SYBASE®

Mobile Application Development Tutorial

Unwired Accelerator

8.0

DOCUMENT ID: DC00124-01-0800-01

LAST REVISED: August 2006

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

Contents

About This Book

Audience

This book is for users that develop applications and deploy them to mobile devices.

How to use this book

- Chapter 1, “Introduction,” introduces Unwired Accelerator, and describes how to get started using the Mobile Web Studio.
- Chapter 2, “Getting Started,” provides basic tutorials for developing mobile applications using Mobile Web Studio.
- Chapter 3, “Creating a Multipage Mobile Application,” describes how to use click-across to create a multipage mobile application that you can deploy to a mobile device.
- Chapter 4, “Creating a Multipage Mobile Application with Transaction Support,” describes how to create a multipage mobile application with transaction support and push sync so that you can view and update data from a mobile device.
- Chapter 5, “Creating a Custom .NET Client,” describes how to create a UA .NET client or Windows Mobile Pocket PCs.
- Chapter 6, “Mobilizing BusinessObjects,” shows you how to mobilize a BusinessObjects Web Intelligence report.
- Chapter 7, “Creating a Domino Application,” describes how to create a Domino application that extracts data from your Domino database.
- Chapter 8, “Creating a Remedy Application,” describes how to create and mobilize a Remedy application.
- Chapter 9, “Creating a Multipage Mobile Charting Application,” explains how to create a mobile application from a Flash charting object.
- Chapter 10, “Deploying Applications to Mobile Devices,” describes how to deploy applications from Mobile Web Studio to mobile devices using M-Business Anywhere.
- Chapter 11, “Deploying Applications to BlackBerry Devices,” describes how to deploy applications to BlackBerry devices for use in online or offline modes.

- Chapter 12, “Troubleshooting,” provides answers for frequently asked questions, and troubleshooting information.

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 Web and a database application to either a PDA or BlackBerry device.
- The *Mobile Application Development Tutorial* (this guide) provides tutorials that help you get started using Mobile Web Studio to develop and deploy mobile applications.

Unwired Accelerator online documentation The following Unwired Accelerator documents are available on the SyBooks CD:

- The *Unwired Accelerator Developer’s Guide* includes developer-related topics for Unwired Accelerator components, Portal Interface applications, and Java Template Framework pages.
- The *Unwired Accelerator Administration Guide* provides administration topics for Unwired Accelerator and its components.
- 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 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 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.

❖ **Finding the latest information on component certifications**

1. Point your Web browser to Availability and Certification Reports at http://certification.sybase.com/.

2 Either select the product family and product under Search by Product; or select the platform and product under Search by Platform.

3 Select Search to display the availability and certification report for the selection.

❖ **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. Enter user name and password information, if prompted (for existing Web accounts) or create a new account (a free service).

3 Select a product.

4 Specify a time frame and click Go.

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: • Command names • C++ and Java method or class names

Formatting example	To indicate
variable, *package*, or *component*	Italic font indicates: • Program variables, such as *myCounter* • Parts of input text that must be substituted, for example: *Server*.log • File names
SYBASE	The variable in this manual used to represent the Sybase installation directory. Forward slashes are used for all path names, regardless of platform; for example, *SYBASE\UnwiredAccelerator80*.
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: • Information that you enter in Mobile Web Studio, on a command line, or as program text • Example program fragments • Example output fragments

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.

CHAPTER 1

Introduction

This chapter introduces Unwired Accelerator, describes how to start Mobile Web Studio, introduces the user interface, and provides information you need to get started on the tutorials.

Product overview

Unwired Accelerator accelerates the mobilization of enterprise Web applications and data sources for constant access. Two major components include:

- Mobile Web Studio – 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, databases, XML, Web Services, HTML, and JSPs/ASPs.
- MobiLink – enables you to synchronize the data in your UltraLite database application, running on your handheld device, with that contained in your consolidated database.

For detailed information, see the *Unwired Accelerator Developer's Guide*.

Introducing Mobile Web Studio

This chapter describes how to start Unwired Accelerator and use the Mobile Web Studio user interface, provides information about the sample database and applications that are included with Mobile Web Studio, and describes the basic workflow for creating and deploying mobile applications.

This chapter assumes you have:

- Installed Unwired Accelerator using information and procedures in the *Unwired Accelerator Installation Guide*
- Network access
- Internet access
- Internet Explorer installed as your browser (see the *Unwired Accelerator Release Bulletin* for version and patches)

Starting Unwired Accelerator

Before you start the tutorials, you must:

- Start the Unwired Accelerator server, which includes the portal database (portaldb) and the sample database (sampledb). The tutorials in this guide use the sampledb database. See "Tutorial environment" on page 9.
- Start the application server. The Tomcat application server is included with Unwired Accelerator.

❖ **Starting Unwired Accelerator**

1 Select Start | Programs | Sybase | UnwiredAccelerator | Start UA (Windows Services). This starts the ASA database, the Tomcat application server, and MobiLink.

2 Start Mobile Web Studio by selecting Start | Programs | Sybase | Unwired Accelerator | Start UA Studio.

3 From the Internet Explorer window, you can log in to Mobile Web Studio or Portal Interface, as described in "Using Mobile Web Studio" on page 3.

❖ **Stopping Unwired Accelerator**

- Select Start | Programs | Sybase | UnwiredAccelerator | Stop UA (Windows Services). This stops MobiLink, Tomcat, and the ASA database.

Note See the *Unwired Accelerator Installation Guide* for information about starting and stopping ASA, MobiLink, and Tomcat independently using the `startdb`, `starttomcat`, `startmlsrv`, `stoptomcat`, `stopmlsrv`, and `stopdatabase` scripts.

Using Mobile Web Studio

Mobile Web Studio is a Web application. Access Mobile Web Studio using Internet Explorer.

❖ **Logging in to the Mobile Web Studio**

1 Select Start | Programs | Sybase | Unwired Accelerator | Start UA Studio to access the Mobile Web Studio login page.

You can also access the login page by entering the following URL in Internet Explorer:

```
http://hostname.domain:port/onepage/loader.jsp
```

For example, if your machine name is "lab2k," your domain is "sybase.com," and your port number is 4040, enter:

```
http://lab2k.sybase.com:4040/onepage/loader.jsp
```

Note In a development environment, your port number may be different, and in a production system, you may not need to specify the port number if you are using port 80.

2 When the Mobile Web Studio Login window displays, enter the user name `masuper` and password `m8super`, and click Login. These are the default entries for the Mobile Web Studio account with administrative privileges.

You see the Mobile Web Studio welcome window.

Figure 1-1: Mobile Web Studio Welcome window

Note If your browser session expires, you see `Session has expired. You need to re-authenticate`. Click OK to close the message window, start a new browser session, log in, and close the old window session.

The welcome window displays:

- Left pane menus – select from menu groupings that allow you to:
  - Build – create, edit, and manage applications, templates, catalogs, pages, page groups, and composite applications.

  - Automate – create agents to automatically process application content, monitor agent activity, or use adapters to write application content to an e-mail message, database, or file system.
  - Manage – manage Mobile Web Studio resources and create predefined application input fields or drop-down lists, and deploy applications to the MobiLink and M-Business Anywhere servers.
- Status bar – always shows the user name of the person logged in to Mobile Web Studio (User), the group of application pages available to this user (Resource), and the version information (About Mobile Web Studio).
- Toolbar – view your account information, change the password, and log out of Mobile Web Studio. Once you make a selection from the left pane, an option-specific toolbar displays.

Note To see all features, you may need to maximize the browser window.

Selecting a Mobile Web Studio menu option

Once 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.

Build menu

Objects on the Build menu include:

- Applications – create, edit, delete, and manage applications and their content.
- Templates – define the organization of aggregate applications (applications with more than one element), where applications are located, background HTML code, and so on. You can apply the templates you create to applications. Templates help applications display on specific device types. You can assign a different template for each device type for a given application.
- Catalogs – create user-defined hierarchies of content for display in an application. See the *Unwired Accelerator Developer's Guide* and the *Portal Interface User's Guide*.
- Pages – create, edit, delete, and manage the Portal Interface pages on which applications display. See the *Unwired Accelerator Developer's Guide* and the *Portal Interface User's Guide*.

- Page Groups – create, edit, delete, and manage page groups to organize Portal Interface application pages. Page groups provide a container for deploying applications or groups of applications to Portal Interface or to mobile devices in offline (disconnected) mode. See the *Unwired Accelerator Developer's Guide* and the *Portal Interface User's Guide*.
- Composite Apps – create virtual Web applications using several existing applications.

Automate menu

Objects on the Automate menu include:

- Agents – create, edit, manage, start, stop, and view logs for agents that automatically process application content.
- Adapters – create, edit, manage, and view logs for adapters that write content to e-mail, databases, or file systems. See the *Unwired Accelerator Developer's Guide*.
- Servers – create, manage, and view logs for servers on which agents run. See the *Unwired Accelerator Developer's Guide*.

Manage menu

Objects on the Manage menu include:

- Users/Roles – create and edit users, and manage the resource with which they are associated. Create, edit, and manage roles, and assign roles to users. See the *Unwired Accelerator Administration Guide*.
- Personalization – create and manage keys that allow users to personalize applications to their needs. See the *Unwired Accelerator Developer's Guide*.
- M-Business – deploy applications to M-Business Anywhere server groups, channels, and users, and perform some M-Business Anywhere administration tasks specifically required by Mobile Web Studio, such as user account maintenance. See the *Unwired Accelerator Administration Guide*.
- MobiLink – view and delete applications deployed to MobiLink.
- Devices – create and edit customized device templates.

Once you make a selection, the Manage menu, detail window, and toolbar for the selected component display.

Object managers

All object managers (Application Manager, Template Manager, and so on) display similar object-specific user interface components:

- Manage menus – display View By options to change the grouping of applications or pages. For applications, the menu changes the grouping from status to category. For pages and page groups, the menu changes the grouping from status to type.
- Detail window – displays the type of items you selected from the manager menu; for example, only new or approved portlets.
- Toolbar – lets you perform object-specific activities; for example, create a new object, edit, preview, and so on.
- Pop-up menus – let you perform object-specific activities on the selected item in the list.

Manager menus

Object managers allow you to display items based on your menu selection. You can display applications by status or category. Additional options on the toolbar allow you to filter the selected lists by additional criteria.

Detail window

The detail window displays items for the selected object based on your selections from:

- Filter By – choose to see all items or only the items created by you.
- Show Active Only – choose to see only the selected items (portlets, templates, and pages) that are marked as active.
- View By – view items by status or category. Templates and catalogs display only by status.
- Menu – view items of a specific status or category. Templates, catalogs, and composite applications display items only by status.

If a list contains more items than can display in one window, numbers display directly below the detail window that allow you to navigate to the next group of items in the list.

Toolbars

Above the detail window are icons that allow you to perform a variety of activities on an item in the detail view. The icons that appear vary, depending on the object manager with which you are working. For applications, the buttons are:

- New – create a new object.

- Edit – edit an existing object.
- Preview – test the object.
- Find – locate an object.
- Replace – replace an object with a newer version.
- Import/Export – import or export an object into or out of Mobile Web Studio.
- Publish to UDDI.

Pop-up menus

Once a list displays in the detail window, right-click a list item to display an object-specific pop-up menu showing activities you can perform on the selected item.

Using Portal Interface

Unwired Accelerator provides browser access to Portal Interface, which can be a useful tool for testing and troubleshooting mobile applications.

❖ **Logging in to the mobile device interface**

1 On your browser or mobile device, enter the following URL:

```
http://hostname.domain:port/onepage/mpindex.jsp
```

For example, if your machine name is “lab2k,” your portal domain is “sybase.com,” and your port number is 4040, enter:

```
http://lab2k.sybase.com:4040/onepage/mpindex.jsp
```

2 Enter a mobile portal user name and password, and click Login.

The home page displays the navigation tree with the names of the mobile applications as links.

Workflow for creating mobile applications

This section describes the basic workflow for creating mobile applications using Mobile Web Studio. Basic tasks include:

1 Create an element – create the element using the Application Builder.

2 Create the application – use the Application Builder wizard to create the application using the element, and customize how the application looks and operates. The Application Builder wizard is described in more detail in "Introduction to creating applications" on page 12.

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.

Tutorial environment

This section describes the tutorial environment. The tutorials in this guide use the sample database and a sample application included with Unwired Accelerator.

- A sample database called sampledb provides tables that are populated with employer, employee, customer, and product information. The sampledb is located in *%SYBASE%\UA80\asa*.

The `masuper`/`m8super` account is the recommended account to use with the tutorials, so you have Unwired Accelerator capabilities. See the *Unwired Accelerator Administration Guide* for more information about security, roles, and procedures for setting up new accounts.

CHAPTER 2

Getting Started

This chapter describes how to get started using Mobile Web Studio and the tutorials.

Overview

This chapter provides several basic tutorials, and discusses basic concepts and terminology you will need for the more difficult tutorials in later chapters.

The procedures in this chapter assume:

- Mobile Web Studio is installed.
- You have Internet access.
- Internet Explorer is installed with the security patches listed in the Unwired Accelerator release bulletin.
- Pop-up blockers installed on your Web browser are disabled.
- Database and application servers are started as described in "Starting Unwired Accelerator" on page 2.
- You know how to log in to Mobile Web Studio as described in "Using Mobile Web Studio" on page 3, and are using the `masuper`/`m8super` account.
- A BlackBerry device or simulator, or PDA is installed and connected as described in the *Unwired Accelerator Installation Guide*.

