view, when there are update/delete/insert actions done, this menu option is available, indicating logs are available for viewing.
- Contents For – this option appears when you have a CGI drop-down list. The list shows possible selection values; in some cases, selection values depend on previous selections made.

## Connection profile settings for synchronization

This section shows you how to set the connection settings for synchronization.

- 1 Select the Profiles menu. The Profiles List displays. Initially, the list is empty. On the Profiles List menu includes these options:
  - New Profile – create new profiles.
  - Edit Profile – edit information in selected profile.
  - Delete Profile – delete profiles.
  - Set as Active – shows which profile is currently active. When a profile is active, it means that is the one that is used when performing a “Refresh All Apps.” If there is only one profile in the list, it is active by default.
  - Advanced – from this option, you can select a push notification type. You can choose Email or SMS Push Notifications, and set up the e-mail address, or SMS number.
- 2 Select the New Profile menu to go to the Connection Settings window. Enter:
  - Profile – the profile to use, such as `mwsAdmin`. You can set up multiple user profiles.
  - User name – this is the user name you use for logging in to UA.
  - Password – the password you use to log in to UA.
  - Resource ID – a drop-down list with names of the registered resources and co-brands.
  - Server Name – fully qualified name of the machine where UA is installed.
  - Port Number – port number used to connect to UA.
  - Connect with HTTPS – select this option if you plan to connect to the UA server with HTTPS. Upon selection, the HTTPS port number, such as 4443, displays.
- 3 Set up the SMS phone number or e-mail address to use for synchronization.

## Inserting multi-level records

UA 8.0 allows you to insert multi-level records when using your BlackBerry device. The Insert menu option is seen when you open a linked application for viewing. See “Complex transactions” on page 144.

❖ **Performing a multi-level insert or modification**

- 1 Open the application, for which you want to insert a multi-level record, on your BlackBerry client.
- 2 On the trackwheel menu, you see Insert appended with the name of the linked application; for example, “InsertAcct” could indicate you would add a new record to a database application that performs an insert to the account table.
- 3 Select the Insert option, and make your changes.
- 4 Select Add to insert the application. The item is inserted into the table at the bottom.

---

**Note** Note that when you highlight an item in the table at the bottom, you see “Edit.” When you perform an “Edit” you see “Save” instead of “Add” since you are now in edit view.

---

If a second level insert application is linked to the original application, you can insert second level records based off a first level record.

- 5 To insert a second level record to an existing first level record, select a new inserted record and click on the trackwheel menu. You see the new menu with the Insert option, appended with the second level application name.
- 6 Select Finish when you are finished. You are prompted whether to auto send the inserts to UA server.

## Using the Symbian/J2ME user interface

See the *Unwired Accelerator Administration Guide* for instructions on installing the UA Client application onto your Symbian/J2ME device. Once installed, the UA client icon displays on the screen of your device.

The following UA features are not supported on Symbian/J2ME devices:

- Push synchronization
- Ask UA
- Applications with linked parameters

## UA client menus on Symbian/J2ME

The UA client uses the standard Symbian/J2ME device navigation model. UA Client home screen displays the applications that are available offline, and the date and time the applications were last synchronized. The UA client has these menu items:

- **Configure Profile** – allows you to configure and save profiles for connecting to the UA server. You can have multiple profiles and select the active profile.
- **Run** – retrieves an application from the device database, and displays it on the screen/
- **Refresh** – forces the sync of all available application content from UA server to the device database.
- **Refresh with images** – forces the sync of applications containing images from UA server to the device database.
- **View Log** – this option is only visible when there is only one application selected in the list, and that application has log data stored on the device.
- **Refresh All** – forces the sync of the list of applications available on UA server and displays it on the device screen.
- **Search** – search for a string across applications.

When you display an application, these menu options are available:

- **Delete** – removes the selected application from the device database.
- **Delete All** – deletes all the applications from the device database.
- **Settings** – displays a menu from which you can:
  - Set the device telephone number
  - Enable and disable the screen saver
  - Set the timeout for the screensaver
  - Change the password
- **Application Info** – shows application information such as name, profile, size, and the last sync date and time.

Applications typically are comprised of a List View and a Detail View. The list view includes the grid columns that the application developer included in the List/Detail tab (see “List/Detail tab” on page 82) and are usually the “key” to the record. You click on the magnifying glass icon to “drill down” to the Detail View, which typically includes more grid columns.

See “Connection profile settings for synchronization” on page 300 and “Inserting multi-level records” on page 301 for additional information that applies to Symbian/J2ME as well.

## Deploying to online devices

To deploy an application to an online device, such as PocketPC and PalmOS, BlackBerry or Symbian/J2ME browser, PDA browser, SmartPhone browser, and so forth, that has HTML or WML content, you must create a page and page group in Mobile Web Studio to act as a container for the application. Upon synchronization, the application is deployed to the PDA in the container.

See the *Mobile Application Development Tutorial* for information about using mobile applications on PDAs.

### ❖ **Deploying applications to online devices (disconnected mode)**

- 1 Select Build | Applications | Approved.
- 2 Select the application.
- 3 Right-click and select Deploy to MobiLink.
- 4 Click OK to confirm when prompted.

### ❖ **Deploying applications to online devices (connected mode)**

- 1 In Mobile Web Studio, create a page for the application:
  - Select Pages in the left pane, and click the New button.
  - Click Add.
  - Click Search.
  - Select the application, and click Add.
  - Click Save.
  - For Name, enter the application name, and select one or more roles, or click Add All to select all roles.
  - Click OK to save, and OK to confirm.
  - Click Close to exit the Page Builder window.
  - Approve the page (click New under Page Manager, right-click the application, select Status | Approved, and click OK to confirm).

- 2 Create a page group using the page you just created:
  - Select Page Groups in the left pane, and click the New button.
  - Click Add.
  - Click Search.
  - Select the page, and click Add.
  - Click Save.
  - Enter the application name for the page group name, and select the appropriate Navigation Styles and Roles.
  - Click OK to save, OK to confirm.
  - Click Close to exit the Page Group Builder window.
  - Approve the Page Group (click New under Page Group Manager, right-click the page group; select Status | Approved; and click OK to confirm).
- 3 Deploy the page group:
  - Select Page Groups, and Approved.
  - Select the page group in the detail pane.
  - Click the Update button, and click OK twice to confirm.
- 4 If you have not already done so, set up a Portal Interface user account for the page group:
  - a Open a second Internet Explorer window, and enter the following in the Location field:  

```
http://hostname.domain.com:port/onepage/mpindex.jsp
```

For example:  

```
http://labxp.sybase.com:4040/onepage/mpindex.jsp
```
  - b Click Join Now, and set up the account using your initials for the member name and password, and PortalUser for the role.

## PDA user interface

The PDA interface is similar to the browser interface used by Portal Interface. Enter a browser address to gain access:

`http://hostname.domain.com:port/onepage/mp.jsp`

The Guest Page displays. Click Join Now to set up an account. The following features are available on the PDA:

- Home – returns you to the list of applications.
- Refresh – refreshes the application content.
- Personalization – personalize applications for your use.
- Applications typically are comprised of a List View and a Detail View. The list view includes the grid columns that the application developer included in the List/Detail tab (see “List/Detail tab” on page 82) and are usually the “key” to the record. You click on the magnifying glass icon to “drill down” to the Detail View, which typically includes more grid columns.
- Paging – navigate to a particular page by clicking the page number.
- Multi-level inserts – for the PocketPC, the .NET container client supports multi-level inserts, as does any custom .NET client developed with the .NET API.

## Deploying to Portal Interface

Deploying to Portal Interface uses the same procedure as deploying to PDAs. Follow the procedures in “Deploying to online devices” on page 304. See the Portal Interface User’s Guide for information about using the portal.

# Importing and Exporting Portal Objects

This chapter describes how to import and export portal entities to and from Mobile Web Studio installations.

Topic	Page
Overview	307
Importing files into Mobile Web Studio	308
Exporting objects from Mobile Web Studio	311
Rules and restrictions	313
Exporting/importing parameters	316

## Overview

Mobile Web Studio allows you to move portal entities—catalogs, applications, page groups, pages, and agents—between different resources for the same UA installation and between installations by exporting their metadata to an XML document. The applications, templates, and personalization adapter database table keys associated with these entities are exported along with their parent objects to the same XML document. This allows you to easily synchronize the portal content between an isolated development environment and a production environment.

---

**Warning!** To use the Import/Export functionality, the Mobile Web Studio user must be granted the PortalAdmin role. When a user is not granted this role, the Import/Export option on the Mobile Web Studio toolbar is not available.

---

## Importing files into Mobile Web Studio

If you are on the receiving side of an exported archive file from a different Mobile Web Studio installation, use the Mobile Web Studio–Export/Import dialog box to import entities into your Mobile Web Studio installation. You must place the archived XML file that you want to import on the same server as your Mobile Web Studio installation.

---

**Note** If you are importing applications from a previous release, or are upgrading a UA server from 7.0 to 8.0, you might end up with templates associated with the application that are not appropriate for 8.0 or cause execution failures. For example, the BlackBerry Online template has changed with 8.0, the Nokia Online template is new, and applications created in 7.0 may use XSL templates that no longer work with 8.0.

To fix this, either edit the application so that each device type points to the correct template; or edit the template itself so that it is correct. If you edit the template, there is no need to edit the application. Decide which to do based on whether the problems is with the template definition, or with the application pointing to the wrong template.

---

### ❖ Importing a file into Mobile Web Studio

- 1 Log in to the Mobile Web Studio as a user that has the StudioAdmin and PortalAdmin role.
- 2 Select Build | Applications from the Mobile Web Studio left pane.
- 3 Click Import/Export on the Application Manager toolbar.
- 4 When the Export/Import window displays, click the Import tab.
- 5 Complete these fields:
  - File Name – enter the name of the XML file to import, or click Browse to locate the file using the Choose File dialog box.
  - Click Upload to upload the file to Mobile Web Studio. When a prompt confirms that the upload was successful, click OK.

---

**Note** Do not import and export documents from the `\tmp\upload\` directory if the client browser and the UA server are running on the same machine, as this truncates the XML file.

---

- Click View to view the XML file contents.

- **Import Catalog As Active** – make the imported catalog the default display catalog. The imported XML document can contain more than one catalog, but only one catalog is marked as the default display catalog—approved and active. If you import this catalog as active, it deactivates the current default catalog and replaces it with the new catalog. You can export more than one catalog at a time because agents can refer to catalogs that are inactive or not approved.
- **Override Existing Entities** – have imported entities (pages, page groups, applications, and agents) replace an equivalent entity with the same name; for example, replace a page with an imported page with the same name. These entities do not use versioning.

---

**Note** If you are importing applications from a previous version, such as UA 7.0, do not check the **Override Existing Entities** box. If you do, older templates will override new templates, causing problems with your applications.

---

- **Import Entities As Active** – import the new entities as active but the catalog as inactive, which makes the imported entities (but not the catalog) available to Portal Interface users.
  - **Set Deferred Update For All Portal Users** – indicates whether to notify all portal users of a change in the portal content so that they can refresh their view the next time they log in to the portal.

---

**Note** During the import process, if you do not select **Set Deferred Update For All Portal Users**, the Portal Interface user must perform an **Update** for each guest page or guest page group.

---

- 6 Click **Upload** to move the file to the UA server.
- 7 Click **Import**. When a prompt confirms that the import was successful, click **OK**.
- 8 Click **View Log** to review the import operation's activities.
- 9 Click **Close** to exit the **Mobile Web Studio–Export/Import** window.
- 10 Select **Build | Catalogs** from the **Mobile Web Studio** left pane.
- 11 Select **Approved** from the **Catalog Manager Status** menu. If you chose to import a new catalog as active, you should see the catalog in the detail view with a check mark in the **Active** column.