Introduction to creating applications

This section provides basic tutorials for:

- Creating a mobile application
- Approving an application
- Searching for an application
- Working with templates
- Defining grid rules
- Defining simple click-across events
- Defining client-side click-across listeners
- Defining parameters using @OP tags

Application Builder wizard

Application Builder is a wizard that includes a succession of windows, you can use to define applications. The windows differ depending on the type of application you are creating (Web, HTML, JSP, and so forth). Application Builder includes:

- New Element window – used to create the application type of your choice, including elements for Web, XML, database, JSP, Web service, HTML, document, file element, and SAP, if configured.
- Split window – used to add parameters for splitting the columns or rows in a table. 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.

By default, the Split window, which is set in the *global.properties.xml file*, is hidden. The property is:

```
<Property name="SplitEnabled" value="false"
description="Whether to expose grid split UI.
true/false" menugroup="10"/>
```

See the *Unwired Accelerator Developer's Guide* for more information about personalization adapters.

- Define window – used to define the grid layout, for example, whether to display record labels or just the records.
- Filter window – used to identify rows, columns, and fields in the application, as well as to specify additional grid rules. Grid rules enable you to manipulate the content and format of an application for display on a mobile device.
- Configure Parameters window – used to customize the parameters, or variables, used to capture the grid or table. This enables application users to customize or personalize parameter values when they view the application on the mobile device.
- Window Preview window – used to view the element and to give it a name.
- Finish window – used to configure the application.

Note Leave the Application Builder window open while using the wizard.

Creating a JSP application

This tutorial demonstrates how to create a JSP application; select and apply a template; create an XML application; and create a server-side click-across event to link a displayed column (Headline) to a hidden column (URL). This tutorial also shows how to use grid rules to optimize data presentation on a mobile device. (See the *Unwired Accelerator Developer's Guide* for more information about grid rules). The resulting NewsStory application enables you to select a headline and display the corresponding news story.

This tutorial is divided into several procedures, so you can refer to them later when you create your own applications.

❖ **Creating the NewsStory (JSP) application**

1 Log in to Mobile Web Studio.

2 Select Applications in the left pane, and click the New button to launch Application Builder.

3 In the Application Builder, click the down arrow next to the Add button. In the menu that appears, select JSP Element.

4 In the JSP Element window, enter:

- Use Web Application – select this option.
- WAR File – enter `onepage.war`.
- Web App Display Name – enter `onepage`.
- Initial Resource – enter `/portlets/jsp/documentdisplay.jsp`.
- Web App Qualified URLs – select this option.
- Enable Grid Rules – unselect this option.
- Single-Sign On Required – unselect this option.
- Input Parameters – enter `url=http://www.yahoo.com`. Later, you will create the TopNews application and create an event for its `url` parameter.

Note For remote JSP applications, only applications containing grid data or simple HTML are supported on Wireless Application Protocol (WAP) devices.

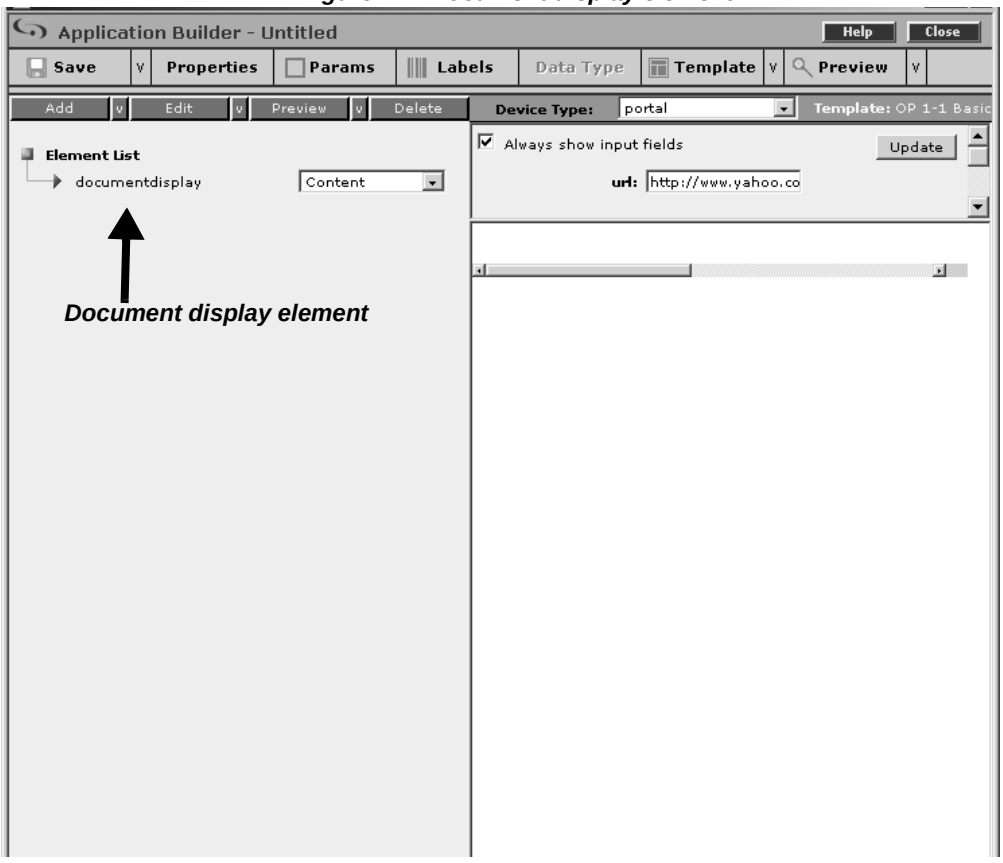
---

5 Click Preview to preview the JSP page showing Yahoo's Web site. If you do not see the Yahoo Web site, verify your input; you must see the Yahoo Web site before you go on.

6 Click Next to continue.

7 In Element Name, enter `documentdisplay`, and click Finish. The document display element is added to the Element List in Application Builder.

Figure 2-1: Document display element



The JSP application does not yet display in the Application Builder preview pane. You must change the template so that the IFRAME in the `document display` JSP expands and fills the application's available screen space.

8 Click the Template button in Application Builder.

9 In the Find Template window, verify that "html" is selected from the Type drop-down list, then click Search.

10 When the template list appears in the detail pane, select the OP Basic template, and click Select. (The template name indicates the template layout; for example, OP-1-1 Basic means the template creates a table with one row and one column).

11 The Find Template window closes and the template is applied to the application. It may take a few moments for the new preview of the application to appear showing the Yahoo.com Web site in the preview pane.

12 In the Application Builder window, click Save.

13 On the Finish window, make these entries; for all other fields, accept the defaults:

Content tab Name – enter `NewsStory` (no spaces).

Presentation tab Select No Popup.

14 Click Finish to create the application.

15 Click OK to confirm.

16 Click the Close button in the upper-right corner to close the Application Builder window.

17 In Mobile Web Studio, approve the application:

- Click New under Application Manager Status in the middle pane. The NewsStory application displays in the detail pane.
- Right-click NewsStory, and select Approval Status | Approved.
- Click OK to confirm.

18 Select Approved from the Application Manager Status menu. You see the NewsStory application in the detail pane.

19 You can preview the NewsStory application, although it is not yet complete:

- Select NewsStory in the detail pane, and click Preview. The Yahoo.com Web site displays in a window.
- Close the window.

❖ **Creating the TopNews (XML) application**

In this step, create an XML application called, “TopNews,” and use grid rules to modify its presentation for the mobile device. You will display the three most important columns, and identify two hidden columns, one of which will be used in a click-across event.

Note Currently, only JSP templates are valid for mobile applications.

---

1. Log in to Mobile Web Studio.
2. Select Applications in the left pane, and click the New button to launch Application Builder.
3. Click the arrow to the right of Add, and select XML Element. You see the XML Element Definition window.
4. In XML URL, enter:

```
http://www.moreover.com/cgi-local/page?o=xml_1&c=Top%20US%20stories
```

5. For Content XSLT, click the Select button next to the text box. The Find XSLT Template window displays.
6. In Type, make sure the drop-down list is set to XSL, and click Search in the upper-left corner of the window. A list of templates displays in the detail pane.
7. Click the moreover_content_xsl template.
8. Click Select to set this as the Content XSLT template for the XML Element. The Find Template window closes.
9. On the XML Element Definition window, click Preview to test the settings entered. After a few moments a seven-column table with multiple rows displays in the Preview pane.
10. Click Next to continue.
11. Use the Define window to identify record 1 as the header row. In the Define Record Layout section, click “Records contain labels.” When the “Labels are displayed in Record” option displays, accept the default, `1`. Click Next.
12. On the Filter window, rename the headers of the second column:, third, fourth, sixth, and seventh columns. The procedure for changing the headers is the same for all the columns:

    To change the header of the second column:

    a. In the left-most drop-down list, select “Edit Record.”
    b. Make sure the second drop-down list is set to “number.”
    c. In the first text box, enter `1` to indicate the first record.
    d. For Field, select “number.”
    e. In the second text box, enter the number of the column of which you are changing the name, for example `2` to indicate the second column.

f In the fourth drop-down list, select “value” from the “value/image” drop-down list.

g In the third text box, enter a name for the field. For example, for column 2, enter `URL`. This creates the header title for the column.

h Click Add. In Preview, the field title has changed to the value you entered, and a new rule is added under Current Filter Rules.

This table shows the values to enter for each column:

	Col 2	**Col 3**	**Col 4**	**Col 6**	**Col 7**
Step e	2	3	4	6	7
Step g	URL	Headline	Source	SourceURL	Date

- To include only columns 3, 4, and 7:
  - In the left-most drop-down list, select “Include field(s).”
  - Make sure the second drop-down list is set to “number.”
  - In the text box, enter `3,4,7` to indicate you want to include these columns (Headline, Source, and Date).
  - Click Add. In Preview, field 3, 4, and 7 are highlighted in blue, and a new rule is added under Current Filter Rules.
- To indicate that columns 2 and 6 are hidden from view:
  - In the left-most drop-down list, select “Include hidden field(s).”
  - Make sure the second drop-down list is set to “number.”
  - In the text box, enter `2,6` to indicate these columns (URL and SourceURL) should be hidden from view.
  - Click Add. In Preview, fields 2 and 6 are highlighted in green, and a new rule is added under Current Filter Rules.

Click Next.

13 In the Configure Parameters window, click Next.

14 The Preview window displays the modified table in the lower part of the screen. Only Headline, Source, and Date are displayed (3,4,7) in grid format.

15 In Element name, enter `TopNews` (no space) as the XML Element name, and click Finish.

The XML Element Definition wizard closes and the XML Element displays under the Element List in the Application Builder window.

16 In the Application Builder window, click Save.

17 In the Finish window, make this entry; accept the defaults for all other fields:

Content tab Name – enter `TopNews` (no space).

18 Click Finish.

19 Click OK to confirm.

20 Click Close in the upper-right corner to close Application Builder.

21 Click New under Application Manager Status in the middle pane. The TopNews application displays in the detail pane. Do not approve it yet.

❖ **Defining a click-across event**

In this step, define a server-side click-across event for the TopNews application that links the Headline field with the hidden URL field. You also use a grid rule formula to link the two fields.

1 Right-click TopNews in the detail pane, and select Define Events.

2 In the Define Click-Across Events window, click Select to the left of the data grid. The hidden fields (URL and SourceURL) are included in the grid.

3 Click Next to continue.

4 Under Assign An Event, enter these values:

- Row – enter `2-` (the number 2 followed by a dash, no space) This places the event on record 2, the last row, excluding the header. Alternatively you could enter `all` to include the header as well as all the rows.
- Column – enter `2` for Headline.
- Event Name – enter `url`. This event name is the display name of the Input Parameter you entered (`url=http://www.yahoo.com`) when you created NewsStory.
- With Drop Down List – select “Cell value (other cell).” A formula (`$R0F0`) displays in the formula box. `R0` indicates a row index, and `F0` indicates the field, or column, index.

- Formula Box – replace the default formula with `$R0F1` to indicate any row (`R0` indicates the current row being processed) and the first column of the row (`F1` indicates column 1) of the application. In this case, `$R0F1` links the hidden URL field (column 1) with Headline (column 2) to. See the *Unwired Accelerator Developer's Guide* for information about formulas in grid rules.
- Multi-value – leave unselected.
- Client-side – make sure the option is unselected to indicate the event is a server-side event. This option acts like a toggle. When the option is selected, it indicates a client-side event; when the option is not selected, it indicates a server-side event.
- Click Find Application to open the Search window, and click Search.
- In the Results pane, select NewsStory and select Add. The Search window closes and NewsStory displays in the Application Name field.

5 In the Assign An Event section, click Add. After a moment, the second field in the grid is highlighted in blue to show the event is defined for the field. The fields are underlined indicating a link is established. This is the link to the hidden URL field.

6 Click Next to continue. You see a preview of the application for which the event was defined.

7 Click Finish to save the event definition to the application.

8 In Mobile Web Studio, approve the application:

- Right-click the TopNews application in the detail pane, and select Approval Status | Approved.
- Click OK to confirm.

9 Select Approved from the Application Manager Status menu. You see the NewsStory application in the detail pane.

❖ **Previewing the TopNews application**

1 Select the TopNews application in the detail pane, and click Preview. The Web site where the news story is located displays.