- 12 If you chose to import the catalog as active, click Update on the Catalog Manager toolbar to update Portal Interface on the same application server with the new default display catalog.
- 13 When you see confirmation that the operation was successful, click OK.
- 14 If you are importing mobile applications from a previous version of UA, test out the applications. You may need to repair them or redeploy them in the new version of UA for them to work. For example, you may need to:
  - Modify fixup JSP.
  - Modify templates.
  - Redeploy applications to MobiLink.
  - Recapture applications.

Optionally, you can add applications to Portal Interface:

- 1 Log in to Portal Interface.
- 2 You can add applications only to your personal page and page group.

If the page.group global property is turned on, create a page group, then add a new page to that page group.

If the `page.group` property is turned off, create a new page.

---

**Note** See the *Unwired Accelerator Administration Guide* for more information about setting the `page.group` property.

---

- 3 Click Add Application. When the Add Application window appears, it should display the contents of the new default display catalog.

---

**Note** When you import portal objects (templates, catalogs, and so on) they maintain their original owner; that is, the Owner column in the detail view for objects lists the person who originally created the object, not the person performing the import.

---

## Exporting objects from Mobile Web Studio

You can export catalogs, applications, page groups, pages, and agents to an archive file (with an XML format and extension) on the Mobile Web Studio's server. You can import that archive file and its contents into a different Mobile Web Studio installation.

You can also filter the entity you want to export so that only specific child objects of the entity are exported.

### ❖ Exporting portal objects to an XML file

- 1 Log in to the Mobile Web Studio as a user associated with a role—`StudioAdmin`, `PortalAdmin`—that has permission to perform all operations. See the *Unwired Accelerator Administration Guide* for more information about roles.
- 2 Select Build | Applications from the Mobile Web Studio left pane.
- 3 Click Import/Export on the Application Manager toolbar.
- 4 When the Export/Import window displays, click the Export tab.
- 5 On the Export tab, select the entity, or entities, you want to export by clicking in the check box to the left of the entity. Entities include:
  - Applications
  - Composite Applications

- Page Groups
- Pages
- Agents
- Catalog

You can export all or any combination of entities. When you select an entity, all other entities referred to by the selected entity are also exported. For example, if you select only page group, all pages, applications, and templates referred to by the page group are included. If you select all entities, the export includes each unique entity once. That is, if an application is referenced by both the catalog and five pages, the application appears only once in the export XML document.

---

**Note** In UA versions earlier than 6.0, only the active catalog could be exported. With UA 6.0 and later, more than one catalog can be exported at a time because agents can refer to catalogs that are inactive or not in an approved state.

---

When you click Select next to the entity you want to export, the Select Export window launches, which allows you to select specific objects on the entity to export.

- 6 This step is optional. Click Select if you do not want to export all the child objects of the entity.
- 7 If you did not filter the entity to export, go to the next step. If you clicked Select next to the entity to export, you see the Select Export window, where all available objects are added to the Selected list by default.  
Use the Add, Add All, Remove, and Remove All buttons to select specific objects for export, then click OK.
- 8 In the Export tab, in File Name, enter the name you want the XML file to have.
- 9 Click Export. The file is saved with the name you entered in the default Mobile Web Studio location that is specified in the *global.properties.xml* configuration file.
- 10 When a prompt advises you that the export operation succeeded, click OK.
- 11 Click View Log to review the export operation activities.

- 12 Click the Download button to the right of the File Name field to download the file from the Mobile Web Studio server to your local hard drive.
- 13 When the File Download dialog box displays, select Save.
- 14 When the Save As window appears, navigate to the location where you want to save the file, then enter a new file name—for example, `export.xml`—and click Save.
- 15 Click Close to exit the Mobile Web Studio–Export/Import window and return to the Application Manager.

Go to the location where the file was saved and send the file to another Mobile Web Studio installation via an e-mail message or import the file into a different Mobile Web Studio installation.

## Rules and restrictions

This section contains rules, restrictions, and additional information for importing and exporting specific portal objects.

### Importing and exporting page groups

Page groups are a portal entity. Page groups created by developers in Mobile Web Studio are called System Page Groups (SPG), as opposed to page groups created by Portal Interface users, which are personal page groups. SPGs represent default or guest page groups.

During the export process, all approved and active SPGs, including related entities, are written to the XML export document.

Like pages, SPGs are not version controlled. During the import process, existing page groups with the same name are replaced if the Replace Existing Entities option is selected in the Import tab of the Mobile Web Studio–Export/Import dialog box.

Pages within page groups are always exported regardless of their state (active or inactive). Even when a page is marked inactive within the Page Group Builder, it is still exported. Whether a page is imported as active or inactive depends on whether the Import Entities as Active option is selected on the Import tab. However, if a page is marked as inactive using the Page Group Builder, the page does not display in Portal Interface regardless of whether the Import Entities as Active option is selected.

There are two places in Mobile Web Studio where you can deactivate or activate a page:

- Page Group Builder – select Build | Page Groups, then right-click an Approved page group in the detail pane and select Edit. The Page Group Builder displays all pages in the page group and each page name is followed by an Active option. When the Active option is not selected, the page is deactivated. The page will not display in only this page group. Other page groups that include this page are not affected; that is, the active/inactive page state in the Page Group Builder is pertinent only to the selected page group.
- Page Properties – select Build | Pages, then right-click an approved page in the detail pane and select Properties. The Page Properties window displays an Active option. Deselect Active to deactivate the page.

---

**Note** The same relationship exists between pages and applications. Each page has a Active option for each application in the page. If Active is deselected for a specific application in the Page Manager, the application does not display on that page. If Active is deselect in the Application Manager, the selected application is deactivated and every page that contains the application does not display it.

---

## Exporting composite applications

A composite application typically consists of pages and page groups. When you choose to export composite applications, all active and approved composite applications, including related entities, are exported. Composite applications do not have versioning control. Existing composite applications with the same name are replaced if the Replace Existing Entities option is selected in the Import tab of the Mobile Web Studio–Export/Import dialog box.

## Exporting agents

The agent architecture contains multiple entities that may need to be exported when you export an agent—agents, agent servers, and adapters. An agent can also refer to pages, catalogs, and other agents. When you choose to export agents, all agent-related entities are exported.

## Roles and security integration

The roles within the XML export document are the union of all the roles used by all the exported entities (page groups, pages, catalogs, applications, templates, agents, and applications).

---

**Note** The portal administrator must create the required J2EE roles and verify that those roles are mapped to the underlying roles provided by EAServer and Enterprise Security or the Tomcat security realm. See the *Unwired Accelerator Administration Guide* for details.

---

At the time of import and export, Enterprise Security authenticates the Mobile Web Studio user to ensure that he or she has the appropriate role to perform the operation. The default role is PortalAdmin, which is specified by the value of the PortalAdministrationRole property in the *global.properties.xml* configuration file.

## Importing Mobile Web Studio objects protected by custom J2EE roles

When you import new Mobile Web Studio objects, new objects may be protected by newly created custom J2EE roles. To see the new objects, verify that the user is granted the PortalAdmin role that allows him or her to view those objects. Once the user is granted the appropriate role, when he or she logs in to Mobile Web Studio again, the new objects are visible and accessible.

Alternately, grant the user the PortalAdmin role. This allows him to add all Mobile Web Studio objects regardless of his current J2EE roles. Users with this configuration do not need to log out and log back in to view the newly imported objects.

## Restrictions

**Files used by applications or templates** You cannot export or import any files used by applications or templates. For example, a document type of application does not work in export/import because it involves installing document files from one deployment to another.

**XML validation** XML validation must be turned off for the import and export functionality to work:

- 1 Open *global.properties.xml* in a text editor. The default location of this file if you are using Tomcat is `%SYBASE%\tomcat\webapps\onepage\config`. If you are using EAServer, the location of this file is `%JAGUAR%\EAServer\Repository\WebApplication\onepage\config`.
- 2 Find `<Property name="XmlValidation">` and verify that the value is set to "off." If it is not, change it.
- 3 Save the file and close the text editor.
- 4 Restart the application server.

**Multiple element applications** To export an application to the Mobile Web Studio, which is used with M-Business Server to form the Unwired Accelerator product, the application can have only one element. Multiple-element applications are not compatible as mobile applications.

**Templates** The JSP template is the only valid template that can be used for exporting an application to M-Business Server.

## Exporting/importing parameters

Unwired Accelerator enables you to export parameters defined for an application as an XML document, and import them. This provides an alternative way to edit parameter definitions for an application, rather than through the usual user interface.

As well as including all the currently defined parameters for an application, the XML document also (for Web element applications) contains "potential" parameters – that is, CGI parameters that are present in HTTP requests but have not so far been selected for inclusion as exposed parameters. These are included in the document, but commented out using XML-style comments. It is then a simple task to expose these as external parameters, by un-commenting the ones that are required.

The same approach can be taken with @OP tags that have not yet been exposed as external parameters – the XML document includes them within XML-style comments as well.

❖ **Exporting application parameters**

- 1 In Mobile Web Studio, select Build | Applications | Approved.
- 2 Select an application in the right pane, and right click.
- 3 Select Export/Import Params from the pop-up menu.
- 4 On Export/Import Parameters, type the file name, including the XML extension.
- 5 Click Export, and OK to confirm. The XML file is saved in *%SYBASE%\tomcat\webapps\onepage*.

❖ **Importing application parameters**

- 1 In Mobile Web Studio, select Build | Applications | Approved.
- 2 Select an application in the right pane, and right click.
- 3 Select Export/Import Params from the pop-up menu.
- 4 On Export/Import Parameters, type the file name, including the XML extension.
- 5 Click Import, and OK to confirm.



# Changing the Portal's Look and Feel

This chapter describes the portal's primary configuration files and explains how to change the look and feel of a portal. Use these instructions for single portals or when you have multiple co-brands.

Topic	Page
Introduction	319
Editing styles.xml	320
Editing css.css	321
Editing messages.xml	322
Editing global.properties.xml	323

## Introduction

The most basic co-branding is accomplished by changing the logos, icons, colors, and visible text in Portal Interface. These attributes are specified in *styles.xml* and *messages.xml* that you copied to the new co-brand directory and in *css.css* that you copied to the new co-brand *style* subdirectory.

The settings you change in these resource-specific files override the settings in the onepage default files located in `%SYBASE%\tomcat\webapps\onpage\fw\baseApps` if you are using Tomcat. If you are using EAServer, the default files are located in `%JAGUAR%\EAServer\Repository\WebApplication\onpage\fw\baseApps`.

You can also make changes to the master configuration file *global.properties.xml*, located in *%SYBASE%\tomcat\webapps\onepage\config* if you are using Tomcat. If you are using EAServer, *global.properties.xml* is located in *%JAGUAR%\EAServer\Repository\WebApplication\onepage\config*.

---

**Note** Enter color values as hexadecimal code rather than color names; for example, “FFFFFF” is white. Use a Web-safe color chart for reference, such as the one located on the W3 Schools Web site at [http://www.w3schools.com/html/html\\_colors.asp](http://www.w3schools.com/html/html_colors.asp) or the Visibone Web site at <http://www.visibone.com/color/>.

---

When you replace Unwired Accelerator’s default images, icons, or tabs across all resources for an installation, put the new graphics in the appropriate subdirectory on *%SYBASE%\Tomcat\webapps\onepage\sy\_images*.

To replace the images only for a particular resource, create an *images* directory within the co-brand’s subdirectory and put a copy of the new files there. When you edit *styles.xml*, you reference the location from which you want that co-brand’s images pulled.

## Editing *styles.xml*

The default *styles.xml* file contains the settings for Portal Interface logos, icons, and colors. Table 13-1 lists the most commonly changed properties. Use any text editor to explore and edit the file.

This file is divided into three major groups, colors, images, and misc.

- Colors – stores font colors, background colors, table colors, and so on.
- Images – stores all the images used on the site. It has been subdivided into smaller groups to keep it more manageable. Keep images in the appropriate subgroups.
- Misc – stores content setting that do not fit into color or image.