2 Select a Headline link.

3 Verify that the correct article displays.

4 Close the window.

❖ **Approving the application**

1 Right-click the TopNews application in the detail pane, and select Approval Status | Approved.

2 Click OK to confirm.

3 Select Approved under Application Manager Status and verify the TopNews application displays.

Using the @OP tag to insert parameters

This tutorial introduces you to the basic process of using the @OP tag to insert a parameter. At runtime, the parameter is replaced with the data that you specify. The @OP tag is often used in SQL queries, as it allows an application to use parameterized queries. You can also use the @OP tag for HTML fields.

See the *Unwired Accelerator Developer's Guide* for information about grid rules, and using @OP tags in grid rules and with HTML.

Note When you create an application with input parameters, you must define a default value to register as click-across event listeners.

❖ **Creating a parameter using the @OP tag**

1 Log in to Mobile Web Studio.

2 Select Applications in the left pane, and click the New button to launch Application Builder.

3 In Application Builder, click the down arrow to the right of Add, and select Database Element.

4 On the Database Element Definition window:

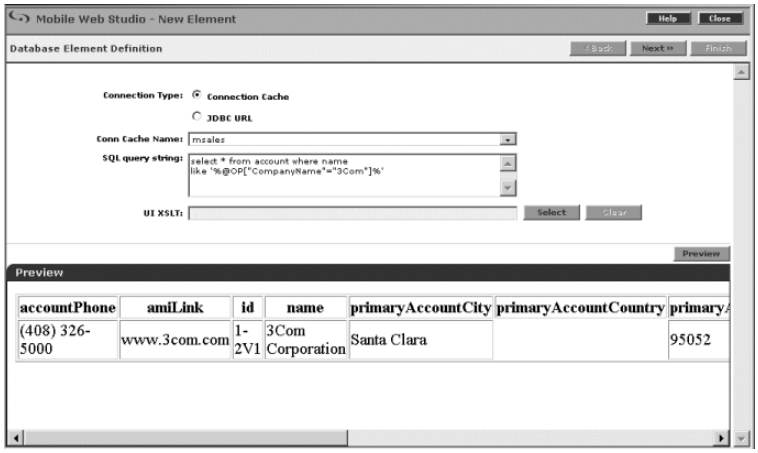
- Click the Connection Cache option.
- Conn Cache name – select `msales` from the drop-down list.
- SQL query string – enter:

```
select * from account where name like
'%@OP["CompanyName"="3Com"]%'
```

This @OP tag creates a parameter named CompanyName and sets its default value to 3Com.

- Click the Preview button to test the database connection and the SQL command.

Figure 2-2: Database element definition



5 On the Database Element Definition window, click Next.

6 Use the Define window to identify record 1 as the header row. In the Define Record Layout section, click “Records contain labels,” accept the default `1` in the “Labels are displayed in Record” option, and click Next.

7 On the Filter window, click Next to continue as this feature is not used in this tutorial. The Configure Parameters window displays with a list of parameters used to capture the table.

8 On the Configure Parameters window:

- Clicking the Variable check box to the left of CompanyName exposes additional parameters.
- Make sure the Default Value is set to `3Com`, and click Next.

9 On Window Preview, in Element Name, enter `CompanyAccount` (no space), and click Finish. The New Web Element window closes.

10 On the Application Builder window, the `CompanyAccount` element displays in the Element List. Click Save.

11 On the Finish window, accept the defaults, and click Finish to save the application.

12 Click OK to confirm.

13 In Application Builder, the CompanyAccount application displays in the detail pane.

14 In Company Name, enter part of a name of any other company or companies in the sample msales database and click Update.

For example, enter `tech` to pull up all account records that have the string "tech" somewhere in the name field. Other examples you can enter include `store`, `oak`, `health`, `auto`, and `corp`.

15 Click Close in the upper-right corner to close Application Builder.

16 In Mobile Web Studio, approve the application:

- Click New under Application Manager Status in the middle pane. The CompanyAccount application displays in the detail pane.
- Right-click the CompanyAccount application, and select Approval Status | Approved.
- Click OK to confirm.

17 Select Approved from the Application Manager Status menu. You see the CompanyAccount application in the detail pane.

You can use multiple @OP tags simultaneously, as shown in this example SQL query that uses two @OP tags to perform a search based on a company name and a state.

```
select * from account where name like
'%@OP["CompanyName"="3Com"]%' and primaryAccountState
like '%@OP["State"="CA"]%'
```

If you use multiple @OP tags, additional parameters appear on the Configure Parameters window when you create the application.

When you create an application with input parameters, you must define default values to register as click-across event listeners.

Deploying the application

You can continue with the tutorials in the order presented, or you can skip ahead and deploy the CompanyAccount application:

- Deploy the application to a PDA, such as PocketPC or Palm OS. See Chapter 10, "Deploying Applications to Mobile Devices."

- Deploy the application to a BlackBerry device. See Chapter 11, “Deploying Applications to BlackBerry Devices.”

CHAPTER 3

Creating a Multipage Mobile Application

This chapter shows how to create a multipage mobile application using several applications and events.

Overview

Mobile Web Studio enables you to create multiple-page mobile applications using a capability called click-across, which enables you to connect related or unrelated applications in a flow using events. An Event wizard guides you through defining events on the rows, columns, or cells in a grid application.

For example, in a human resources application that lists departments and department heads, you click on a department name, such as Finance, to view a list of employees in the department, or click on a department head to view information about the department manager and about employees in the department.

See the *Unwired Accelerator Developer's Guide* for more information.

Multipage mobile application tutorial

This tutorial shows how to create a multiple-page mobile application. In the tutorial, you set up three applications—a master application, and two drill-down applications—that extract data from the sample database (sampledb). Then use the Event wizard to create two events—one for extracting department information and one for extracting employee information. Finally, use all the components to create a multiple-page mobile application that can be deployed to a mobile device. These tasks are described in these sections:

- "Creating multiple-page applications" on page 27
- "Creating events" on page 35

This procedure assumes you:

- Know how to log in to Mobile Web Studio.
- Know how to create and approve an application in Mobile Web Studio.
- Understand basic SQL syntax.
- Understand how the @OP tag works as demonstrated in "Using the @OP tag to insert parameters" on page 21, and described in the *Unwired Accelerator Developer's Guide*.
- Are running the sampledb database. The example uses port 4747.

Note This tutorial uses a direct JDBC connection to sampledb when creating database elements. An alternate approach is to use a connection cache with sampledb, which is demonstrated in "Mobile applications with transaction support tutorial" on page 41.

The advantage of connection caches is that connection details, such as JDBC connection string, user name and password, driver, and so on, are defined when the application server is set up and available to developers building JDBC applications. The disadvantage is that connection changes, which must be made through the application server, may cause applications to break.

In general, connection cache definitions are safer and easier to use, but if a connection to a database is required that is not available through a connection cache, you may have to either use the full JDBC specification, or to create the necessary connection cache definition through the application server.

Creating multiple-page applications

This procedure describes how to create the three applications needed for the multiple-page application example. You will set up a master application and two child, or drill-down, applications, using database elements that extract data from sampledb. The three applications are:

- SSCAMaster – displays a list of departments and department heads within an organization.
- SSCADepartment – displays the employees in a department. An @OP tag is defined so that a default Dept Name of “Shipping” is displayed if no other department name is returned from the database at runtime.
- SSCAEmployee – displays information about an employee. An @OP tag is defined so that a default Emp Name of “Fran Whitney” is displayed if no other employee name is returned from the database at runtime.

The completed example lets you click a department name to display a list of employees in the department, and click a department head name to display information about the employee.

You can click an employee to display details about that employee.

Creating the SSCAMaster application

The master application displays a list of departments within an organization. The application has two columns for department name, and department manager data.

❖ **Defining the SSCAMaster application**

1 Log in to Mobile Web Studio.

2 Select Applications from the Build menu in the left pane, select New under Application Manager, and click the New button to launch Application Builder.

3 Click the down arrow to the right of Add, and select Database Element.

4 In the Database Elements Definition window, enter the JDBC information. The JDBC connection allows the master application to access a database and extract specific database information. This example uses sampledb on port 4747.

   a Select the JDBC URL option.

b Complete these options:

- Username – enter `dba`, which is the ASA database user name.
- Password – enter `SQL`, which is the password used to access the ASA database.
- JDBC Connect URL – enter the URL used by JDBC to connect to the database:

```
jdbc:sybase:Tds:localhost:4747?ServiceName=sampledb
```

- JDBC driver – enter the JDBC driver used to connect to the database, in this case:

  ```
  com.sybase.jdbc2.jdbc.SybDriver
  ```

- SQL Query String – enter the following SQL code to select department heads for each department from sambledb:

  ```
  select d.dept_name as 'Dept Name',
  e.emp_fname + ' ' + e.emp_lname as 'Dept Head'
  from department d, employee e
  where d.dept_head_id = e.emp_id
  ```

- UI XSLT – leave this field empty.

5 Click Preview. A two-column table showing Dept Name and Dept Head displays in the Preview panel.

Dept Name	Dept Head
R & D	David Scott
Sales	Judy Snow
Finance	Mary Anne Shea
Marketing	Scott Evans
Shipping	Jose Martinez

Note If you click on the R & D row, the query does not work because of the "&".

6 Click Next. The Define window displays.

7 Use the Define window to identify record 1 as the header row. In the Define Record Layout section, click "Records contain labels," accept the default, `1`, in the "Labels are displayed in Record" option, then click Next.

8 On the Filter window, click Next.

9 On the Configure Parameters window, click Next.

10 In the Window Preview window, in Element Name, enter `SSCAMaster`. Under Window Preview, notice the grid showing Dept Name and Dept Head.

Click Finish.

11 On the Application Builder window, click Save.

12 In the Finish window, accept all the defaults and click Finish to save the application..

13 Click OK to confirm.

14 Click Close to exit the Application Builder.

15 In Mobile Web Studio, approve the application:

- Right-click the SSCAMaster application in the detail pane, and select Approval Status | Approved.
- Click OK to confirm.

16 Select Approved from the Application Manager Status menu. You see your newly approved SSCAMaster application in the detail pane.

Creating the SSCADepartment application

This procedure describes how to define the drill-down application named SSCADepartment. This application displays the employees in a department. The application uses an @OP tag to dynamically select rows from the database to replace the Dept Name parameter. See “Using the @OP tag to insert parameters” on page 21 for information.

❖ Defining the SSCADepartment application

1 In Mobile Web Studio, select Applications from the Build menu in the left pane, select New under Application Manager, and click the New button to launch Application Builder.

2 Click the down arrow to the right of Add, and select Database Element.

3 In the Database Elements Definition window, enter JDBC connection information needed for the application to access a database and extract specific database information.

This example uses sampledb on port 4747.

a Click the JDBC URL option button.

b Complete these options:

- Username – enter `dba`, which is the user name used to access the database.
- Password – enter `SQL`, which is the password used to access the database.
- JDBC Connect URL – enter the URL used by JDBC to connect to the database:

  ```
  jdbc:sybase:Tds:localhost:4747?ServiceName=
  sampledb
  ```

- JDBC driver – enter the JDBC driver used to connect to the database, in this case:

  ```
  com.sybase.jdbc2.jdbc.SybDriver
  ```

- SQL Query String – enter the following SQL code to select department heads for each department from sampledb:

  ```
  select e.emp_fname + ' ' + e.emp_lname
  as Employee from employee e, department d
  where e.dept_id = d.dept_id and
  '@OP["Dept Name"="Shipping"]' = d.dept_name
  ```

  Notice the use of the @OP tag. This indicates that the application uses a parameter (Dept Name) to replace the tag at runtime. If the parameter is not defined, the @OP tag is replaced by the default value, Shipping. This ensures that previews, where parameters may not be explicitly defined, result in some valid data being returned.

  Note When you create an application with input parameters, you must define a default value to register as click-across event listeners.

- UI XSLT – leave this field empty.

4 Click Preview. A one column table showing Employee displays in the Preview panel.

Employee
Jeannette Bertrand
Jose Martinez

Employee
Jane Braun
Felicia Kuo
Charles Crowley
Joseph Barker
Anthony Rebeiro
Sheila Romero
Michael Lynch

5 Click Next. The Define window displays.

6 On the Define window, identify record 1 as the header row. In the Define Record Layout section, click "Record Contains Labels," accept the default, `1`, and click Next.

7 On the Filter window, click Next. The Configure Parameters window displays with a list of parameters.

8 On the Configure Parameters window, specify "Shipping" as the default Department Name for the @OP tag, and define Dept Name as the default value. Accept the defaults including:

- Variable – select the Variable option to the left of Dept Name. The variable moves to the bottom of the list.
- Display Name – accept Dept Name.
- Default Value – accept Shipping.
- Type – accept Text Field.

Click Next.

9 The Window Preview displays. In Element Name, enter `SSCADepartment`. Under Window Preview, notice the grid showing Employee.

Click Finish.

10 When you return to the Application Builder, click Save.

11 In the Finish window, accept the defaults, and click Finish to save the application.

12 Click OK to confirm.

13 Click Close to exit the Application Builder.

14 In Mobile Web Studio, approve the application:

- Right-click the SSCADepartment application in the detail pane, and select Approval Status | Approved.
- Click OK to confirm.

15 Select Approved from the Application Manager Status menu. You see your newly approved SSCADepartment application in the detail pane.

Creating the SSCAEmployee application

Define the second drill-down application—SSCAEmployee. This application displays employee information. The application uses an @OP tag to dynamically select rows from the database to replace the Emp Name parameter.

❖ Defining the SSCAEmployee application

1 From Mobile Web Studio, select Applications from the Build menu in the left pane, select New under Application Manager, and click the New button to launch Application Builder.

2 Click the down arrow to the right of the Add button, and select Database Element.

3 In the Database Elements Definition window, enter JDBC connection information needed for the application to access a database and extract specific database information.

The values in the following example assume that sampledb has been started, and is accessible on port 4747.

a Click the JDBC URL option button.

b Complete these options:

- Username – enter `dba`, which is the user name used to access the database.
- Password – enter `SQL`, which is the password used to access the database.

- JDBC Connect URL – enter the URL used by JDBC to connect to the database:

```
jdbc:sybase:Tds:localhost:4747?ServiceName=sampledb
```

- JDBC driver – enter the JDBC driver used to connect to the database, in this case:

```
com.sybase.jdbc2.jdbc.SybDriver
```

- SQL Query String – enter the following SQL code to select employees from sampledb. Cut and paste this code from the HTML version of this guide:

```
select 1 as ID, 'Employee ID' as Item,
convert(varchar(255), emp_id) as Data from employee e
where '@OP["Emp Name"="Fran Whitney"]' = emp_fname + ' ' + emp_lname

                    UNION

select 2 as ID, 'Employee Name', emp_fname + ' ' + emp_lname
from employee e where '@OP["Emp Name"="Fran Whitney"]' = e.emp_fname
+ ' ' + e.emp_lname

                    UNION

select 3 as ID, 'Manager Name', e2.emp_fname + ' ' + e2.emp_lname
from employee e, employee e2
where e2.emp_id = e.manager_id and '@OP["Emp Name"="Fran Whitney"]'
= e.emp_fname + ' ' + e.emp_lname

                    UNION

select 4 as ID, 'Dept Name', d.dept_name
from employee e, department d
where e.dept_id = d.dept_id and '@OP["Emp Name"="Fran Whitney"]' =
e.emp_fname + ' ' + e.emp_lname

                    UNION

select 5 as ID, 'Birthdate', convert(varchar(255), birth_date)
from employee e
where '@OP["Emp Name"="Fran Whitney"]' = e.emp_fname + ' ' +
e.emp_lname

                    order by 1
```

This includes the @OP tag, Emp Name, which sets the default value to Fran Whitney. In a later step, you will assign the default on the Parameter page to complete the process.

Note When you create an application with input parameters, you must define a default value to register as click-across event listeners.

4 Click Preview. A three-column table showing ID, Item, and Data displays in the Preview panel.

ID	Item	Data
1	Employee ID	102
2	Employee Name	Fran Whitney
3	Manager Name	David Scott
4	Dept Name	R & D
5	Birthdate	Jun 05 1959 12:00AM

5 Click Next. The Define window displays.

6 Use the Define window to identify record 1 as the header row. In the Define Record Layout section, click "Records contain labels," accept the default, `1`, then click Next.

7 In the Filter window, define a filter that excludes column (field) 1.

- In the left-most drop-down list, select "Exclude field(s)."
- The second drop-down list is set to "number."
- In the text box, enter `1` to exclude column 1.
- Click Add. In Preview, field 1 is highlighted in a contrasting color, and a new rule is added under Current Filter Rules.

Click Next.

8 When the Configure Parameters window displays, specify the @OP tag as an application parameter, and define Fran Whitney as the default value for Emp Name. Co. Select the Variable option to the left of Emp Name. The variable moves to the bottom of the list. Accept these defaults:

- Display Name – Emp Name.
- Default Value – Fran Whitney.
- Type – Text Field.

Click Next.

9 The Window Preview displays. In Element Name, enter `SSCAEmployee`. Under Window Preview, notice the grid showing Item and Data.

Click Finish.

10 When you return to the Application Builder window, click Save.

11 In the Finish window, accept the defaults, and click Finish to save the application.

12 Click OK in the confirmation pop-up window.

13 Click Close to exit the Application Builder.

14 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The SSCAEmployee application displays in the detail pane.

15 Right-click the SSCAEmployee application in the detail pane, and select Approval Status | Approved.

Click OK to confirm.

16 Select Approved from the Application Manager Status menu. You see the SSCAEmployee application in the detail pane.

Creating events

This section shows how to create events for the three applications you created in “Creating multiple-page applications” on page 27. The events link the master application with the two drill-down applications. This lets you view details obtained from the drill-down applications through the master application. You use the @OP tag to display default data if no data is available. These tasks are described in these sections:

- “Creating a link to department information” on page 36
- “Creating a link to employee information” on page 38

This procedure assumes you:

- Know how to log in to Mobile Web Studio.

- Have built the database applications described in “Creating multiple-page applications” on page 27.

Note If the user is deactivated or deleted, the Mobile Web Studio administrator must also locate the events the user created and delete them as well. See the *Unwired Accelerator Administration Guide* for user account information.

Creating a link to department information

In this procedure, create an event that links department information with the SSCAMaster application.

❖ **Creating an event for department information**

1 From Mobile Web Studio, select Applications from the Build menu in the left pane, and select Approved under Application Manager.

2 From the list of approved applications that displays in the detail pane, right-click `SSCAMaster` and select Define Events.

The Define Click-Across Events window appears showing a grid with two columns – Dept Name and Dept Head.

3 Click Select to the left of the grid, then click Next.

The Preview window displays.

4 Look for the Assign an Event section of the window, and create a server-side event that associates the cells in column 1 (Dept Name) with the SSCADepartment application. To do so, make these changes (accept defaults for all others):

- Row – enter `2-` (a 2 followed by a dash, no space) to indicate every record from 2 to the last record, excluding the header). Alternatively, you can enter `all` to place the event on the header as well as the records.
- Column – enter `1` to indicate only column 1 (Dept Name) is included.
- Event Name – enter `Dept Name`. This event name is used as the parameter name by the target application, in this case, SSCADepartment.

- With – select “cell value (this cell)” from the drop-down list to indicate that the value associated with the event will be extracted from the cell itself.
- Multi-value – leave unselected.
- Client-side – make sure this option is unselected to indicate the event is a server-side event. This option acts like a toggle. When the option is selected, it indicates a client-side event; when the option is not selected, it indicates a server-side event.
- Click Find Application to open the Search window. Click Search.

  In the Results pane, select the SSCADepartment application and click Add. This forms the event association between the SSCAMaster application and the SSCADepartment application.
- The Search window closes, and the application details are displayed.
- In the Assign An Event section, click Add. The values in column 1 are highlighted, and the event definition displays under Current Assigned Events.

5 Create a server-side event that associates the cells in column 2 (Dept Head) with the SSCAEmployee application. Under Assign Event For, complete these options:

- Row – enter `all`. This places the event on the header as well as the records; alternatively you can enter `2-` (a 2 followed by a dash, no space) to indicate every record from 2 to the last record, excluding the header).
- Column – enter `2` to indicate only column 2 (Dept Head) is included.
- Event Name – enter `Emp Name`. This event name is used as the parameter name by the target application, in this case SSCAEmployee.
- With – select “cell value (this cell)” to indicate that the value associated with the event is extracted from the cell itself.
- Multi-value – leave unselected.

- Client-side – the option is unselected to indicate the event is a server-side event. This option acts like a toggle. When the option is selected, it indicates a client-side event; when the option is not selected, it indicates a server-side event.

Note For mobile applications, you must always create a server-side event.

- Click Find Application to open the Search window. Click Search.

  In the Results pane, select the SSCAEmployee application and click Add.

- The Search window closes, and the application details are displayed.
- In the Assign Event section, click Add. The values in column 2 are highlighted, and the event definition displays under Current Assigned Events.

6 Click Next.

7 From the Preview window, click Finish.

8 To preview the multipage application, select Approved under the Application Manager Status menu, select the SSCAMaster application in the detail pane, and click the Preview button.

  You can try either link:

  - Under Dept Head, select an employee, such as Judy Snow. Judy's employee information displays. In Emp Name, type in an employee, such as Mary Anne Shea and click OK to see Mary Anne Shea's information. Then close the window.
  - Under Dept Name, select a department link, such as Finance. Employees in the Finance department display. In Dept Name, type another department, such as Sales, and click OK to see employees in the Sales department. Then close the window.

9 Close the window to exit.

Creating a link to employee information

In this procedure, create an event that links employee information with the SSCAMaster application.

❖ **Creating an event for employee information**

In this procedure, create an event that links employee information with the SSCAMaster application.

1 From Mobile Web Studio, select Applications from the Build menu in the left pane, and select Approved under Application Manager.

2 From the list of approved applications that displays in the detail pane, right-click SSCADepartment and select Define Events.

The Define Click Across Events window appears showing a grid with one column – Employee.

3 Click Select to the left of the grid, then click Next. The Preview window displays.

4 Look for the Assign An Event section of the window, and create a server-side event that associates the cells in column 1 (Employee) with the SSCADepartment application. To do so, make these changes (accept the defaults for all others):

- Row – enter `2-` (a 2 followed by a dash, no space) to indicate every record from 2 to the last record, excluding the header). Alternatively, you can enter `all` to place the event on the header as well as the records.
- Column – enter `1` to indicate column 1 (Employee) is included.
- Event Name – enter `Emp Name`. This event name is used as the parameter name by the target application, in this case, SSCADepartment.
- With – select "cell value (this cell)" from the drop-down list to indicate that the value associated with the event is extracted from the cell itself.
- Multi-value – leave unselected.
- Client-side – this option is unselected to indicate the event is a server-side event. This option acts like a toggle. When the option is selected, it indicates a client-side event; when the option is not selected, it indicates a server-side event.

Note For mobile applications, you must always create a server-side event.

- Click Find application to open the Search window. Click Search.

- In the Results pane, select the SSCAEmployee application and click Add.

  The Search window closes, and SSCAEmployee is added to the Application field.

- In the Assign An Event section, click Add. The values in column 1 are highlighted, and the event definition displays under Current Assigned Events.

5 Click Next.

6 From the Preview window, click Finish.

7 To preview the application, select Approved under the Application Manager Status menu, select the SSCAMaster application in the detail pane, and click the Preview button.

- Click one of the links under Dept Name to view a list of employees in the department. For example, click Shipping to view the employees in the Shipping department.
- Click one of the links under Employee to view details about the selected employee, for example Jeannette Bertrand.
- In Employee Name, enter the manager's name, in this case Jose Martinez, and click OK to view details about her manager, Jose.
- Close the window.

CHAPTER 4

Creating a Multipage Mobile Application with Transaction Support

Overview

This chapter describes how to create mobile applications that provides transaction support. You can create a personalization key, which enables the user to personalize the application. You can create mobile applications to view and update data from a mobile device. To support transaction processing, you must create an update application that acts as a holding queue until results can be uploaded from or downloaded to the mobile device.

Mobile applications with transaction support tutorial

In this tutorial, you set up a personalization key so you can sort customers by geographic region, an update application to hold updated data, a customer list application that selects customers from the sampledb database, and an employee list that selects sales representatives from sampledb. You also create a link from the customer list application to the update application, and a listener process that detects changes to a customer identifier.

Once these steps are finished, the tutorial leads you through creating the mobile application to view sales figures for each employee, by region and customer identifier, creating an event for the application, and publishing the event as a mobile application. Finally, you create a mobile application to update values from a mobile device.

The final application enables you to view a list of employees in the sales department with the total sales for each of their customers. Clicking the customer ID shows the customer's name with the sales order of each product. There, the user can update the quantity requested for each of the products ordered.

These tasks are described in these sections:

- "Creating a personalization key" on page 43
- "Creating an update application" on page 43
- "Creating a customer list" on page 47
- "Creating a link between applications" on page 49
- "Creating an employee list" on page 50
- "Creating an event for employeeSales" on page 52

This procedure assumes you:

- Know how to log in to Mobile Web Studio
- Know how to create and approve an application in Mobile Web Studio
- Understand basic SQL syntax
- Have sampledb running
- Have access to one of the following:
  - Portal Interface
  - Mobile device, or desktop simulator

Creating the application components

This section shows how to create the application components.

Creating a personalization key

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 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 files of applications that belong to different users can receive input values based on each user.

To use personalization adapters, you must first create new key values for the adapter you want to use, and then create an application that provides input values using that key. This section shows how to create the personalization key "region," which enable the user to select a specific region—Western or Central—for which to display data.

❖ **Creating the personalization key**

1 From Mobile Web Studio, select Manage | Personalization in the left pane, select "database table" in the Personalize Manager pane, and click New.

2 In Create New Key:

   a Name – enter `region`.

   b Type – from the drop-down list, choose Select.

   c Accept the defaults for all other options.

   d Click OK to save.

3 Click OK in the "Save Personalized Key successful" pop-up window. The New personalization key appears in the Key Name detail pane.

Creating an update application

This section shows how to create an application that updates the database.

Note This tutorial demonstrates using connection cache to sampledb when creating database elements. An alternate approach is to use a direct JDBC connection to sampledb, as demonstrated in "Multipage mobile application tutorial" on page 26.

❖ **Creating the update application**

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click New. The Application Builder launches.