**Table 13-1: *styles.xml* settings**

Setting	Description
bodyColorTopMenu	The body HTML of the Portal Interface menu at the top of the Portal Interface window.

Setting	Description
CorporateURL	URL link.
deleteIcon	Delete portlet icon.
expandIcon	Maximize portlet icon.
framesetAllIE	Frameset specification; for example, the default is 89,0,0,0.
iconAddPage	The Add Page icon at the top of the Portal Interface window.
iconHelp	The online Help icon at the top of the Portal Interface window.
iconLogout	The Logout icon at the top of the Portal Interface window.
iconManagePages	The Manage Pages icon at the top of the Portal Interface window.
iconManageTabs	Reserved for future use.
iconMyInfo	The My Info icon at the top of the Portal Interface window.
imgTopLogo, imgBottomLogo	The masthead image at the top and bottom of the Portal Interface window.
minimizeIcon	Minimize portlet icon.
mopTabCorner	Left unselected tab corner image.
nonSelTab	Unselected page group background color.
pageServBg, bodyColor	Page background.
portletHelpIcon, helpIcon	Portlet online help icon.
selTabBg	The selected page's background color.
tileBg	Portlet background color.
tileBorderColor	Portlet border color.
tileCorner	Portlet header corner image. The default is blue.
tileFont	Portlet font color.
tileHeadBg	Portlet header background color.
tileLock	Secure lock portlet icon.
topmenu_mainbg	The Portal Interface top menu background color.
topmenu_navrowbg	The Portal Interface navigation bar background color.
topmenu_tabrowbg	Portal Interface page group tab background color.

## Editing `css.css`

The `css.css` file is the cascading style sheet (CSS) used for the portal. Use any text editor to open `css.css` and change the settings you want to override. Save the file using the same file name "`css.css`." If you use a different file name, the portal does not recognize the file.

**Table 13-2: css.css settings**

Setting	Description
.active	Selected page group font style (font, size, color, and so on).
.nonactive	Unselected page group font style.
.topnavtext	User greeting font style.
.topnavsitelinks	Secondary navigation active links font style.
.inactivetopnavtext	Secondary navigation inactive links font style.
.toptabsetdropdown	Page group drop-down menu style.
.tileTitle	Portlet title font style.
.tileFunctions	Portlet header functions (refresh, edit).

## Editing messages.xml

The *messages.xml* file contains settings for Portal Interface text that displays in menus, messages, and so on. Table 13-3 lists additional message and text groups that you can customize. Use any text editor to explore and edit the file.

**Table 13-3: messages.xml settings**

Setting	Description
DSF_ABOUT_EP	Link message.
TM_HELP	Help message.

The setting names in an application element tag must be unique across all applications. To ensure that the names are unique, use the application name (short name) as a prefix. See Table 13-4 for reference.

**Table 13-4: message group short names**

Message group	Setting section
Days and months	DaysAndMonths short name/prefix=DATE
Page titles	PageTitles short name/prefix=PT
Portal login	LoginPage short name/prefix=LP
Top menu names	FWTopMenu short name/prefix=TM
Portlet edit	FWTileEdit short name/prefix=TE

Message group	Setting section
Portal registration	FWRegistration short name/prefix=RG
User's information	FWMyInfo short name/prefix=MI

## Editing *global.properties.xml*

The *global.properties.xml* file describes database table contents that must be modified when administering some features such as Default Tabs and Pages, and Default Tiles on the Default Tabs and Pages.

See the *Unwired Accelerator Administration Guide* for more information on viewing and modifying this file's contents.

---

**Note** When you make changes to *global.properties.xml*, you must restart the application server for your changes to take effect.

---



## Overview

This appendix provides troubleshooting information for various problems. See the *Unwired Accelerator Administration Guide* for additional troubleshooting information.

## SMS user not specified

**Description** User is not specified error.

**Solution** This can happen when the `roleBasedAccess` from `http://host:port/dejima/admin/home.jsp` is set to `false` and the `role.access` property in `uadejima.properties` is set to `true`. You must either set both of them to `false`, or both of them to `true`.

## Agent log fills up the database

**Description** The agents transaction logging fills up the database too fast and the View Log or other agent functionality seems slow. Check the database logs to see if the database is out of space. You may see an error like this:

```
Tasks are sleeping waiting for space to become
available in the log segment for database tempdb
```

**Solution** Increase the database size. If you are using Adaptive Server Anywhere with your UA installation, see the *ASA Database Administration Guide* for instructions. If you are using Adaptive Server Enterprise, see the *System Administration Guide*.

## Auto indexer does not work

**Description** The autoindexer does not work.

**Solution** Restart the autoindexer:

- 1 Go to the %SYBASE%\PortalSearch\Engine\main directory on Windows
- 2 Delete the *indexq.dat* file:
- 3 Restart the PortalSearchQueryh DRE service.

## Chart is not updating to show new data

If your chart is not updating to show new data, your IE browser may be caching the old chart image. To configure IE so that it is not pulling any data from its cache:

- 1 In Internet Explorer 6.0 go to Tools | Internet Options.
- 2 In the Internet Options window, in the General tab, click Settings.
- 3 In the Settings window, select Every Visit to the Page, and click OK.
- 4 Click OK to close the Internet Options window.

## Co-brands error

**Description** The following error displays in the *uwp.err* log:

```
ERROR Id='211' is NOT defined for OEM resources,  
defaulting to OEM rid 1
```

There are no other signs that the co-brand is set up incorrectly.

**Solution** The co-brand name is misspelled.

# .NET Container Client

This appendix describes how to use the sample .NET container client included with Unwired Accelerator. This client can be used on Windows C- based mobile devices.

Topic	Page
Overview	327
Installing the .NET container client	327
Starting and stopping the .NET client	335
Setting up .NET container client accounts	336
Customizing the .NET container client	337

## Overview

Unwired Accelerator includes client software for the mobile device that communicates with the Unwired Accelerator server. This enables you to download mobile applications to the mobile device, and to synchronize data between the Unwired Accelerator server and client.

## Installing the .NET container client

This section provides the information for setting up the 8.0 .NET client software on a Windows mobile PDA. The .NET client enables you to deploy and synchronize mobile applications between Unwired Accelerator and Windows-based mobile devices. This section includes:

- Requirements for using the .NET client software on a Windows device with Unwired Accelerator to deploy mobile applications to the device.

- Procedures for installing and setting up the Unwired Accelerator .NET client software on the mobile device.
- Procedures for setting up Unwired Accelerator .NET user profiles on a mobile device.

## Understanding .NET client requirements

These are the requirements for deploying mobile applications from the Unwired Accelerator server to a Windows Mobile PDA using .NET client software:

- Any of these operating systems (with, or without, telephone features):
  - Windows Mobile 2003
  - Windows Mobile 2003 Second Edition
  - Windows Mobile 5.0

---

**Note** Windows Smartphone devices (like the Motorola MPx220) are not supported. If you are unsure of your device's operating system, see "Determining the device's operating system" on page 329.

---

- Approximately 20 MB of free storage space to install the following components (installers for these components are included with the Unwired Accelerator .NET client).
  - .NET Compact Framework 1.0 Service Pack 3 – this must be installed to the device's memory; it cannot be stored to a memory card. The Service Pack is not included in the release, but can be downloaded from the Microsoft Web site at <http://www.microsoft.com>.

---

**Note** If you are unsure of your device's service pack, see "Determining the device's operating system" on page 329.

---

- Adaptive Server Anywhere for Windows CE (ASA for CE) database – ASA for CE is included in the release. this can be installed to a removable memory card, if your device is low on storage space and supports one.

- Unwired Accelerator 8.0 .NET client – this should be installed to the device memory for optimal performance. The .NET client can be installed to a removable memory card, if your device is low on storage space and supports one, but it is not recommended because frequent accesses to the database file degrade performance noticeably. The .NET client is included in the release.
- An Unwired Accelerator 8.0 server installation with which to connect (described in the *Unwired Accelerator Installation Guide*).
- Network connection that allows the PDA to communicate with the Unwired Accelerator server. Possible network connections include:
  - ActiveSync – share your laptop/desktop network connection by connecting the PDA to the PC using ActiveSync. You can download ActiveSync from the Microsoft Website.
  - 802.11 wireless network – requires that your device has a wireless network antenna.
  - GPRS connection – requires that your device access the Internet via GPRS, and that your Unwired Accelerator server is publicly accessible on the Internet.

## Determining the device's operating system

You can determine the operating system of your device by checking the About window.

### ❖ Checking the About window

- 1 On the mobile device, click the Start menu and select Settings.
- 2 Click the System tab.
- 3 Click the About icon.
- 4 Click the Version tab if it is not already selected.
- 5 Check the version information (the Microsoft flag logo in the upper left corner of the Start menu indicates the version visually):
  - Windows Mobile 2003 – the text states Microsoft Pocket PC and Version 4.20 (the Microsoft flag logo is outlined in white).
  - Windows Mobile 2003 Second Edition – the text states Windows Mobile 2003 Second Edition and Version 4.21 (the Microsoft flag logo is not outlined in white).

- Windows Mobile 5.0 – the text states Microsoft Windows Mobile Version 5.0 (the Microsoft flag logo is not outlined in white, and the OK button in the upper right corner is square in shape).

## Installing the .NET software

This section provides information for helping .NET client mobile device users set up to use Unwired Accelerator. Use Mobile Web Studio to set up an account for each user, and supply the following information to the user:

- Location of downloads – each user must install the .NET Compact Framework, ASA for CE, and Unwired Accelerator .NET client on his or her mobile device. Develop a process for making the version and location information available to users.
- User Name – the user name for the profile, such as `masuper`.
- Password – the password for the profile, such as `m8super`.
- Server – the server and domain on which the Unwired Accelerator server is running, such as `labxp.sybase.com`.
- Port – the port used to access the Unwired Accelerator server, such as `4040`.
- RID – the default resource identifier (RID) for the account, such as `21` for Unwired Accelerator.

Each user must install the following software on a Windows mobile PDA:

- .NET Compact Framework
- ASA for Windows CE
- Unwired Accelerator 8.0 .NET client

## Installing the .NET Compact Framework

Install the .NET Compact Framework 1.0 Service Pack 3 on your device. You may want check to see whether the required .NET Compact Framework is already installed. If it is not installed, download the Service Pack from the Microsoft Web site and install it.

### ❖ Checking the .NET Compact Framework version

In this step, verify the Service Pack installed on your mobile device. Skip this procedure if you already know you need to install .NET Compact Framework 1.0 SP3 on your device.

- 1 Open the File Explorer application on the device.
- 2 Browse to the `\Windows` folder.
- 3 Locate and click once on the file `cgacutil`. If there is no `cgacutil` file, the .NET Compact Framework is not installed.

A pop-up window displays the version number of the installed .NET Compact Framework. Use the information below to determine which version or versions of the .NET Compact Framework is installed.

**Table 13-5: .NET Compact Framework versions**

Version number	Software version
1.0.2268.0	.NET Compact Framework 1.0 RTM
1.0.3111.0	.NET Compact Framework 1.0 SP1
1.0.3226.0	.NET Compact Framework 1.0 SP2 Recall
1.0.3227.0	.NET Compact Framework 1.0 SP2 Beta
1.0.3316.0	.NET Compact Framework 1.0 SP2 Final
1.0.4177.0	.NET Compact Framework 1.0 SP3 Beta
1.0.4292.0	.NET Compact Framework 1.0 SP3
2.0.4037.0	.NET Compact Framework 2.0 (part of VS2005 CTP May)
2.0.4135.0	.NET Compact Framework 2.0 (part of VS2005 Beta 1)
2.0.4317.0	.NET Compact Framework 2.0 (part of VS2005 CTP November)
2.0.4278.0	.NET Compact Framework 2.0 (part of VS2005 CTP December)
2.0.5056.0	.NET Compact Framework 2.0 (BETA 2)
2.0.5238.0	.NET Compact Framework 2.0 Release

- 4 If you do not see version 1.0.4292.0 or higher, install .NET Compact Framework Service Pack 3.