2 Create a database element:

a Click the down arrow to the right of Add, and select Database Element.

b On the Database Element Definition window, make sure the Connection Cache option is selected.

c In Conn Cache Name, select `sampledb` from the drop-down list.

d In SQL Query String, enter this query for update (you can copy and paste this code from an electronic source):

```
update sales_order_items set quantity =
@OP["quantity"="13"]where id = @OP["id"="-1"]
and prod_id = @OP["prod_id"="-1"]
```

Note This SQL code updates the database, so no preview is available.

e Click Next. The Define window displays.

f Click Next. The Filter window displays.

g Click Next. The Configure Parameters window displays, with a list of the parameters used to create the database table.

h On the Configure Parameters window, make these modifications (accept the defaults for all others):

- Quantity – click the Variable box, and from the Kind drop-down list, select Update.
- Click the Variable box next to the other two parameters and accept the defaults.

i Click Next. The Window Preview window displays.

j In Element Name, enter `updateOrder`, and click Finish.

3 In Application Builder, click Save.

4 In the Finish window, click the Data Validation tab and make these changes to set up data validation for the Quantity field:

a Common Expressions – select Number Only from the drop-down list. The expression `^[0-9]*$` displays in the Validation Expression box.

b Click Test to test the expression you entered. The Regular Expression Tester window displays.

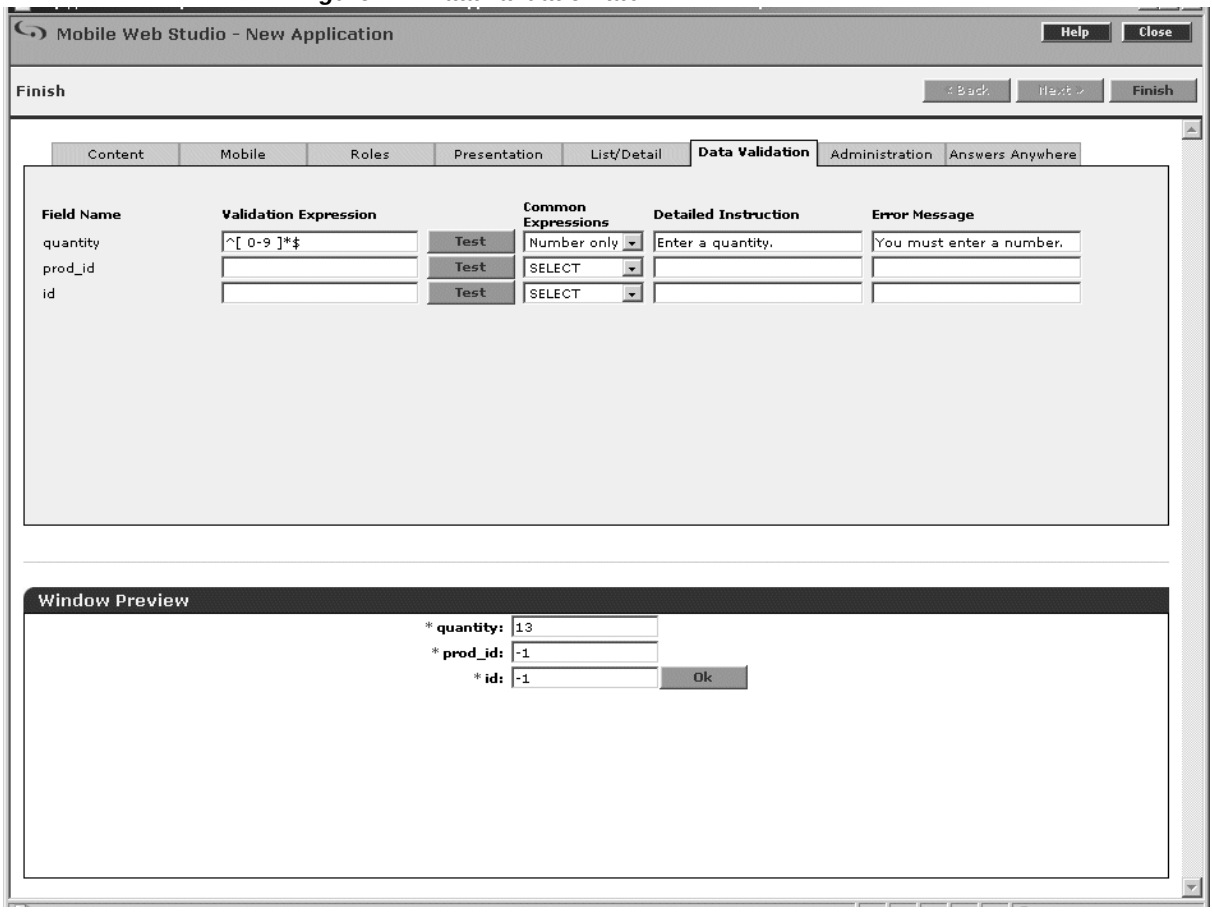
c In Test Value, enter `10`, then click Test. The Test Result field displays Valid.

d Click Save to close the window.

e Detailed Instructions – optionally enter instructions for the user, such as `Enter the quantity`.

f Error Message – optionally enter a brief error message, such as `You must enter a number`.

Figure 4-1: Data validation tab



5 Click Finish to save the application.

6 Click OK in the confirmation pop-up window.

7 Click Close to exit the Application Builder.

8 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The updateOrder application displays in the detail pane.

9 In Mobile Web Studio, right-click the updateOrder application in the detail pane, and select Approval Status | Approved.

Click OK.

10 Select Approved from the Application Manager Status menu. You see the newly approved updateOrder application in the detail pane.

Creating a customer list

This section shows how to create the customer list application. You will create another database element using the same sampledb connection, but with a different SQL query. You will also link the customer list to the update application, and establish a listener for the cust_id field.

❖ Creating the customer list

1 In Mobile Web Studio, select Applications in the left pane, New in the Application Builder Status menu, and click the New button.

2 In Application Builder, click the down arrow to the right of Add, and select Database Element.

3 On the Database Element Definition window, define the database element:

a Make sure the Connection Cache option is selected.

b In Conn Cache Name, select sampledb from the drop-down list.

c In SQL query string, enter this query:

```
set rowcount 10
select c.fname as customer_fname , c.lname as customer_lname, si.id,
si.prod_id, p.name, si.quantity
from customer c, sales_order s, sales_order_items si, product p
where c.id =@OP["cust_id"="101"] and s.id = si.id and si.prod_id =p.id
and s.region ='@OP["region"="Western"]'
set rowcount 0
```

d Click Preview to make sure these columns are included: customer_fname, customer_lname, id, prod_id, name, and quantity.

4 Click Next. The Define window displays.

5 Use the Define window to identify record 1 as the header row. In the Define Record Layout section, click “Records contain labels.”

The “Labels are displayed in Record” option displays. Accept the default, 1, and click Next.

6 In the Filter window, click Next. The Configure Parameters window displays.

7 On the Configure Parameters window, make these modifications (accept the defaults for all others):

- Region – click the Variable box and make these changes:

  Default Value – enter `Western,Central` (no spaces).

  Type – choose Select from the drop-down list, and click Add. The Edit Default Value window displays.

  - In the first row, Western displays for both Display Name and Value.
  - In the second row, Central displays for both Display Name and Value.
  - Click OK to accept the values, and close the window.

  Personalize – choose "database table" from the drop-down list.

  Key – `region` displays as the personalization key (you created the region key in "Creating a personalization key" on page 43). If a different key displays, select "region" from the drop-down list.

- Cust_id – click the Variable box and accept the defaults:

  Default Value – accept `101` as the default value for this parameter.

  Type – accept Text Field from the drop-down list.

8 Click Next. The Window Preview window displays.

9 In Element Name, enter `customerOrder` (no spaces), and click Finish.

10 In Application Builder, click Save.

11 In the Finish window, accept the defaults, and click Finish to save the application.

12 Click OK to confirm.

13 Click Close to exit the Application Builder.

14 When you return to the Mobile Web Studio main window, select New from the Application Manager Status menu. The customerOrder application displays in the detail pane.

15 In Mobile Web Studio, right-click the customerOrder application in the detail pane, and select Approval Status | Approved.

Click OK to confirm.

16 Select Approved from the Application Manager Status menu. You see your newly approved customerOrder application in the detail pane.

17 Optionally, select customerOrder, and click Preview.

In region, select Central from the drop-down list and click OK. The first customer ID value changes from 2020 to 2006.

In cust_id, enter 102 and click OK. The customer name changes from Michaels to Beth. (Other customer IDs include `103`, `104`, and so forth).

Creating a link between applications

This section shows how to link the update application (updateOrder) with the customer order application (customerOrder). This link enables you to update or edit some of the values in the customerOrder list when it is deployed to a mobile device. In this case, the Quantity variable is set to “update” in updateOrder, so the Quantity column can be updated.

❖ **Creating a link to the update application**

1 In Mobile Web Studio, select Applications in the left pane, and Approved in the Application Builder Status menu.

2 Select customerOrder in the detail pane and click Edit.

3 In Application Builder, click Properties. The Properties editor displays.

4 In Properties editor, make these changes; accept the defaults for all other fields:

Mobile tab Make this entry:

- Linked Applications – identify the application to which to link by clicking the Add button to launch the Find Application window.

  On the Find Application window, click Search to locate and select updateOrder, and click Add. The Find Application window closes and updateOrder displays under Linked Applications.

  Under Action type, select Update from the drop-down list.

Click OK to save, and close the Properties Editor.

5 On the Application Builder window, click Save to save the changes, and OK to confirm.

6 Click Close to close the Application Builder.

Creating an employee list

This shows how to create an employee list application. The employee list shows total sales figures for each sales representatives using information in sampledb.

❖ **Creating the employee list of sales reps**

1 In Mobile Web Studio, select Applications in the left pane, New in the Application Builder Status menu, and click the New button.

2 In Application Builder, click the down arrow to the right of Add, and select Database Element.

3 On the Database Element Definition window, define the database element:

a Select the Connection Cache option.

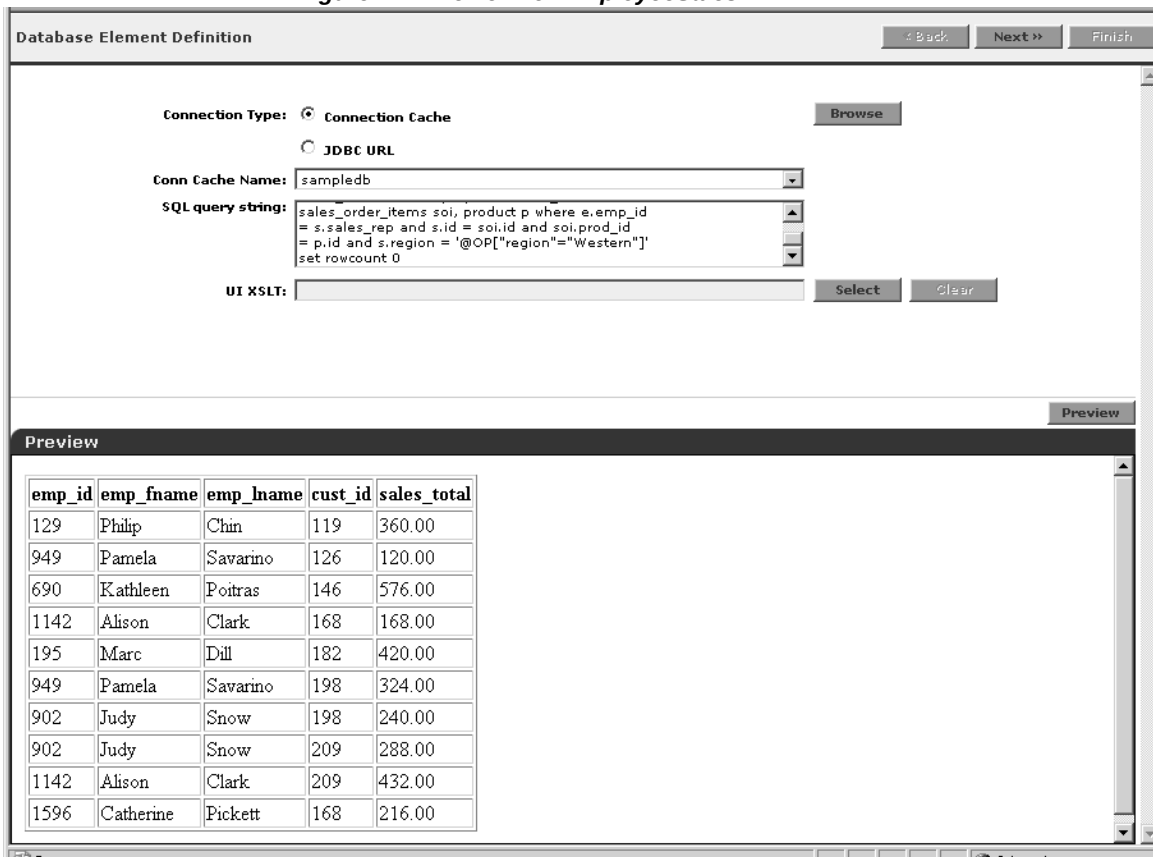
b In Conn Cache Name, select sampledb from the drop-down list.

c In SQL query string, enter this query:

```
set rowcount 10
select e.emp_id, e.emp_fname, e.emp_lname,
s.cust_id,  (soi.quantity * p.unit_price) as
sales_total from employee e, sales_order s,
sales_order_items soi, product p where e.emp_id
= s.sales_rep and s.id = soi.id and soi.prod_id
= p.id and s.region = '@OP["region"="Western"]'
set rowcount 0
```

d Click Preview and make sure these columns are included: emp_id, emp_fname, emp_lname, cust_id, and sales_total.