---

**Note** You may see multiple version numbers in the pop-up window, meaning that multiple CFs are installed, such as CF 1 and CF 2. This is acceptable, as long as 1.0.4292.0, the version number for .NET Compact Framework 1 Service Pack 3, is included in the list of installed versions.

---

❖ **Downloading .NET Compact Framework Service Pack 3**

In this step, download the Service Pack from the Microsoft download Web site to your PC. Skip this procedure if .NET Compact Framework SP3 is installed on your mobile device.

- 1 Navigate to:

```
http://www.microsoft.com/downloads/details.aspx  
?FamilyId=A5A02311-194B-4C00-B445-F92BEC03032F&  
displaylang=en
```

- 2 Review the download instructions on the Microsoft Web site.
- 3 Download the Service Pack to your PC, according to the instructions.

❖ **Installing .NET Compact Framework Service Pack 3**

In this step, download the Service Pack from your PC to your mobile device, and install the Service Pack on the mobile device. Skip this procedure if .NET Compact Framework SP3 is installed on your mobile device.

- 1 Use ActiveSync to connect your mobile device to your PC.
- 2 On your PC, with your mobile device connected, run the Service Pack installer. The installer detects your device operating system, copies the appropriate configuration file to your device, and installs it.

- If the device already has this version of the .NET Compact Framework installed, this prompt displays:

```
Microsoft .NET Compact Framework is already  
installed. Reinstall?
```

Click Cancel and proceed to “Installing ASA for Windows CE” on page 333 to install the database. Or, click OK to reinstall.

- If you continue with the installation, and if a memory card is plugged into your mobile device, you may be prompted to select the installation destination for the .NET Compact Framework.

---

**Note** Select the device memory as the installation destination, not the memory card.

---

## Installing ASA for Windows CE

Install the Adaptive Server Anywhere for Windows CE (ASA for CE) database on your mobile device. The database is used for persistent storage of system and mobile application data.

The CABinet (CAB) file for ASA for CE is located in:

```
SYBASE\UA80\tomcat\webapps\onepage\ota\wm
```

If you cannot access the Unwired Accelerator file system, or if you are WiFi-enabled, you can obtain the CAB file at:

```
http://hostname.domain:port/onepage/ota/wm/
```

---

**Note** To install a CAB file into any folder other than the default destination, use a CAB file installer, such as the freeware tool CabInstl, that allows you to browse for a CAB file, then select where to install it.

CabInstl is available from <http://s-k-tools.com/util.html> (or go directly to <http://s-k-tools.com/freeware/cabinl.zip> to obtain the downloadable zip file). The zip file includes three versions of the tool—English, German, and Russian. Other similar tools are available on the Internet.

If you are using CabInstl, manually create the folders to which you want to extract the CAB file content, before you unpackage the CAB file.

---

### ❖ Installing ASA for CE

In this step, install ASA for CE on your mobile device. Since the CAB file is quite large (about 9MB), you may want to use ActiveSync to copy the file from Unwired Accelerator to the mobile device. Skip this procedure if ASA for CE is already installed on your device.

- 1 Use ActiveSync to connect your mobile device to your PC.
- 2 Use File Explorer to navigate to:  
`SYBASE\tomcat\webapps\onepage\ota\wm`
- 3 Explore the mobile device and copy the `asa_ce.ARM.30.CAB` file to the device.
- 4 Use File Explorer on the device to browse to where you placed the `asa_ce.ARM.30.CAB` file.
- 5 Tap on `asa_ce.ARM.30.CAB` once to start the installation.

If you have a memory card plugged into the device, you may be prompted to install the database to the device or the memory card. Optimally, select the device memory as the installation destination for faster runtime performance.

However, if your device is running low on storage space, you can select the memory card.

---

**Note** On Windows Mobile 2003 devices, create the destination folder on the memory card, `\Storage Card\Program Files\Sybase\ASA9`, then use the CAB installer to extract the files into this folder.

On Windows Mobile 5, the operating system is designed to create the destination folder automatically.

---

## Installing the Unwired Accelerator .NET client

In this step, install the Unwired Accelerator .NET client on your mobile device. The client is used to synchronize with the Unwired Accelerator server.

### ❖ Installing the .NET client

- 1 Use ActiveSync to connect your mobile device to your PC.
- 2 Use File Explorer to navigate to:  
`SYBASE\tomcat\webapps\onepage\ota\wm`
- 3 Copy the `UnwiredAcceleratorClient_PPC.ARMV4.CAB` file to the mobile device.
- 4 Use File Explorer on the device to browse to where you placed the `UnwiredAcceleratorClient_PPC.ARMV4.CAB` file.
- 5 Tap on `UnwiredAcceleratorClient_PPC.ARMV4.CAB` once to start the installation.

If you have a memory card plugged into the device, you may be prompted to install the database to the device or the memory card. Sybase recommends that you install to device memory for optimal runtime performance.

---

**Note** If you do install to memory card, performance will be very slow, since the .NET client frequently accesses the ASA database, and file access on the memory card is slower than file access on the main device memory.

To install to a memory card on Windows Mobile 2003 devices, create the destination folder on the memory card, *\Storage Card\Program Files\Sybase\ASA9*, and then use the CAB installer to extract the files into this folder.

On Windows Mobile 5, the operating system is designed to create the destination folder automatically.

---

## Starting and stopping the .NET client

This section describes how to start the .NET client and the ASA for CE database server on the mobile device.

### ❖ Starting the .NET client application

- 1 On your mobile device, open the Start menu and select Programs.
- 2 Locate the Unwired Accelerator icon, and tap it once. The ASA for CE database is loaded, and then the Unwired Accelerator 8.0 .NET client runs. You need to select a user profile to work with mobile applications.

### ❖ Stopping the .NET client application

- 1 On the applications window, click the Menu.
- 2 Select Exit. The .NET client shuts down, followed by the ASA for CE database server.

## Setting up .NET container client accounts

This section describes how to set up .NET container client user profiles. Topics include:

- Setting up the .NET user
- Editing the .NET user
- Deleting the .NET user

### ❖ **Setting up a .NET user profile**

The user account must correspond to a profile on the Unwired Accelerator 8.0 server. Either set up the user account in advance through Mobile Web Studio as a system administrator, or have the user log in to Portal Interface and use Join Now to set up an account. Once the account is set up, the user can set up a .NET user profile from the mobile device.

- 1 Click the Menu in the lower left corner of the screen.
- 2 Click Profiles to view the Profiles Manager screen. Initially no profiles are listed on the screen.
- 3 Click Menu and select New Profile to open the Profile Editor screen.
- 4 Enter the details for the Unwired Accelerator 8.0 server to which you will be connecting.
  - Profile Name – the profile name for the account, such as `myUAserver`.
  - User Name – the user name for the profile, such as `masuper`.
  - Password – the password for the profile, such as `m8super`.
  - Server – the server and domain on which the Unwired Accelerator server is running, such as `labxp.sybase.com`.
  - Port – the port used to access the Unwired Accelerator server, such as `4040`.
  - RID – the default resource identifier (RID) for the account, such as `21` for Unwired Accelerator.
- 5 Click the Menu and select Save. The Profile Manager screen displays. The new profile is listed on the screen.
- 6 Click on the Menu and select Close to return to the main screen.

### ❖ **Editing a .NET user profile**

- 1 Select the profile to edit. You can either:

- Click once on the profile to edit, then select Menu | Edit Profile. The Profile Editor screen appears.
  - Tap and hold the profile to edit, then select Edit Profile from the context menu. The Profile Editor screen appears.
- 2 In Profile Editor, edit the profile values for the account.
  - 3 Select Menu | Save to save the changes, or select Menu | Close to cancel the changes, and return to the main screen.
- ❖ **Setting up a .NET user profile**
- 1 Select the profile to delete. You can either:
    - Click once on the profile to delete, then select Menu | Delete Profile. You are prompted to confirm the deletion request.
    - Tap and hold the profile to delete, then select Delete Profile from the context menu. You are prompted to confirm the deletion request.
  - 2 Select Yes to confirm.

## Customizing the .NET container client

This section describes how to customize the .NET container client on your mobile device without recompiling software. You can customize:

- Language
- Message text
- Interface strings
- Graphics
- Icons

Perform customization on the mobile device after you deploy the .NET client to the mobile device. Each user can customize the .NET client to individual preference by modifying default files. Alternatively, acting as system administrator, you can customize the default files for your users, who in turn can download the files to mobile devices, and further modify the files.

This section assumes the .NET client configuration is in place. If not, see “Installing the .NET container client” on page 327 for installation information and procedures.

## Customization overview

This section provides information and procedures for customizing the .NET client. Modify the *Strings.xml* file to customize the .NET client.

---

**Note** Typically, modifications to the *Strings.xml* file are performed by each user on the mobile device. Alternatively, you can copy the *Strings.xml* file to a PC and use a UTF-8-capable text editor to edit the strings. Then, copy the *Strings.xml* file back to the PDA, and restart the .NET client to see the changes on the mobile device.

You may need to develop procedures for deploying updated *Strings.xml* files to mobile device users. You may want to consider using a remote device management tool, such as Afaria, to place new images and files on PDAs.

---

## Customizing language

To customize language for the .NET client:

- Obtain the proper language files and related font files, and download the files to your mobile device.
- Obtain or create localized user interface graphics, logos, and icons, using the sizes recommended for your device. Download the files to your mobile device.
- Modify the *Strings.xml* file on your mobile device:
  - Specify the appropriate language-related resource values, such as language, font, and text orientation.
  - Translate message strings to the language, such as error and validation messages.
  - Translate user interface strings to the language, especially user profile fields and prompts.
  - Update the user interface graphics, logos, and icons file names.

See “Understanding the Strings.xml file” on page 341 for information before you start.

When you restart the application, the .NET client:

- 1 Checks the device’s regional settings for the current language code and country code. For example, a PDA set to “Dutch (Belgium)” would be “nl-BE.”

- 2 Checks for the string localization file that matches the language and country code; in this case, *Strings.nl-BE.xml* for Dutch Belgium.

If the file does not exist, the client looks for an alternative file that contains the language code, but no country code (for example, *Strings.nl.xml*, a generic Dutch localization file).

If that is not found, the client defaults to the *Strings.xml* file with default English strings.

#### ❖ Localizing the .NET client

This procedure assumes you have changed the regional setting, downloaded any necessary fonts or character sets, and moved localized graphics files to the mobile device.

- 1 Navigate to the *Strings.xml* file.
- 2 Make a copy of the *Strings.xml* file, and include the language and country code in the new file name. For example:

*Strings.nl-BE.xml* for Dutch Belgium

*Strings.fr-CA.xml* for French Canadian

*Strings.fr-FR.xml* for French

- 3 Modify the new file on your mobile device:
  - Specify the appropriate language-related resource values, such as language, font, and text orientation. See Table B-1 on page 343 for information about the resource values.
  - Translate message strings to the language, such as error and validation messages. See “Messages in Strings.xml” on page 344 for default messages.
  - Translate user interface strings to the language, especially user profile fields and prompts. See “Strings in Strings.xml” on page 346 for default strings.
  - Update the user interface graphics, logos, and icons file names. See “Images in Strings.xml” on page 348 for information about graphics.

See “Understanding the Strings.xml file” on page 341 for information about working with the *Strings.xml file* (or the localized version of the file).

- 4 Save the file and close it.

- 5 You can also localize the template files, as described in “Understanding the device template files” on page 349. The template files include some language strings.
- 6 Restart the application. The .NET client should detect the regional language and country codes, locate the localized *Strings.xml* file, and present the localized interface.

---

**Note** Any reference to the *Strings.xml* file extends to your localized version of the file, such as *Strings.nl-BE.xml*.