Figure 4-2: Preview for EmployeeSales



emp_id	emp_fname	emp_lname	cust_id	sales_total
129	Philip	Chin	119	360.00
949	Pamela	Savarino	126	120.00
690	Kathleen	Poitras	146	576.00
1142	Alison	Clark	168	168.00
195	Marc	Dill	182	420.00
949	Pamela	Savarino	198	324.00
902	Judy	Snow	198	240.00
902	Judy	Snow	209	288.00
1142	Alison	Clark	209	432.00
1596	Catherine	Pickett	168	216.00

4 Click Next. The Define window displays.

5 On the Define window, select “Records Contain Labels,” in “Labels are Displayed in Record” accept the default `1`.

6 Click Next. The Filter window displays.

7 Click Next. The Configure Parameters window displays.

8 On the Configure Parameters window, make these modifications (otherwise accept the defaults):

- Region – click the Variable box, and make these changes:
- Type – choose “Select” from the drop-down list and click Add. The Edit Default Value window displays.

  In the first row, Western displays for both Display Name and Value.

Click OK to accept the values and close the window.

- Personalize – choose “database table” from the drop-down list.

  Key – `region` displays as the personalization key (you created region in “Creating a personalization key” on page 43). If a different key displays, select “region” from the drop-down list.

9 Click Next. The Window Preview displays.

10 In Element Name, enter `employeeSales`, and click Finish.

11 In Application Builder, click Save to create the application.

12 On the Finish window, make these changes (otherwise accept the defaults):

Content tab Select 15 minutes for the Content Cache Interval. The Content Cache Interval determines how frequently the UA server retrieves data from the back-end system (in this case the sampledb database).

Mobile tab Select “Make Available to Offline Devices,” then click Finish.

13 Click OK to confirm.

14 Click Close to close Application Builder.

15 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The employeeSales application displays in the detail pane.

16 In Mobile Web Studio, right-click the employeeSales application in the detail pane, and select Approval Status | Approved.

Click OK to confirm.

17 Select Approved from the Application Manager Status menu. You see your newly approved employeeSales application in the detail pane.

Creating an event for employeeSales

In this step, create an event for the customerOrder application. The event links the cust_id field in employeeSales to the customerOrder application. The employeeSales application is the master application, and the customerOrder application is the linked, or child, application.

❖ **Creating an event for employee sales**

1. In Mobile Web Studio, select Applications in the left pane, and Approved in the Application Builder Status menu.
2. From Application Builder, right-click employeeSales in the detail pane, and select Define Events.
3. On the Define Click-Across Events window, click Select next to the grid format, and click Next. The Preview window displays.
4. In Preview, link cust_id in column 4 to the customerOrder application. Under Assign An Event:
   a. Row – enter `2-` (a 2 followed by a dash, no space) to indicate every record from 2 to the last record, excluding the header). Alternatively, you can enter `all`, to place the event on the header as well as the records.
   b. Column – enter `4` to indicate column 4 (cust_id).
   c. Event Name – enter `cust_id` as the event name.
   d. Multi-value – unselect this option.
   e. Client-side – unselect this option to indicate this is a server-side click-across event.
   f. Click Find Application. In the Search window, click Search and select the customerOrder application, then click Add. The window closes and customerOrder displays in the Application Name field.
   g. In Assign An Event, click Add. Column 4 is now selected and a new rule is added under Current Assigned Events.
5. Click Next. The Preview window displays.
6. Click Finish.

Publishing and viewing employeeSales

This section shows how to deploy the employeeSales application to Portal Interface. Portal Interface may be a useful testing and debugging tool. The same mechanism used to deploy to Portal Interface is used to deploy to mobile devices with Pocket Internet Explorer.

- If you are deploying to mobile devices using M-Business Anywhere server, see Chapter 10, “Deploying Applications to Mobile Devices” for information and additional tutorials.

Note To export an application to M-Business Anywhere, the application can contain only one element; multiple-element applications are not compatible as mobile applications.

- If you are deploying to BlackBerry devices using BlackBerry Enterprise Server, instead of M-Business Anywhere server, see Chapter 11, “Deploying Applications to BlackBerry Devices” for information and additional tutorials.

Deploying the mobile application

This section shows how to deploy the employeeSales application to the Portal Interface.

❖ **Deploying the application to Portal Interface**

1 From Mobile Web Studio, click Pages in the Build menu and Approved in the Page Manager Status menu. The Page Manager window displays.

2 In Page Manager, add the employeeSales application to the DefaultPage:

a In Page Builder, select DefaultPage.

b Click Edit, click Add, click Search, select employeeSales, and click Add. The employeeSales application is listed under the Welcome application. The page layout is 50/50.

c Click Save in the upper-left corner, and click OK to confirm.

d Click Close to close Page Builder.

e In the list of Approved pages, right-click the DefaultPage and select Update Users. Click OK in the next two confirmation pop-ups.

The application is now available when you log in to the Portal Interface, or available for download to mobile device.

Testing the employeeSales application

In this step, try out the employeeSales application from the Portal Interface. The application enables you to view sales figures for each sales representative, by region and customer identifier.

❖ **Using the mobile application in Portal Interface**

1. Access Portal Interface by opening another browser session and navigating to:

   `http://`*hostname.domain:port*`/onepage/mpindex.jsp`

2. Click Join Now to set up a new account. See the *Portal Interface User's Guide* for information about Portal Interface and accounts.
3. Log in using the new account. The DefaultPage displays. You see the Welcome window and the employeeSales application.
4. Try out the employeeSales application. For convenience, maximize the window. By default, "Western" is selected in the region drop-down list.
   a. With Western selected for the region, click OK. Sales representatives in the Western region display. You can see their sales figures.
   b. Click the 119 link to display information about one of Philip Chin's customers. Customer information displays for Thomm Smith.
   c. In cust_id, enter `126` and click OK to see customer information for Sam Ovar.
   d. Select "Central" from the region drop-down list, enter `110 in cust_id`, and click OK. Data for Michael Agliori displays.
5. Personalize the region parameter to display only customers in the Central region:
   a. Click My Info, and select the Personalize tab.
   b. Region key – enter `Central`.
   c. Click Done.
6. Try out the employeeSales application again:
   a. Click the Sybase logo in the upper-left corner to return to the employeeSales application.
   b. Click the region drop-down list and notice that only Central is included.

c Click Log Out in the upper-right corner when you are finished. The Portal Login window displays.

Setting up employeeSales for automated push

There are two ways to configure this application so that your offline mobile users will automatically get updates as the database changes:

- An administrator can use the Mobile Studio to assign an application to a registered mobile user. See the *Unwired Accelerator Administration Guide* for information.
- An individual user can subscribe to the application and modify their subscription to enable push notification of changes.

End users with the UA mobile offline client installed on their mobile devices can view available mobile applications, subscribe to them, and manage their subscriptions from their mobile devices. This tutorial shows how this is done from a BlackBerry device, but the same basic process applies to all of the various mobile device clients.

❖ **Self-registration for push notification on an application**

For push synchronization to work for BlackBerry devices, you must configure the device so that it can receive e-mail or SMS messages. This tutorial assumes e-mail is the notification protocol.

1 The Unwired Accelerator BlackBerry client must be running on the device (the UA icon should appear in the application menu).

2 Use the trackwheel to navigate to the Settings menu.

3 Scroll to the Email Address item and open it.

4 Enter an e-mail address for which the BlackBerry receives mail, and click Save.

5 Return to the main form of UA Application, and use the trackwheel to select the Profiles | New Profile menu item.

6 Fill out the form with your user name, password, UA server address, and port number. Save your profile.

When you save your profile, the UA application attempts to verify the UA server address data by connecting to the server. If successful, the device's information is registered with the UA server and an administrator will be able to see your user profile from Studio under Manage | Users/Roles | Mobile Users.

7 Navigate back to the UA main menu, and choose Refresh All Apps to retrieve a list of all available mobile applications from this server.

8 Click the EmployeeSales checkbox and select Refresh App from the menu.

The application and its data are synchronized. Several hundred rows of data may be returned, so this process could take some time.

9 After the initial synchronization finishes, select the employeeSales application and use the menu to enter Application Info. The Application Info form allows you to edit how you are subscribed to this application.

10 Click on the Push Enabled checkbox to configure this application to be automatically updated to your device as the application data changes.

From this form you can also control the minimum update frequency in minutes. You can also control what happens when an update notification arrives at your device for this application. It can either re-synchronize the application automatically, or display a dialog allowing you to choose whether to synchronize or not; or you can ignore the synchronization completely.

11 Use the Save menu to save your changes to this application.

12 Perform a Refresh App on the changed application to update the UA server with this change in the application subscription. After this, the UA server sends push notificatons to the client when the application data changes.

Using push agents

The first time any user obtains a push-subscription for an application, the UA server will automatically create an agent to monitor this application for changes

❖ **Building the agent**

1 In Mobile Web Studio, select Automate | Agents | Ready (or Stopped) item to see the poll-employeeSales agent that has been created.

2 Click the Edit button. In the Agent Builder, on the Schedule tab you'll see that the Recurrence setting matches the Content Cache Interval of 15 minutes for the employeeSales application.

3 On the Action tab, click Deliver, then click the Edit button.

4 In the Action Details window:

- Description – enter a description for the action.
- Destination Viewer – select PDA. The BlackBerry device is considered a PDA in this instance.
- User List Adapter – select Push Registered Users.

Click OK.

5 In the Agent Builder window, click Save.

6 Click OK.

Now, each time the agent executes (every 15 minutes), it retrieves a list of all the users who are subscribed for push notifications on this application, and executes the application as each of these users in turn check to see if the data has changed. Whenever the data changes, UA sends a notification only to those users who are monitoring the same regions.

Note Using the Push-Registered Users User-List feature can cause performance problems with the UA server if the frequency of the agent is high, or the number of users is large. It is recommended that if you are going to create push subscriptions for an application that you do *not* use autofill adapters to personalize the application. That way the agent can simply execute the application once each time it runs, in the context of a default user, and generate notifications to all subscribers as data changes.

CHAPTER 5

Creating a Custom .NET Client

Overview

This tutorial demonstrates how to develop a custom Unwired Accelerator .NET Client for Windows Mobile Pocket PCs using the sales order sample data that is contained in the sample database sampledb, which is included with Unwired Accelerator. The client application built throughout this tutorial is written in C#. Since the client application and the Unwired Accelerator APIs are .NET assemblies, you can also create this client application using Visual Basic .NET.

This tutorial assumes:

- You have experience with the Microsoft .NET Framework or .NET Compact Framework.
- You have experience using the Visual Studio .NET 2003 development environment.
- You have knowledge of either C# or Visual Basic .NET.
- You have knowledge of Unwired Accelerator topics such as Server Side Click Across, Data Validation, and so on.

Requirements

This tutorial requires the following products to be installed:

- Microsoft Visual Studio .NET 2003 Professional (or a higher edition such as Enterprise Developer or Enterprise Architect) is installed. You can also use Visual Studio 2005 with this tutorial, but some of the steps will differ.
- Windows Mobile 2003 Pocket PC SDK.
- ActiveSync version 4.1 or higher.
- The Unwired Accelerator .NET API assembly files and the UltraLite schema and database files that are included with the Unwired Accelerator at `http://host.domain:port/onepage/dotnet/UA.API-bin.zip`.

  For example, if your machine name is "lab2k," your domain is "sybase.com," and your port number is 4040, enter:

```
http://lab2k.sybase.com:4040/onepage/dotnet/UA.API-bin.zip
```

Creating a .NET application

Using the sample data from sampledb, you will build six applications using the database element, then mobilize the applications to the custom client. The sampledb database contains sample sales order, sales order line items, employee, customer, and product information.

The database tables that are used include:

- sales_order – contains sales order information such as order ID, date, customer ID, sales representative ID (which is an employee ID from the employee table, the employee table is not mobilized or used in this tutorial).
- sales_order_items – contains line item information for the sales orders such as product ID and quantity.
- product – contains product information such as product ID, description and price.
- customer – contains customer information such as customer ID, name and contact information.

Creating the personalization key

Before creating the database applications, you must create a personalization key called XYZSalesRepID. For information about personalization keys, see “Creating a personalization key” on page 43.

❖ **Creating the XYZSalesRepID key**

1. From Mobile Web Studio, select Personalization in the left pane, select “database table” in the Personalize Manager pane, and click the New button.
2. In the Create New Key window:
   a. Name – enter `XYZSalesRepID`.
   b. Type – from the drop-down list, choose Text.
   c. Accept the defaults for all other options.
   d. Click OK to save.
3. Click OK in the Save Personalized Key successful window. The new personalization key appears in the Key Name detail pane.

❖ **Adding the personalization key to your account**

1. In the top Mobile Web Studio menu, click Account Info.
2. In the Account Info window, click Personalize.
3. In the Personalization window, click Add.
4. In the Add Personalization Keys window, click Search.
5. In the Results pane, select XYZSalesRepID, and click OK.
6. In the Personalization window, you see the new XYZSalesRepID personalization key listed. In the Value text field, enter `299`.

   Click OK to save the changes.
7. Click OK in the confirmation pop-up window.

Creating the database applications

The procedures for building the six database applications are described in these sections:

- “Building the SOList application” on page 62

- “Building the SOItemList application” on page 63
- “Building the CustomerList application” on page 64
- “Building the ProductList application” on page 66
- “Building the SOCreate application” on page 67
- “Building the SOItemCreate application” on page 69

After building the database applications, you will create an event using the Event wizard to link the SOList application with the SOItemList application as shown in “Creating an event for SOList” on page 71.