---

## Customizing messages

The *Strings.xml* file includes a section of messages that can be localized to a different language, or that can be customized to use terms and phrasing more meaningful for a business setting, or for an individual user. See “Messages in Strings.xml” on page 344 for message text.

## Customizing graphics

The *Strings.xml* file includes a section of graphics. You can obtain or create localized graphics, and replace the defaults included with the .NET client. After the .NET client is installed, replace the images on the device with new images, and update the graphic file names in *Strings.xml*. See “Images in Strings.xml” on page 348 for graphics file names.

## Customizing the user interface

The *Strings.xml* file includes sections for changing Unwired Accelerator strings and icons that appear on the mobile device user interface.

- For user interface strings, such as “Menu,” customize the default text that comes with the .NET client. See “Strings in Strings.xml” on page 346 for string text.

---

**Note** Additional user interface strings appear in the template files. You can customize or localize these strings. See “Understanding the device template files” on page 349 for information.

---

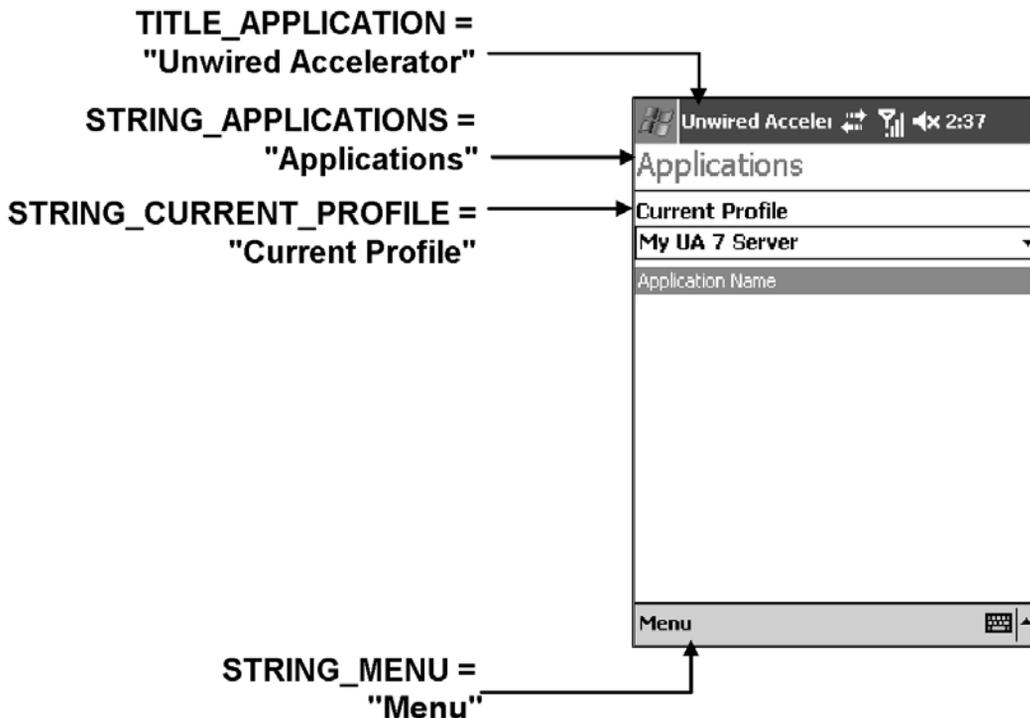
- For icons, such as buttons, you can obtain or create custom graphics, and replace the defaults included with the .NET client. After the .NET client is installed, replace the default icons on the device with new icons, and update the icon file names in *Strings.xml*. See “Images in Strings.xml” on page 348 for icon file names.

## Understanding the *Strings.xml* file

The *Strings.xml* file is used to control what is printed on the device, including user interface labels, messages, logos, icons, and other graphics. This is the primary file used to customize the Unwired Accelerator .NET client. Resource names and values in the file equate to text or visual cues displayed on the screen.

Figure B-1 on page 342 shows several resource names and values, and the equivalent PDA screen representation.

Figure B-1: Resource names and PDA screen equivalents



Following is a partial, sample *Strings.xml* file, with key resource names highlighted. See Table B-1 on page 343 for descriptions of these resource names.

```

<?xml version="1.0" encoding="utf-8" ?>
<root>
<Resource Name="TITLE_APPLICATION" Value="Unwired Accelerator" />
<Resource Name="FORMAT_DISPLAY_TEXT_RTL" Value="false" />
<Resource Name="FORMAT_FONT" Value="Arial" />
<Resource Name="FORMAT_LAST_DOWNLOAD_DATE_FORMAT" Value="yyyy-MM-dd hh:mm:ss" />
<Resource Name="FORMAT_LIST_VIEW_ROW_HEIGHT_FOR_DPI_96" Value="16" />
<Resource Name="FORMAT_LIST_VIEW_ROW_HEIGHT_FOR_DPI_192" Value="32" />
...
<Resource Name="MESSAGE_ASA_NOT_FOUND" Value="Adaptive Server Anywhere for Windows CE could not be found. This application will now exit." />
<Resource Name="MESSAGE_CAN_NOT_ACCESS_APPLICATION_DATA" Value="Unable to access Application data." />
<Resource Name="MESSAGE_CAN_NOT_CREATE_NEW_APPLICATION" Value="Can not create

```

```

new Application." />
<Resource Name="MESSAGE_CAN_NOT_READ_PROFILE" Value="Can not read Profile." />
...
<Resource Name="STRING_ABORT" Value="Abort" />
<Resource Name="STRING_*ALL_COLUMNS*" Value="ALL COLUMNS" />
<Resource Name="STRING_APPLICATIONS" Value="Applications" />
...
<Resource Name="IMAGE_ADD_MOUSEUP_24" Value="add_white_24.bmp" />
<Resource Name="IMAGE_ADD_MOUSEDOWN_24" Value="add_black_24.bmp" />
<Resource Name="IMAGE_ADD_DISABLED_24" Value="add_disabled_24.bmp" />
...
</Root>

```

**Table B-1: Resource names in Strings.xml**

Resource name	Description	Default value
encoding	Character encoding for Unicode. The default, UTF-8, is ideal for mobile device applications.	utf-8
TITLE_APPLICATION	The default application name for Unwired Accelerator that displays on the mobile device.	Unwired Accelerator
FORMAT_DISPLAY_TEXT_RTL	Indicates whether application text displays right to left on the mobile device. False indicates text displays left to right; true indicates text displays right to left.	false
FORMAT_FONT	The default application font used on the mobile device.	Arial
FORMAT_LAST_DOWNLOAD_DATE_FORMAT	The date-time format used on the mobile device.	yyyy-MM-dd hh:mm:ss
FORMAT_LIST_VIEW_ROW_HEIGHT_FOR_DPI_96	The row height in pixels for 96 dots-per-inch (DPI).	16
FORMAT_LIST_VIEW_ROW_HEIGHT_FOR_DPI_192	The row height in pixels for 192 DPI.	32
MESSAGE_message_text	Message text, such as “Can not read Profile.” You can modify or localize each message.	See Table B-2 on page 344 for additional information about the messages.
STRING_string_text	Application labels, such as “Profile Name.” You can modify or localize each message. See Figure B-1 on page 342 for several string examples.	See Table B-3 on page 346 for additional information about the user interface strings.

Resource name	Description	Default value
IMAGE_image_name	Application images, such as buttons, graphics, and logos. Many images are provided in normal, selected, or unavailable style, and multiple sizes of images are available, depending on the mobile device window dimensions.	See Table B-4 on page 348 for additional information about images.

## Messages in Strings.xml

Table B-2 provides the message resources and their default values. You can change the default values, either by localizing the text or modifying it for your environment.

Some messages include variables, or placeholders, that are populated depending on the circumstances. For example, the message `Delete {0}?` includes one variable ( `{0}` ). During execution, the variable is replaced with another message or number, for example, the message might read `Delete Stock?`. Do not change the placeholder, just the text

**Table B-2: Messages in Strings.xml**

Message resource	Default value
MESSAGE_ASA_NOT_FOUND	Adaptive Server Anywhere for Windows CE could not be found. This application will now exit.
MESSAGE_CAN_NOT_ACCESS_APPLICATION_DATA	Unable to access Application data.
MESSAGE_CAN_NOT_CREATE_NEW_APPLICATION	Can not create new Application.
MESSAGE_CAN_NOT_READ_PROFILE	Can not read Profile.
MESSAGE_DELETE_ALL_APPLICATIONS?	Delete all Applications?
MESSAGE_DELETE_APPLICATION_X?	Delete {0}?
MESSAGE_DELETE_ALL_LOGS?	Delete all Logs?
MESSAGE_DELETE_LOG?	Delete the selected Log?
MESSAGE_DELETE_PROFILE_X?	Delete {0}? This will also delete all Applications related to this profile.
MESSAGE_ERROR_GETTING_SCREEN_ORIENTATION_X	Error getting screen orientation: {0}
MESSAGE_ERROR_SETTING_SCREEN_ORIENTATION_X	Error setting screen orientation: {0}
MESSAGE_HTTP_ERROR_X	HTTP Error: {0}
MESSAGE_INVALID_HTTP_RESPONSE	Invalid HTTP response.

Message resource	Default value
MESSAGE_INVALID_LINKED_UA_APPLICATION	The Unwired Accelerator linked application is invalid. Please notify your system administrator about this message and include the name of the application you are trying to synchronize.
MESSAGE_INVALID_PROFILE_KEY	Invalid Profile Key.
MESSAGE_INVALID_PROFILE_NAME	The profile name is empty.
MESSAGE_INVALID_PROFILE_PASSWORD	The profile password is empty.
MESSAGE_INVALID_PROFILE_PORT	The profile port number is invalid.
MESSAGE_INVALID_PROFILE_RID	The profile RID number is invalid.
MESSAGE_INVALID_PROFILE_SERVER	The profile server is empty.
MESSAGE_INVALID_PROFILE_USERNAME	The profile username is empty.
MESSAGE_INVALID_UA_VERSION_CANNOT_SYNC_APP_X	The Unwired Accelerator server you are trying to synchronize with is not a version 8.0 server. The application {0} can not be synchronized.
MESSAGE_INVALID_VALUE_FOR_X	Invalid value for '{0}'.
MESSAGE_NETCF_1.0SP3_NOT_FOUND	This application requires that the .NET Compact Framework 1.0 Service Pack 3 be installed. This application will now exit.
MESSAGE_NO_RESX_STRING_FOR_X	No resource found for {0}.
MESSAGE_NO_APPLICATIONS_AVAILABLE	No Applications are available for download.
MESSAGE_PROFILE_NAME_NOT_UNIQUE	Another Profile already has this name. Please enter a unique name for this Profile.
MESSAGE_UA_AUTHENTICATION_FAILED	Unable to authenticate your username and password with the Unwired Accelerator server.
MESSAGE_UA_CONNECTION_FAILED	Unable to connect to the Unwired Accelerator server.
MESSAGE_UNABLE_TO_CONNECT_WITH_UA_SERVER	Unable to connect with the Unwired Accelerator server. Please check and verify your network access and coverage.
MESSAGE_UNABLE_TO_CREATE_LINKED_APPLICATION	Unable to create the linked Application. Please notify your system administrator about this message and include the name of the application you are trying to synchronize.
MESSAGE_UNABLE_TO_FIND_TEMPLATE_X	Unable to find the template named: {0}
MESSAGE_UNKNOWN_UA_SERVER_ERROR	There was an unknown error on the Unwired Accelerator server. Please notify your system administrator.

## Strings in *Strings.xml*

Table B-3 provides the user interface string names found in *Strings.xml*. These are the labels, menu options, and so forth that appear on the application user interface. You can change the default values, either by localizing the text or modifying it for your environment.

Some messages include variables, or placeholders, that are populated depending on the circumstances. For example, the message `Saving Record {0} of {1}` includes two variables ( `{0}` and `{1}`). During execution, the variables are replaced with other messages, for example, the message might read `Saving Record 20 of 25`. Do not change the placeholder, just the text.