❖ Building the SOList application

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.

2 In the Application Builder, click the down arrow to the right of Add, and select Database Element.

3 On the Database Element Definition window, make sure the Connection Cache option is selected.

Enter:

- Connection Cache Name – select sampledb from the drop-down list.
- SQL Query String –

```
select id as SalesOrderID, cust_id as CustomerID,
left(order_date, 10) as OrderDate, region as
Region from sales_order where sales_rep =
@OP["SalesRepID"="299"]
```

- UI XSLT – leave this field empty.

Click Next.

4 Use the Define window to identify record 1 as the header row. In the Define Record Layout section, select Rows and click “Records contain labels.” Accept the `1` in “Labels are displayed in Record,” then click Next.

5 In the Filter window, click Next.

6 In the Configure Parameters window:

a Under Variable, select the box next to SalesRepID.

b Under Personalize, select Database Table from the drop-down list.

c Under Key, select XYZSalesRepID.

d Click Next.

7 In the Preview window, in Element Name, enter `SOList`.

Click Finish.

8 In the Application Builder, click Save.

9 On the Finish window, make these entries (otherwise accept the defaults):

Content tab In Name, enter `SOList`.

Roles tab Click Add All to add all roles to this application.

10 Click Finish to save the application, and click OK to confirm.

11 Click Close in the upper-right corner to close the Application Builder window.

12 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The `SOList` application displays in the detail pane.

13 In Mobile Web Studio, right-click the `SOList` application in the detail pane, and select Approval Status | Approved.

Click OK.

14 Select Approved from the Application Manager Status menu. You see the newly approved `SOList` application in the detail pane.

❖ **Building the SOItemList application**

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.

2 In the Application Builder, click the down arrow to the right of Add, and select Database Element.

3 On the Database Element Definition window, make sure the Connection Cache option is selected.

Enter:

- Connection Cache Name – select sampledb from the drop-down list.
- SQL Query String –

```
select id as SalesOrderID, line_id as LineItemID,
prod_id as ProductID, quantity as Quantity,
```

```
left(ship_date, 10) as ShipDate from
sales_order_items
where SalesOrderID = @OP["SalesOrderID"="2001"]
```

- UI XSLT – leave this field empty.

Click Next.

4 Use the Define window to identify record 1 as the header row. In the Define Record Layout section, select Rows and click "Records contain labels. Accept the `1` in the Labels are displayed in Record option, then click Next.

5 In the Filter window, click Next.

6 In the Configure Parameters window, under Variable select the checkbox next to SalesOrderID.

7 In the Preview window, in Element Name, enter `SOItemList`.

Click Finish.

8 In the Application Builder, click Save.

9 On the Finish window, make these entries (otherwise accept the defaults):

Content tab In Name, enter `SOItemList`.

Roles tab Click Add All to add all roles to this application.

10 Click Finish to save the application, and click OK to confirm.

11 Click Close in the upper-right corner to close the Application Builder window.

12 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The `SOItemList` application displays in the detail pane.

13 In Mobile Web Studio, right-click the `SOItemList` application in the detail pane, and select Approval Status | Approved.

Click OK.

14 Select Approved from the Application Manager Status menu. You see the newly approved `SOItemList` application in the detail pane.

❖ **Building the CustomerList application**

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.

2 In the Application Builder, click the down arrow to the right of Add, and select Database Element.

3 On the Database Element Definition window, make sure the Connection Cache option is selected.

Enter:

- Connection Cache Name – select sampledb from the drop-down list.
- SQL Query String –

```
select id as CustomerID, fname as FirstName,
lname as LastName, address as Address, city as
City, state as State, zip as PostalCode, phone as
Telephone, company_name as CompanyName from
customer
```

- UI XSLT – leave this field empty.

Click Next.

4 Use the Define window to identify record 1 as the header row. In the Define Record Layout section, select Rows and click "Records contain labels." Enter `1` in "Labels are displayed in Record," then click Next.

5 In the Filter window, click Next.

6 In the Preview window, in Element Name, enter `CustomerList`.

Click Finish.

7 In the Application Builder, click Save.

8 On the Finish window, make these entries (otherwise accept the defaults):

Content tab In Name, enter `CustomerList`.

Roles tab Click Add All to add all roles to this application.

9 Click Finish to save the application, and click OK to confirm.

10 Click Close in the upper-right corner to close the Application Builder window.

11 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The `CustomerList` application displays in the detail pane.

12 In Mobile Web Studio, right-click the `CustomerList` application in the detail pane, and select Approval Status | Approved.

Click OK.

13 Select Approved from the Application Manager Status menu. You see the newly approved `CustomerList` application in the detail pane.

❖ **Building the ProductList application**

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.

2 In the Application Builder, click the down arrow to the right of Add, and select Database Element.

3 On the Database Element Definition window, make sure the Connection Cache option is selected.

Enter:

- Connection Cache Name – select sampledb from the drop-down list.
- SQL Query String –

```
select id as ProductID, name as Name, description
as Description, prod_size as Size, color as
Color, unit_price as UnitPrice from product
```

- UI XSLT – leave this field empty.

Click Next.

4 Use the Define window to identify record 1 as the header row. In the Define Record Layout section, select Rows and click "Records contain labels." Enter `1` in "Labels are displayed in Record," then click Next.

5 In the Filter window, click Next.

6 In the Preview window, in Element Name, enter `ProductList`.

Click Finish.

7 In the Application Builder, click Save.

8 On the Finish window, make these entries (otherwise accept the defaults):

Content tab In Name, enter `ProductList`.

Roles tab Click Add All to add all roles to this application.

9 Click Finish to save the application, and click OK to confirm.

10 Click Close in the upper-right corner to close the Application Builder window.

11 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The `ProductList` application displays in the detail pane.

12 In Mobile Web Studio, right-click the `ProductList` application in the detail pane, and select Approval Status | Approved.

Click OK.

13 Select Approved from the Application Manager Status menu. You see the newly approved `ProductList` application in the detail pane.

❖ **Building the SOCreate application**

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.

2 In the Application Builder, click the down arrow to the right of Add, and select Database Element.

3 On the Database Element Definition window, make sure the Connection Cache option is selected.

Enter:

- Connection Cache Name – select sampledb from the drop-down list.
- SQL Query String –

```
begin
declare @SalesOrderID int;
select max("id")+1 into @SalesOrderID from
sales_order;
if @SalesOrderID is null then
select 1 into @SalesOrderID;
end if;
insert into sales_order ("id", cust_id,
order_date, fin_code_id, region, sales_rep)
values (@SalesOrderID, @OP["CustomerID"="101"],
'@OP["OrderDate"="2006-01-01"]', 'r1',
'@OP["Region"="Eastern"]',
@OP["SalesRepID"="299"]);
select @SalesOrderID as SalesOrderID;
end
```

- UI XSLT – leave this field empty.

Click Next.

4 Use the Define window to identify record 1 as the header row. In the Define Record Layout section, select Rows and click “Records contain labels.” Enter `1` in “Labels are displayed in Record option,” then click Next.

5 In the Filter window, click Next.

6 In the Configure Parameters window:

   a Select all the parameters.

   b For the SalesRepID parameter, under Personalize, select Database Table from the drop-down list.

   c Under Key, select XYZSalesRepID.

   d Change the Kind for all parameters to Update.

   e Accept all values in the Default fields.

   f Click Next.

7 In the Preview window, in Element Name, enter `SOCreate`.

   Click Finish.

8 In the Application Builder, click Save.

9 On the Finish window, make these entries (otherwise accept the defaults):

   Content tab In Name, enter `SOCreate`.

   Roles tab Click Add All to add all roles to this application.

   Click OK.

10 Click Close in the upper-right corner to close the Application Builder window.

11 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The `SOCreate` application displays in the detail pane.

12 In Mobile Web Studio, right-click the `SOCreate` application in the detail pane, and select Approval Status | Approved.

   Click OK.

13 Select Approved from the Application Manager Status menu. You see the newly approved `SOCreate` application in the detail pane.

❖ **Building the SOItemCreate application**

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.

2 In the Application Builder, click the down arrow to the right of Add, and select Database Element.

3 On the Database Element Definition window, make sure the Connection Cache option is selected.

Enter:

- Connection Cache Name – select sampledb from the drop-down list.
- SQL Query String –

```
begin
declare @SalesOrderLineItemID int;
select max(line_id)+1 into @SalesOrderLineItemID
from sales_order_items where "id" =
@OP["SalesOrderID"="2001"];
if @SalesOrderLineItemID is null then
select 1 into @SalesOrderLineItemID;
end if;
insert into sales_order_items ("id", line_id,
prod_id, quantity, ship_date) values
(@OP["SalesOrderID"="2001"],
@SalesOrderLineItemID, @OP["ProductID"="300"],
@OP["Quantity"="1"], '@OP["ShipDate"="2006-01-
01"]');
select @SalesOrderLineItemID as
SalesOrderLineItemID;
end
```

- UI XSLT – leave this field empty.

Click Next.

4 Use the Define window to identify record 1 as the header row. In the Define Record Layout section, select Rows and click “Records contain labels.” Enter `1` in “Labels are displayed in Record option,” then click Next.

5 In the Filter window, click Next.

6 In the Configure Parameters window:

a Select all the parameters.

b Change the Kind for all parameters to Update.

c Click Next.

7 In the Preview window, in Element Name, enter `SOItemCreate`.

Click Finish.

8 In the Application Builder, click Save.

9 On the Finish window, make these entries (otherwise accept the defaults):

Content tab In Name, enter `SOItemCreate`.

Roles tab Click Add All to add all roles to this application.

10 Click Finish to save the application, and click OK to confirm.

11 Click Close in the upper-right corner to close the Application Builder window.

12 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The `SOItemCreate` application displays in the detail pane.

13 In Mobile Web Studio, right-click the `SOItemCreate` application in the detail pane, and select Approval Status | Approved.

Click OK.

14 Select Approved from the Application Manager Status menu. You see the newly approved `SOItemCreate` application in the detail pane.

Linking the applications

Once you have created the database applications, you must link the SOCreate and SOCreateItem update applications to the SO List master application.

❖ **Linking the update applications to the master application**

1 In Applications | Approved, select the SOList application, and click Edit.

2 In the Application Builder, click Properties.

3 In the Properties Editor window, click the Mobile tab.

4 Under Linked Applications, click Add.

5 In the Find Application window, click Search.

6 From the Results pane, select the SOCreate application, and click Add.

7 You return to the Properties Editor window, where you see the SOCreate application listed as a linked application. Verfiy that the SOCreate application has these settings:

- Action Type – Insert
- Menu Level – 1

8 Link the SOItemCreate application using the same procedure shown in steps 4 – 6. When you return to the Properties Editor window, verify the SOItemCreate application has these settings:

- Action Type – Insert
- Menu Level – 2

9 Click OK.

10 In the Application Builder, click Save.

11 Click Close in the upper-right corner of the Application Builder.

❖ **Creating an event for SOList**

In this procedure, create an event that links the ID column of the SOList master application to the SOItemList application.

1 From Mobile Web Studio, select Applications from the Build menu in the left pane, and select Approved under Application Manager.

2 From the list of approved applications that displays in the detail pane, right-click SOList and select Define Events.

The Define Click-Across Events window appears showing a grid with four columns.

3 Click Select to the left of the grid, then click Next.

The Preview window displays.

4 Look for the Assign an Event section of the window, and create a server-side event that associates the cells in column 1 (ID) with the SOItemList application. To do so, make these changes (accept defaults for all others):

- Row – enter `all` to place the event on the header as well as the records.
- Column – enter `1` to indicate only column 1 (ID) is included.
- Event Name – enter `SalesOrderID`. This event name is used as the parameter name by the target application.

- Multi-value – leave unselected.
- Client-side – leave unselected.
- Click Find Application to open the Search window. Click Search.
- In the Results pane, select the SOItemList application and click Add. This forms the event association between the SOList application and the SOItemList application.

  The Search window closes, and the application details are displayed.
- In the Assign An Event section, click Add. The values in column 1 are highlighted, and the event definition displays under Current Assigned Events.

5 Click Next.

6 From the Preview window, click Finish.

7 Close the window to exit.

Deploying the applications to MobiLink

After creating all the database applications, you must deploy the SOList, CustomerList, and ProductList applications to MobiLink.

1 In Applications | Approved, select the SOList application.

2 Right-click the SOList application and select Deploy to MobiLink.

3 Click OK in the confirmation pop-up window.

4 Repeat the same steps for the CustomerList and ProductList applications.

Creating the custom .NET client

A sample implementation of this project is available as a reference for following along with this tutorial. The sample implementation is located at `http://`***`host`***`.`***`domain`***`:`***`port`***`/onepage/apps/xyzsales/XYZSales.zip`.

1 Launch Visual Studio .NET 2003 and start the New Project wizard.

2 Select Visual C# Projects as the Project Type, and select the Smart Device Application template.

3 Enter `XYZSales` as the project name.

4 Set the Location for where the project will be saved.

5 Click the More button and enter `XYZSales` as the solution name.

6 Select Create Directory for Solution, and click OK.

7 In the next window, select Pocket PC as the platform to target and Windows Application as the project type. Click OK.

8 The wizard takes a few moments to create the project folders and files at the location specified in the first step of the New Project wizard.