**Table B-3: User interface string names in *Strings.xml***

User interface string name	Default
STRING_ABORT	Abort
STRING_*ALL_COLUMNS*	ALL COLUMNS
STRING_APPLICATIONS	Applications
STRING_BACK	Back
STRING_BUILD_DATE:_X	Build Date: {0}
STRING_CANCEL	Cancel
STRING_CANCELLED	Cancelled
STRING_CONFIRM	Confirm
STRING_CURRENT_PROFILE	Current Profile
STRING_DELETE	Delete
STRING_DELETE_ALL	Delete All
STRING_DELETE_APPLICATION	Delete Application
STRING_DELETE_LOG	Delete Log
STRING_DELETE_PROFILE	Delete Profile
STRING_DELETING_LOGS	Deleting Logs
STRING_DESELECT_ALL	Deselect All
STRING_DETAIL	Detail
STRING_DETAIL_VIEW	Detail View
STRING_DOWNLOADING	Downloading
STRING_EDIT	Edit
STRING_EDIT_PROFILE	Edit Profile
STRING_EQUALS	Equals
STRING_ERROR	Error
STRING_EXIT	Exit
STRING_FAILED	Failed

User interface string name	Default
STRING_IGNORE	Ignore
STRING_INSERT	Insert
STRING_INVERT_SELECTION	Invert Selection
STRING_LIKE	Like
STRING_LOG_DETAILS	Log Details
STRING_LOG	Log
STRING_LOGS	Logs
STRING_MENU	Menu
STRING_MESSAGE	Message
STRING_NEW	New
STRING_NEW_PROFILE	New Profile
STRING_NO	No
STRING_OK	OK
STRING_OPEN	Open
STRING_OVERALL_PROGRESS	Overall Progress
STRING_PASSWORD:	Password :
STRING_PORT:	Port :
STRING_PROFILE	Profile
STRING_PROFILE_NAME:	Profile Name :
STRING_PROFILES	Profiles
STRING_PROGRESS	Progress
STRING_REQUIREMENT_MISSING	Requirement Missing
STRING_RETRY	Retry
STRING_RID:	RID :
STRING_SAVE	Save
STRING_SAVED	Saved
STRING_SAVING	Saving
STRING_SAVING_RECORD_X_OF_Y	Saving Record {0} of {1}
STRING_SEARCH	Search
STRING_SEARCH_FOR:	Search For :
STRING_SEARCH_IN:	Search In :
STRING_SEARCH_TYPE:	Search Type :
STRING_SELECT_ALL	Select All
STRING_SENDING_UPDATES	Sending Updates
STRING_SERVER:	Server :
STRING_SHOW_ALL	Show All
STRING_SORT	Sort

User interface string name	Default
STRING_STATUS	Status
STRING_STATUS:	Status:
STRING_SYNCHRONIZE_ALL	All
STRING_SYNCHRONIZE_SELECTED	Selected
STRING_SYNCHRONIZE	Synchronize
STRING_SYNCHRONIZE_UPDATES	Sync Updates
STRING_SYNCHRONIZATION_CANCELLED	Synchronization Cancelled
STRING_SYNCHRONIZING	Synchronizing
STRING_TIMESTAMP	Timestamp
STRING_TYPE	Type
STRING_USER_NAME:	User Name :
STRING_VALIDATION_FAILED	Validation Failed
STRING_VERSION:_X	Version: {0}
STRING_X/Y	{0}/{1}
STRING_YES	Yes

## Images in *Strings.xml*

Table B-4 provides image file names found in the *Strings.xml* file. You can change these images. To change an image, you actually need six images; for example:

- Each button has three states (mouse up, mouse down, and disabled). The default buttons use a black background when selected, and a gray background when disabled, as a cue to the user.
- Each button must be set to the proper size for both high (48x48 pixels) and low (24x24 pixels) resolution screens.

Files with the BMP extension work best.

**Table B-4: Image file names in *Strings.xml***

Image file name	Default
IMAGE_ADD_MOUSEUP_dpi_value	add_white_xx.bmp
IMAGE_ADD_MOUSEDOWN_dpi_value	add_black_xx.bmp
IMAGE_ADD_DISABLED_dpi_value	add_disabled_xx.bmp
IMAGE_BACK_MOUSEUP_dpi_value	back_white_xx.bmp
IMAGE_BACK_MOUSEDOWN_dpi_value	back_black_xx.bmp
IMAGE_BACK_DISABLED_dpi_value	back_disabled_xx.bmp

Image file name	Default
<i>IMAGE_DELETE_MOUSEUP_dpi_value</i>	delete_white_xx.bmp
<i>IMAGE_DELETE_MOUSEDOWN_dpi_value</i>	delete_black_xx.bmp
<i>IMAGE_DELETE_DISABLED_dpi_value</i>	delete_disabled_xx.bmp
<i>IMAGE_EDIT_MOUSEUP_dpi_value</i>	edit_white_xx.bmp
<i>IMAGE_EDIT_MOUSEDOWN_dpi_value</i>	edit_black_xx.bmp
<i>IMAGE_EDIT_DISABLED_dpi_value</i>	edit_disabled_xx.bmp
<i>IMAGE_OPEN_MOUSEUP_dpi_value</i>	folder_white_xx.bmp
<i>IMAGE_OPEN_MOUSEDOWN_dpi_value</i>	folder_black_xx.bmp
<i>IMAGE_OPEN_DISABLED_dpi_value</i>	folder_disabled_xx.bmp
<i>IMAGE_SAVE_MOUSEUP_dpi_value</i>	ave_white_xx.bmp
<i>IMAGE_SAVE_MOUSEDOWN_dpi_value</i>	save_black_xx.bmp
<i>IMAGE_SAVE_DISABLED_dpi_value</i>	save_disabled_xx.bmp
<i>IMAGE_SYNC_MOUSEUP_dpi_value</i>	sync_white_xx.bmp
<i>IMAGE_SYNC_MOUSEDOWN_dpi_value</i>	sync_black_xx.bmp
<i>IMAGE_SYNC_DISABLED_dpi_value</i>	sync_disabled_xx.bmp

## Understanding the device template files

The template files are used to define visual aspects of the mobile device screen, for example, the color of the heading row, and the color of a row when it is selected and when it is not. Template files contain a few strings that can be customized or localized, and settings for font type and size, and encoding.

Template files are provided for mobile device dimensions, orientation (portrait and landscape), and resolution. When you modify template files for a particular screen size and configuration, modify similar values in all files.

When modifying the template files, keep in mind the following guidelines:

- You can edit any strings that are English language strings, such as Application [Last Downloaded] in this property:

```
<Property Name="CellSource" Value="&quot;Application [Last Downloaded]&quot;" />
```

Note that the `&quot;` XML entities identify this as a human language value, which can be translated and localized. Be sure to leave the XML entities intact [including the ampersand (&) and semicolon (;)], otherwise the XML document will be invalid, causing the XML parser to fail, and making the .NET client unable to load the templates.

- Do not edit any variables, such as `APPLICATION_NAME` in this property:  

```
<Property Name="CellSource" Value="APPLICATION_NAME" />
```

`APPLICATION_NAME` is a special variable flag that is replaced by the name of the application. If you edit this value, no application name will appear, even for other languages.

Template values like `APPLICATION_NAME` that are not surrounded by the `&quot;`; XML entity should not be edited. They serve as variable flags that are replaced by the client application when there is data to display on the screen.

- Do not modify any coordinate values, such as:  

```
<Property Name="Location" Value="0,28" />
```
- Templates should not be edited on the device. Instead, they should be transferred to the PC (which can be done using the ActiveSync File Explorer), and edited on a text editor that supports UTF-8. After editing the files, move them back to the device, and restart the application to see the changes.

As with the *Strings.xml* file, you may need to develop procedures for deploying updated template files to mobile device users. You may want to consider using a remote device management tool, such as Afaria, to place new images and files on PDAs.

- For localization, you can edit the default template files to something other than English, or you can add multiple language support by creating copies of the templates using the same naming technique used for the *Strings.xml* file (described in “Customizing language” on page 338). For example, with a 240x320 device set to Belgian Dutch, the client would look for templates in this order:

- a Template.Applications.240x320.nl-BE.XML
- b Template.Applications.240x320.nl.XML
- c Template.Applications.240x320.XML

❖ **Customizing the device template files**

- 1 Identify the dimensions of your mobile device screen, for example, 240 x 320 pixels.
- 2 Identify the templates available for the screen dimensions, for example:
  - Template.Applications.240x320

- `Template.ApplicationDataDetailView.240x320`
  - `Template.ApplicationDataListView.240x320`
  - `Template.ApplicationsSyncList.240x320`
  - `Template.DataEdit.240x320`
  - `Template.InsertDataForm.240x320`
  - `Template.InsertDataGrid.240x320`
  - `Template.LogsListView.240x320`
  - `Template.LogsDetailView.240x320`
  - `Template.ProfileManager.240x320`
  - `Template.ProfileEditor.240x320`
- 3 On the mobile device, navigate to the template files.
  - 4 Transfer the files to the PC (which can be done using the ActiveSync File Explorer).
  - 5 Using a text editor that supports UTF-8, search each template for properties, and update the values for your environment. Keep in mind the guidelines provided above. Typical properties to customize include:
    - Font (or `TextFont`)
    - Colors (`BackColor`, `ForeColor`, `AlternateForeGround`, `SelectionBackColor`, `ActiveColor`, `VisitedColor`, `GridColor`, and so forth)
    - Encoding (such as UTF-8)
    - English language string values (such as the profile labels used when you set up a user profile, found in *Template.ProfileEditor.240x320*)
  - 6 Save each file and close it.
  - 7 Move the template files back to the device.
  - 8 Restart the application to view the changes.



# Glossary

<b>API</b>	An acronym for application program interface – a set of routines, protocols, and tools for building software applications that enable programs to communicate with each other.
<b>ASP</b>	Active Server Pages. An open, compile-free application environment in which Web developers can combine HTML, scripts, and reusable Active Server components. ASP technology enables server-side scripting for IIS with native support for both Visual Basic Scripting Edition and JScript.
<b>adapter</b>	A component that provides an interface between an internal application and external applications or messaging systems. An adapter detects events and validates event contents.
<b>application</b>	Applications are the primary objects that developers work with in Mobile Web Studio. Applications are comprised of one or more content elements and portal pages can contain one or more applications.
<b>Application Builder</b>	An Unwired Accelerator wizard used to define applications. A succession of windows guides you through the process of creating, configuring, and customizing the application. You do not need to use all the windows to define your application; the windows needed vary depending on the type of application you are creating (for example, Web, HTML, JSP, database, document, and so forth).
<b>CGI parameters</b>	Common gateway interface parameters.
<b>channel</b>	Web content that is optimized for and delivered to mobile devices by M-Business Anywhere server. Channels are defined by a base URL and by other parameters such as channel size, link depth, image preferences, and frequency of refresh. M-Business Anywhere server automatically delivers new information from the specified URL to M-Business Client on the connecting mobile device.
<b>click across</b>	An Unwired Accelerator feature that enables you to connect related or unrelated applications in a flow using events. Alternatively, you can use the Linked Applications feature.

<b>client/server</b>	<p>A network architecture in which one or more computers (servers) accept requests for services from one or more workstations (clients).</p> <p>This may also refer to a back-end application (server) that accepts requests for information from a front-end application (client).</p>
<b>connection pooling</b>	<p>Connection pooling is a performance optimization based on using collections of pre-allocated resources, such as objects or database connections. Pooling results in more efficient resource allocation.</p>
<b>connectionless communications</b>	<p>Communications that do not require a dedicated connection or session between applications.</p>
<b>element</b>	<p>Elements are used to build applications. An application can consist of one or more elements. See <b>application</b>.</p>
<b>Define window</b>	<p>Application Builder window used to define the grid layout of an application.</p>
<b>DIOP</b>	<p>Domino Internet Inter-ORB Protocol. This method uses a CORBA connection to connect Unwired Accelerator and a Domino server.</p>
<b>enterprise</b>	<p>A reference to all aspects of a large business organization—from manufacturing to finance, marketing, to human resources. This term can also refer to an organization plus its partners, vendors, suppliers, and customers.</p>
<b>ERP</b>	<p>An acronym for Enterprise Resource Planning. An ERP is an integrated information system that serves all departments within an enterprise. ERP includes the use of packaged software rather than proprietary software. ERP components can interface with a company's own software, and also, depending on the software, ERP components can be altered using the vendor's proprietary tools as well as proprietary or standard programming languages</p>
<b>event</b>	<p>An event is a notification that occurs in response to some action. It can be a change in state or as a result of the user clicking or moving the mouse, pressing a keyboard key, or other actions that are focus-related, element-specific, or object-specific. Programmers write code that respond to these actions. An event can also be an object that is imported, passed between processors, and exported to an external database.</p>
<b>event definition</b>	<p>A set of criteria that are used to determine the contents of events.</p>
<b>Filter window</b>	<p>Application Builder window used to identify which rows, columns, and fields to use in the application and which to exclude; and to define additional grid rules.</p>
<b>Finish window</b>	<p>Application Builder window used to configure the application for use.</p>

---

<b>grid rules</b>	The Unwired Accelerator feature for manipulating the content and format of an application for display on a mobile device.
<b>HTTP</b>	HyperText Transport (or Transfer) Protocol is the set of rules that governs the exchange of text, graphic, sound, and video files on the World Wide Web.
<b>HTTPS</b>	The secure version of HTTP.
<b>Internet</b>	A global network connecting millions of computers.
<b>intranet</b>	A private network within an organization.
<b>JDBC</b>	JDBC is a data access interface based on <b>ODBC</b> and used with the <b>Java</b> programming language.
<b>J2EE</b>	Sun software: Java 2 platform, Enterprise Edition.
<b>Java</b>	Developed by Sun Microsystems, Java is an object-oriented programming language, similar to C++. Java-based applications, or applets, can be quickly downloaded from a Web site and run using a Java-compatible Web browser such as Microsoft Internet Explorer or Netscape Navigator. Java applets are the most widespread use of Java on the Web.
<b>LDAP</b>	Lightweight Directory Access Protocol. LDAP is a software protocol that allows anyone to locate organizations, individuals, and other resources (files, devices, etc.) on the Internet or on a corporate intranet.
<b>Linked applications</b>	Feature that enables you to link applications.
<b>Linked parameters</b>	Feature that enables you to set up a chained relationship among values; for example, among the drop-down fields in a Remedy applicator.
<b>M-Business Anywhere</b>	A platform for delivering Web-based content and applications to mobile devices rapidly and cost-effectively, with minimal recoding. Web developers can leverage their existing skill sets and open standards to develop and deploy fully interactive Web applications with wireless capabilities.
<b>metadata</b>	Data that describes other data. Any file or database that holds information about another database's structure, attributes, processing, or changes.
<b>Mobile Web Studio</b>	A platform for developing applications for mobile devices. Mobile Web Studio is a Web-based rapid development tool for creating powerful and interactive mobile Web applications or for mobilizing existing Web applications or data sources like databases, XML, Web Services, HTML and JSPs/ASPs.
<b>.NET container client</b>	The default .NET UA client that is packaged with Unwired Accelerator for Windows mobile devices.

<b>New element window</b>	Application Builder window used to create the element of your choice, including elements for Web, XML, HTML, JSP, database, document, and so forth.
<b>NoteID</b>	Notes Document Identifier. A unique Notes document identifier field within a Domino database. This is generally a Hex value, such as 3fe.
<b>NRPC</b>	Notes Remote Procedure Call. This is one of the connection methods for connecting Unwired Accelerator and a Domino server. Requires Notes Client to be installed.
<b>ODBC</b>	Open Database Connectivity. ODBC is a Windows standard API that is used for SQL communication to connect applications to a variety of data sources. Access is generally provided through the Control Panel, where data source names (DSNs) can be assigned to use specific ODBC drivers.
<b>Parameter definition window</b>	Application Builder window used to customize the parameters, or variables, used to capture the grid. This enables application end-users to customize or personalize parameter values when they view the application.
<b>SOAP</b>	<p>Simple Object Access Protocol. SOAP provides a way for applications to communicate with each other over the Internet, independent of platform. Remote objects can give a program almost unlimited power over the Internet, but most firewalls block non-HTTP requests. SOAP, an XML-based protocol, gets around this limitation to provide intraprocess communication across machines.</p> <p>In Unwired Accelerator, the implementation of SOAP is intended to provide businesses with a way to expose corporate software functionality to their customers with minimal firewall constraints, platform dependencies or complex development implementations involving DCOM or CORBA.</p> <p>SOAP was developed by Microsoft, DevelopMentor, and Userland Software and has been proposed to the Internet Engineering Task Force (IETF) as a standard.</p>
<b>SQL</b>	Structured Query Language.
<b>SSL</b>	Secure Sockets Layer. SSL is a standard for providing encrypted and authenticated service over the Internet. Using the Rivest Shamir and Adleman (RSA) public key, a public key cryptography for Internet security, specific TCP/IP ports can be encrypted. Primarily used for handling commerce payments, SSL is a general-purpose encryption standard for the Internet.
<b>server</b>	A computer or software package that provides specific capabilities to client software running on other computers.

<b>servlet</b>	<p>A servlet is a small, persistent, low-level program that runs on a server. The term was coined in the context of the Java applet, a small program that is sent as a separate file along with a Web (HTML) page.</p> <p>Some programs that access databases based on user input need to be on the server. These programs were most often implemented using a Common Gateway Interface (CGI) application. However, if a Java virtual machine is running in the server, servlets can be implemented in Java. A Java servlet can execute more quickly than a CGI application. Instead of creating a separate program process, each user request is invoked as a thread in a single daemon process, so that the system overhead for each request is slight.</p>
<b>sockets</b>	<p>A portable standard for network application providers on TCP/IP networks.</p>
<b>Split window</b>	<p>Application Builder window used to add parameters for splitting rows and columns in a grid. Split rules are defined for rows and columns; for delimiters; and for personalization adapters. By default this feature is not enabled, so the Split window does not appear. Set SplitEnabled to “true” in the <i>global.properties.xml</i> file to enable the feature; the Split window then appears before the Define window during capture.</p>
<b>stored procedure</b>	<p>A program that creates a named collection of SQL or other procedural statements and logic that is compiled, verified and stored in a server database.</p>
<b>style sheet</b>	<p>General term for software that transforms XML documents based on one XML vocabulary into XML documents based on a different XML vocabulary. Example stylesheets are Java Server Pages (JSPs) and XSLT stylesheets.</p>
<b>TCP/IP</b>	<p>Transmission Control Protocol/Internet Protocol—the network protocol for the Internet that runs on virtually every operating system. IP is the network layer and TCP is the transport layer.</p>
<b>Unwired Accelerator</b>	<p>A software solution that accelerates the mobilization of enterprise Web applications and data sources for constant access. Unwired Accelerator is comprised of Mobile Web Studio and M-Business Anywhere.</p>
<b>WebSession</b>	<p>Web Session based connection method. This connection method is the default connection method for connecting Unwired Accelerator and a Domino server. Requires a local Domino server to be installed.</p>
<b>Window preview window</b>	<p>Application Builder window used to view the element and give it a name.</p>

**workflow**

Software used to automatically route events or work-items from one user or program to another. Workflow is synonymous with process flow, although traditionally has been used in the context of person-to-person information flows.

**XML**

eXtensible Markup Language—a simplified subset of Standard Generalized Markup Language (SGML)—is a way to that provides a file format for representing data, a method for describing data structure, and a mechanism for extending and annotating HTML with semantic information.

As a universal data format, XML provides a standard for the server-to-server transfer of different types of structured data so that the information can be decoded, manipulated, and displayed consistently and correctly. In addition, it enables the development of three-tier Web applications, acting as the data transfer format between the middle-tier Web server and the client.

# Index

## Symbols

- @OP parameters 154
  - @OPCGIPARAM 157, 158
  - connection pools 71, 157
  - cookies 157
  - defined 157
  - exporting in XML document 317
  - grid rules 141, 157
  - inserting 93
  - modifying in XML document 317
  - SQL query 92, 157
  - URLS 157
- @OPCGIPARAM tag 158

## A

- ActiveX (ACX) application capture option 20
- ACX
  - see ActiveX 20
- adapters
  - building applications with 202
  - database table 200
  - database table, using 202
  - personalization 91, 200
  - types 200
  - user name/password 200
  - user name/password, using 202
- adding
  - applications to catalogs 267
  - new catalog branches 267
  - pages 277
  - split rules 132
  - XML\_sample application to a page 218
- agent log fills up the database 325
- agents
  - changing status 242
  - creating 242
  - running once 242

- alerts 227
  - creating 228
  - creating a schedule in Portal Interface 239
  - creating rules and schedules in Mobile Web Studio 235
  - requirements and restrictions 227
  - setting up a schedule in Portal Interface 231
  - setting up rules in Portal Interface 230
  - setting up SMS 232
  - viewing 240
- ambiguity resolution 190
- Answers Anywhere 183
  - agent network 184, 185
  - connection cache 185
  - date queries 188
  - execute agent 187
  - language syntax 186
  - library and properties folders 185
  - other language support 186
  - parameters and filters 187
  - send agent 187
  - synonyms files 184
  - uadejima.properties file 185
- Answers Anywhere tab 84
  - application synonyms 84
  - default filter field 85
  - field synonyms 84
  - parameter synonyms 85
- Application Builder 15
  - Split window 182
- application labels
  - assigning 95
  - creating 94
  - mapping 95
- Application Manager 14
  - application refresh interval 80
- applications
  - @OP parameters 92
  - ACX capture option 20
  - adding to a catalog 267

- administration tab 84
- approving 87
- CGI parameters 89
- charting 141
- click-across 103
- complex transactions 144
- content tab 77
- creating select lists for input fields 143
- creating update link 148
- creating XML 216
- database 28
- data-capable elements 130
- default size 81
- deploying to mobile devices 304
- dragging and dropping on a page 275
- enabling support for multiple languages 196
- filter rules 133
- finding and replacing 96
- HTML elements 48
- JSP 33
- JSP elements 33
- JSP SSO check box 36
- labels 94
- linked, MApp XML 146
- linking 144
- linking, restrictions 146
- List/Detail tab 82
- management functions 14
- no URL stretch option 79
- presentation tab 80
- properties 87
- publishing to a UDDI registry as a Web service 45
- removing 279
- roles tab 80
- saving 76
- secure 127
- selecting 8
- split rules 132
- types 14
- using IFrames 81
- viewer 18
- Web 18
- Web elements from secure sites 127
- Web, navigation options 19
- with personalization adapters 202
- XML 26, 28

- archive file, importing to 311
- arguments
  - avg 139
  - concatenate 140
  - diff 139
  - div 140
  - int 139
  - max 141
  - min 140
  - sum 138
- ASA for Windows CE
  - CabInstl tool 333
  - described 333
  - installing 333
- assigning application labels 95
- auto indexer
  - troubleshooting 326
- avg argument 139

## B

- BlackBerry
  - connection profile settings for synchronization 300
  - inserting multi-level records 301
- BlackBerry trackwheel menu 299
- BlackBerry/Symbian device
  - retrieving applications 296
  - syncing applications 296
- branch, catalog
  - defined 266
- branches 269
  - catalog, adding new 267
  - deleting catalog 270
  - renaming catalog 270
- broadcast events 109
- browser
  - requirements 14
- building
  - page groups 289
  - pages 277
  - templates 206, 207

**C**

- CabInstl tool
  - downloading freeware 333
  - used to install CABs 333
- CapAll 20
  - performance hit 20
- capture options
  - advanced navigation 20
  - CapAll 20
- capture strategies
  - crystal (reports) 22, 25
  - custom 22, 25
  - custom xsl 22, 25
  - description 21
  - DOM 21, 24
  - grid 22, 24
  - one-click 21, 23
  - Remedy 177
- catalog branch and subbranches, illustrated 264
- Catalog Manager 263–270
  - adding new branches 267
  - catalog branch defined 266
  - changing catalog properties 269
  - changing catalog status 269
  - creating a new catalog with 266
  - deleting applications from catalogs 270
  - deleting existing branches 270
  - described 264
  - moving existing branches 269
  - renaming existing branches 270
  - viewing catalog properties 269
- catalogs
  - adding applications to 267
  - adding new branches 267
  - changing catalog properties 269
  - changing status 269
  - creating a 266
  - deleting applications from 270
  - deleting branches 270
  - display 263
  - moving existing branches 269
  - renaming existing branches 270
  - viewing properties 269
- catalogs, updating changes to Portal Interface 270
- CellularModemController 195
- cgacutil* file 331
- CGI parameters 89
- changing
  - an agent's status 242
  - catalog properties 269
  - catalog status 269
  - page layout 274
  - template workflow status 214
- changing default express application templates 216
- channel, defined 353
- charting applications 141
- charts 142
- click-across
  - applications 103
  - improvements 102
- CMC 195
  - maintaining an access log 195
- co-brands
  - troubleshooting 326
- columns
  - changing width for pages 276
- concatenate argument 140
- conflict resolution 150
- connection cache
  - database element 28
  - JNDI option 30
- connection pool
  - SAP 73
- ConnectionPools property 73
- content style, device type 284
- content types, adding new 51
- ContextID, used to maintain linkage across applications 151
- conventions xiv
- cookies 154
- createAgent 140
- creating
  - agents 242
  - applications with personalization adapters 202
  - catalogs 266
  - database elements 28
  - HTML elements 48
  - JSP elements 33
  - mobile applications, workflow 11
  - new online device templates 220
  - new templates 213
  - new UDDI registry entries 47

- page groups 289
- personalization adapter keys 201
- select lists for application input fields 143
- user-specific UI XSLT template 218
- user-specific XML 216
- Web elements 18
- Web portlets 25
- Web service element using a service key 44
- Web service element using WSDL 42
- XML application 26, 28
- XML sample application 216
- creating catalogs 263–270
  - catalog branch and subbranches 264
  - Catalog Manager 264
- Crystal Reports capture strategy 22, 25
- css.css 321
- custom capture strategy 22, 25
- custom XSL capture strategy 22, 25
- customizing
  - BlackBerry client device templates 212
  - online client templates 205
- customizing user XML element templates 216
- cutomizing the portal 278

## D

- data types
  - creating 95, 96
- data validation 150
- data validation tab 83
  - common expressions 83
  - test regular expression 83
- database elements 28
- database table
  - adapter 200
  - adapter, using 202
- data-capable elements 130
  - filter rules 133
  - split rules 132
- date queries, Answers Anywhere 188
- default express application template, changing 215
- default filter field, Answers Anywhere tab 85
- DefaultConnectionPoolName property 75
- defining
  - broadcast events 109

- definition options, file element 54, 55
- definition, catalog branch 266
- deleting
  - applications from catalogs 270
  - catalog branches 270
  - page groups 292
  - pages 280
  - templates 214
- deploying
  - applications to BlackBerry/Symbian 296
  - applications to online devices 304
  - applications to Portal Interface 306
- determination of parameters 191
- device manager
  - and differences with template manager 205
  - customizing offline BlackBerry client templates 212
- diff argument 139
- display catalog 263
- div argument 140
- documentation
  - Adaptive Server Anywhere xiii
  - jConnect for JDBC xiii
  - Unwired Accelerator xii
- DOM capture strategy 21, 24
- Domino
  - connection pool, selecting 59
  - find function 60
  - global.properties.xml* properties 57
- downloading
  - CabInstl tool 333

## E

- EAServer
  - setting up for secure application creation 128
- editing
  - global.properties.xml* file 323
  - pages 280
  - styles.xml file 319
  - templates 214
- element list
  - Application Builder 17
  - applications 17
- elements

- database 28
- data-capable 130
- document 49
- Domino 56
- file 53
- HTML 48
- JSP 33
- SAP 70
- Web 18
- Web application 18
- Web from secure sites 127
- XML 26, 28
- elements, enabling different templates for the same 219
- enabling
  - applications and pages to support multiple languages simultaneously 196
  - different templates for the same element 219
- entering
  - user-specific personalization values in Mobile Web Studio 203
- events
  - defining broadcast 109
  - spaces in event names 120, 121, 122, 125
- examples,HTML template 208
- exporting
  - objects from Mobile Web Studio to an XML file 311
  - personalization keys 204
- exporting application parameters 317
- exporting parameters
  - @OP tags 317
  - alternative editing procedure 316
  - exposed and potential parameters 316
  - XML document 316

**F**

- field synonyms, Answers Anywhere tab 84
- file element
  - creating from a delimited file 55
  - creating from Excel spreadsheet 54
- files
  - cgacutil* file 331
  - global.properties.xml* file 57

- filter rules 133
- filter rules, adding 133
- finding applications 96

## G

- global.properties.xml*
  - configuring a Domino connection 57
- global.properties.xml* file, editing 323
- grid capture strategy 22, 24
- grid rules 135
  - arguments 138
  - include options 136
  - inserting hidden records or fields 137
  - using @OP tags 141

## H

- help templates, building 209
- HTML
  - elements 48
  - templates 208
- HTML templates 207
  - OPContent tag 207

## I

- IFrames 81
- importing
  - personalization keys 204
  - XML files into Mobile Web Studio 308
- importing application parameters 317
- In Context, used to maintain linkage 151
- inline frames, using 81
- inserting @OP parameters 93
- installing
  - .NET client on Windows device 330
  - ASA for Windows CE 333
- int argument 139
- intent XML 187, 191
- interfaces
  - Catalog Manager 264

## Index

### J

- JDBC connection
  - documentation xiii
- JNDI, using default data source 30
- JSESSIONID handling 154
- JSP
  - application SSO check box 36
  - elements 33
  - templates 211
  - templates, assigning 91

### K

- keys
  - importing and exporting personalization 204
  - UDDI service 44

### L

- labels
  - application 94
  - mapping application 95
- languages, application support for multiple languages 196
- layout, changing page 274
- limiting log size 261
- linked applications 79
  - Mobile Offline 297
  - Mobile tab 79
- linked parameters 152
- lists, creating select 143
- localization
  - display LTR or RTL 81
- logs
  - limiting the log size 261

### M

- management, application 14
- managers
  - Application 14
- map all check box, application labels 95
- MApp XML and server code 146
- master application 297

- max argument 141
- messages.xml file 322
- min argument 140
- Mobile tab
  - linked applications 79
  - offline devices 79
- moving
  - existing catalog branches 269
- moving catalog 269
- multiple template support 205

### N

- navigation options 19
- navigation styles 284, 305
- new file element 53
- no URL stretch option 79

### O

- offline devices, Mobile tab 79
- one-click capture strategy 21, 23
- OPContent tags 207

### P

- Page Builder, interface 272
- Page Group Builder, interface 282
- page groups 271
  - building 289
  - deleting 292
- Page Manager
  - displaying 271
  - using 271
- pages
  - adding 277
  - adding to page groups 289
  - adding XML applications to portal 218
  - building 277
  - changing column width 276
  - changing layout 274, 275
  - deleting 280
  - editing 271, 279

- editing and deleting 279
- introduction 271
- page groups 271
- status 214
- types 273
- viewing or editing 280
- viewing properties 280
- parameter synonyms, Answers Anywhere tab 85
- parameters 89
  - @OP 92
  - CGI 89
  - exporting and importing 316
- PDA
  - deploying applications to 304
- personalization
  - adapters 91, 200
  - keys, creating 201
  - keys, importing and exporting 204
  - values, entering user-specific in Mobile Web Studio 203
- portal objects, deploying 307
- portal page
  - adding XML applications to 218
  - creating and populating 277
- post-installation tasks
  - installing .NET client (Windows devices) 327
- properties
  - application 87
  - catalog 269
  - changing catalog 269
- publishing applications to a UDDI registry as a Web service 45

## R

- record layout
  - applications 133
  - data-capable elements 133
  - defining 133
  - tables 133
- Remedy
  - capture strategies 177
- renaming
  - catalog branches 270
- requirements and restrictions, alerts 227

- resolving ambiguity 190
- roles 305
- roles tab 80
- rules
  - avg argument 139
  - concatenate argument 140
  - diff argument 139
  - div argument 140
  - int argument 139
  - max argument 141
  - min argument 140
  - sum argument 138
- running agents once 242

## S

- SAP
  - connection pool 73
  - connection pool, selecting 71
  - find function 71
  - login 73
  - reordering columns 75
  - selecting the header row 75
- saving applications 76
  - administration tab 84
  - content cache interval 77
  - content tab 77
  - presentation tab 80
  - roles tab 80
- search APIs
  - createAgent 140
- search for application 97
- secure applications 127
  - setting up EAServer 128
- security
  - .NET accounts 330, 336
  - security with e-mail and SMS request handling 195
  - selecting, applications 8
  - server-side click-across applications 102
  - service keys, using to create Web service element 44
  - size
    - application default 81
  - SMS messages, specifying maximum number 85
  - SMS messaging 194
  - split rules

## Index

- adding 132
- split rules, defined 183
- Split window
  - defined 182
  - rules 183
- SSL 128
- Statistics tab 88
- status
  - changing agent 242
  - changing catalog 269
  - page 214
- styles.xml 320
- styles.xml file, editing 319
- Symbian/J2ME
  - menu 303

## T

- tables
  - filter rules 133
  - split rules 132
- Template Manager 205, 214
  - building templates 207
  - changing default express application templates 216
  - deleting templates 214
  - editing templates 214
  - help templates 209
  - interface 205
  - Template Upload window 206
  - XSL templates 211
- template manager 213, 220
  - and differences with device manager 212
  - online client templates 205
- templates
  - building new 207
  - changing active status 215
  - changing default express application template 216
  - changing workflow status 214
  - creating and editing 213
  - creating custom user-specific XML 216
  - creating new 213, 220
  - deleting 214, 215
  - enabling different for the same element 219
  - help 209
  - HTML 207, 208

- JSP 211
  - Template Manager interface 205
- test regular expression, data validation tab 83
- toolbar
  - Application Builder 16
- troubleshooting 325
  - agent log fills up the database 325
  - auto indexer 326
  - broken images 79
  - co-brands 326
  - display problems on BlackBerry 79
- tutorials
  - requirements 14
- typographical conventions xiv

## U

- UDDI registries
  - creating new 47
  - publishing applications to as a Web service 45
- update and conflict resolution 149
- updating
  - catalog changes to Portal Interface 270
  - records from PDA applications 149
- user name/password
  - adapter, using 202
  - adapters 200
- UserAgentMapping 285
- using @OP tags with grid rules 141
- UTF-8 196
  - and Web browsers 198

## V

- validating data 150
- validation expression, data validation tab 83
- viewing
  - application properties 87
  - applications 18
  - catalog properties 269
  - page properties 280

**W**

## Web

- browsers and UTF-8 198
- service elements 38

Web elements 18, 127

## Web service elements

- creating 42
  - creating new UDDI registry entries 47
  - creating using a service key 44
  - publishing to a UDDI registry 45
- workflow, for creating mobile applications 11
- WSDL, using to create Web service elements 42

**X**

## XML

- adding applications to a page 218
- creating applications 216
- creating user-specific XSLT templates 218
- templates, creating custom user-specific 216

XML applications 26, 28

## XML files

- exporting from Mobile Web Studio 311
- importing into Mobile Web Studio 308

## XSL templates

- creating 211
- sample 211

## XSLT templates

- creating user-specific 218

XSLT, creating 211