Changing the settings of the project

After you create the project, there are some default settings you must change:

1 Right-click the XYZSales project in the Solution Explorer and select Properties.

2 In the Propeerties window, select Common Properties | General from the left-hand tree navigation menu.

3 In the Application category of the Properties editor, on the right side of the screen, change the Assembly Name field to `XYZ Sales` (there is a space between XYZ and Sales).

4 Open the *Form1.cs* file and search the source code for `Form1`. Replace all instances of the `Form1` string with `FormSalesOrders`.

5 Rename the *Form1.cs* file to *FormSalesOrders.cs*.

6 In the *FormSalesOrders.cs* file, locate and remove the Main function. You will add a new customized Main function to the project later, in a different class.

Now you must add the references to the Unwired Accelerator API assemblies that are included in the *UA.API-bin.zip* archive file:

1 Unzip the *UA.API-bin.zip* file on your *C:* drive.

2 In the Visual Studio .NET Solution Explorer window, from the Option menu, select Add Reference.

3 In the window that appears, on the .NET tab, click Browse.

4 From the unzipped archive, browse to the*UA.API\bin\CF1.0* directory and select these two files:

- *Sybase.UnwiredAccelerator.Commons.dll*
- *Sybase.UnwiredAccelerator.Data.dll*

5 Browse to *UA.API\bin\ultralite\ce* and select *iAnywhere.Data.UltraLite.dll*.

6 Browse to *UA.API\bin\ultralite\ce\en* and select *iAnywhere.Data.UltraLite.resources.dll*.

7 Click OK to add the references to the project.

8 In the Solution Explorer, right-click the project and select Add | Add Existing Item and browse to *UA.API\bin\ultralite\ce\arm*.

9 Change the Files of Type drop-down to All Files.

10 Select ulnet9.dll, click the arrow next to the Open button, and select Link File. Repeat this step for these files:

- *UA.API\bin\base.usm*
- *UA.API\bin\localdb.usm*

11 After linking the files in the previous step, select each file individually in the Solution Explorer and check the Properties window to verify the Build Action property is set to Content for each file.

Now add a new class file to the project named *Program.cs* that contains a class named *Program*. The purpose of the *Program* class is to serve as a main point of entry for the client as well as a single interface for some global management classes that will be added next. See the sample implementation to get the source code of the Program class.

Creating the FormSalesOrders form

The default X button that appears in the upper-right corner of the Pocket PC applications is a minimize button rather than an exit button. In the procedure below, you will first change the default behavior of the client, then set the FormSalesOrders screen to display an OK button, which will terminate the client application when clicked.

❖ **Changing the default close behavior of the client**

1 In the View Designer, open FormSalesOrders and click once on the form.

2 In the Properties window, set the MinimizeBox to False.

3 Add a using directive for both the Sybase.UnwiredAccelerator.Windows.Forms and the Sybase.UnwiredAccelerator.Data assemblies to the FormSalesOrders code.

4 Place an Input Panel control on FormSalesOrders and name the object `sip` (Soft Input Panel, another name for on-screen keyboards) and implement an empty handler function for the EnabledChanged event.

5 Edit FormSalesOrders so it implements the Sybase.UnwiredAccelerator.Windows.Forms.IStackFormsManagerForm interface.

   See the sample implementation of this tutorial for the implementation of the IStackFormsManagerForm interface.

   Note See the Unwired Accelerator .NET API Cookbook and API documentation for more information on the IStackFormsManagerForm interface and its use. Every form in this custom client application implements the IStackFormsManagerForm interface.

6 Add a handler function for the FormSalesOrders Closing event as shown in the sample implementation of this tutorial.

7 Now place a label, a data grid and a combo box on the FormSalesOrders form. Refer to the sample implementation of this tutorial for more detailed information on the placement and settings of these controls.

   The combo box will be used to show the list of customer names. When a name is selected the sales orders for that customer will be populated into the data grid. The label is used to show a Customers: title above the combo box.

Creating the FormSalesOrderDetail form

The FormSalesOrderDetail form is used to show a detailed listing of a specific sales order. This form makes use of the TabControl, where the first tab shows the sales order ID, the sales order date, and the customer information. The second tab contains a DataGrid control that lists the line items of the sales order.

❖ **Creating FormSalesOrderDetail**

1. Start by adding a new Form to the project. Name the Form file *FormSalesOrderDetail.cs*. The class contained in the file is automatically named FormSalesOrderDetail.
2. Set the FormSalesOrderDetails MinimizeBox property to False.
3. Add a using directive for both the Sybase.UnwiredAccelerator.Windows.Forms and the Sybase.UnwiredAccelerator.Data assemblies to the FormSalesOrderDetails code.
4. Place an Input Panel control on FormSalesOrderDetails and name the object "sip." Implement an empty handler function for the EnabledChanged event.
5. Edit FormSalesOrderDetails so it implements the Sybase.UnwiredAccelerator.Windows.Forms.IStackFormsManagerForm interface. See the sample implementation of this tutorial for the implementation of the IStackFormsManagerForm interface.
6. Add a handler function for the FormSalesOrderDetails Closing event as shown in the sample implementation of this tutorial.
7. Place a TabControl, DataGrid, and several Label controls on the form. Organize them as shown in the sample implementation of this tutorial.

Creating the FormNewSalesOrder form

The FormNewSalesOrder form is used to enter a new sales order. This form makes use of the TabControl, where the first tab shows ComboBox controls for selecting the customer, the sales order date, and the sales order region. The second tab contains a DataGrid control that lists the line items of the sales order. There is also a MainMenu control that has MenuItem entries for adding, editing, and deleting line items, as well an entry for saving the sales order.

❖ **Creating FormNewSalesOrder**

1. Add a new Form to the project and name the Form file FormNewSalesOrder.cs. The class contained in the file is automatically named FormNewSalesOrder.
2. Set the FormNewSalesOrder MinimizeBox property to False.

3 Add a using directive for both the Sybase.UnwiredAccelerator.Windows.Forms and the Sybase.UnwiredAccelerator.Data assemblies to the FormNewSalesOrder code.

4 Place an Input Panel control on FormNewSalesOrder and name the object "sip." Implement an empty handler function for the EnabledChanged event.

5 Edit the FormNewSalesOrder form so it implements the Sybase.UnwiredAccelerator.Windows.Forms.IStackFormsManagerForm interface. See the sample implementation of this tutorial for the implementation of the IStackFormsManagerForm interface.

6 Add a handler function for the FormNewSalesOrder Closing event as shown in the sample implementation of this tutorial.

7 Now place a TabControl, DataGrid, several Label controls, and several ComboBoxes on the form and organize them as shown in the sample implementation of this tutorial.

Creating the FormEditSalesOrderItem form

The FormEditSalesOrderItem form is used to enter a new sales order. This form makes use a ComboBox for selecting a product and a TextBox for entering a quantity. There is also a MainMenu control that has MenuItem entries that serve as OK and Cancel buttons to either save the line item to the sales order or cancel the line item entry.

❖ **Creating FormEditSalesOrderItem**

1 Add a new Form to the project and name the Form file *FormEditSalesOrderItem.cs*. The class contained in the file is automatically named FormEditSalesOrderItem.

2 Set the FormEditSalesOrderItem MinimizeBox property to False.

3 Add a using directive for both the Sybase.UnwiredAccelerator.Windows.Forms and the Sybase.UnwiredAccelerator.Data assemblies to the FormEditSalesOrderItem code.

4 Place an Input Panel control on FormEditSalesOrderItem and name the object "sip." Implement an empty handler function for the EnabledChanged event.

5 Edit FormEditSalesOrderItem so it implements the Sybase.UnwiredAccelerator.Windows.Forms.IStackFormsManagerForm interface. See the sample implementation of this tutorial for the implementation of the IStackFormsManagerForm interface.

6 Add a handler function for the FormEditSalesOrderItem Closing event as shown in the sample implementation of this tutorial

7 Now place a ComboBox, TextBox, and a couple of Labels on the form and organize them as shown in the sample implementation of this tutorial.

Adding the remaining files

There are some remaining files that you must add to your project. You can copy these files from the sample implementation of this tutorial and add them directly to your project. These files are:

- *Config.xml* – verify the BuildAction of this file is set to Content
- *ConfigurationManager.cs*
- *DatabaseManager.cs*
- *ListItem.cs*
- *Literals.cs*

Building and deploying the project to a Pocket PC from Visual Studio

This section describes how to build and deploy the project to your PocketPC from Visual Studio.

❖ **Deploying from Visual Studio**

1 Launch Visual Studio.

2 Select Build | Build Solution

3 After the project builds successfully you can deploy the project to a Pocket PC by selecting Build | Deploy Solution.

Building a redistributable CAB file

In Visual Studio 2005, the automatic CAB generation process has been replaced with a new feature called a Smart Device CAB Project. See the MSDN documentation for information about using the Smart Device CAB Project feature.

Visual Studio .NET 2003 has a Build CAB File feature that you can use to automatically generate a redistributable CAB file. This section of the tutorial explains how to use the automatic CAB build process. After using the automatic CAB build process, you must change some of the default settings. For example, the default installation path of the application, the name of the Programs shortcut icon, and the name of the Remove Programs screen entry for the client contain default values such as `My Company`. You will customize the CAB file to reflect the names `XYZ Sales` and `Unwired Accelerator 8.0`.

The process for making these customizations is two-fold. First, you must use the Build CAB File feature to generate an INF file to drive the CAB building process. After the INF file is created, you will customize it and use it to manually rebuild the XYZ Sales CAB file. You must manually generate the CAB after editing the INF file because if you use the Build CAB File feature in Visual Studio again, the INF file is overwritten with the original default settings, and you will lose your customized settings.

❖ **Generating the INF file**

1. Build the project to make sure the latest version of the binary files are compiled and placed in the project's temporary build folder named *obj*. The *obj* folder is automatically created in the project's source code folder when you build the project.
2. Right-click the XYZSales project in the Solution Explorer and select Build CAB File.
3. After a moment, a series of command prompts appear and close. If successful, the Visual Studio Output window displays the following message:

   ```
   " ------ Build Cab File: succeeded -----"
   ```

The next procedure shows how to customize the INF file that you generated in the *obj* folder. You will make the following customizations:

- Installation Path – change to the installation directory of the application, for example *Program Files\Sybase\UA 8\XYZ Sales*.

- Application title in the Remove Programs screen – change to `Sybase UA 8 XYZ Sales`.
- Name of the shortcut in the Programs screen – change to `XYZ Sales`.

❖ **Customizing the INF file**

1. Open the *obj* folder in Windows Explorer. If you are building a Release version, go into the *Release* folder. Similarly, if you are building a Debug version go into the *Debug* folder. If not explicitly set to Release, the project is set, by default, to Debug.
2. From the *Debug* or *Release* folder, locate and open the file named *XYZSales_PPC.inf* with Notepad or any other basic text editor.
3. Search for the line that says `Provider="My Company"` and change it to `Provider="Sybase"`.
4. Locate the line that says `AppName="XYZSales"` and change it to `AppName="UA 8 XYZ Sales"`.
5. Locate the line that says `InstallDir=%CE1%\%AppName%` and change it to `InstallDir=%CE1%\Program Files\Sybase\UA 8\XYZ Sales`.
6. Near the bottom of the INF file, locate the line that says XYZ Sales,0,XYZ Sales.exe,%CE11% and change it to "Sybase UA 8 XYZ Sales,0,XYZ Sales.exe,%CE11%."

   Note A space was placed between "XYZ" and "Sales" in the first part of this comma-delimited string. Do not add a space to the second "XYZSales" because that is the name of the EXE the shortcut will reference.

7. Save the INF file.
8. From a command prompt, navigate to the folder the INF file is in and run the *BuildCab.bat* script file.

   The same command prompts you saw when you built the CAB from within Visual Studio reappear momentarily.
9. After the script is finished running and the command prompts have stopped opening and closing, the newly built CAB file is placed in a folder named *cab*. The *cab* folder is parallel to the *obj* folder under the project's source code folder.

Using Windows Explorer, navigate to the *cab* folder. Depending on the Release/Debug setting you chose, there will be either a Debug or Release folder like there is under the *obj* folder. Go into the folder that is appropriate for the build version you are using.

Once you navigate into the appropriate *Debug* or *Release* folder, you see a set of multiple CAB files for different CPUs. The file for deploying to mobile devices is the CAB named *XYZSales_PPC.ARMV4.CAB*. If you are going to run this client on a Visual Studio 2003 emulator, use the *XYZSales_PPC.X86.CAB* file.

❖ **Deploying the CAB file to your PocketPC**

1. CAB files must be transferred to your PocketPC before they can be installed. Open Microsoft ActiveSync, and open ActiveSync Explorer.
2. When the Mobile Device window appears, navigate to the location of the *XYZSales_PPC.ARMV4.CAB* file.
3. Right-click the *XYZSales_PPC.ARMV4.CAB* file, and select Copy.
4. Return to the Mobile Device explorer window, right-click in the right pane of the window and select Paste. The file is copied to your PocketPC.
5. Use the File Explorer application on the PocketPC to navigate to the location where you copied the file. Once you locate the file, tap it once to start the installation process.
6. The application is added to the Start | Programs menu.

Additional information about building redistributable CAB files

If you add or remove files from the project's file structure, change references, or make any other important changes, you must regenerate the INF file that is used to build the final CAB because the INF contains information about all files that will be installed to the Pocket PC during installation. The only time this INF file updated with the latest list of files is when you use the automated Build CAB File feature in Visual Studio. Since the Build CAB File feature resets the INF file to all its default values, you must edit the INF every time the Build CAB File feature is used. After editing the INF file, you must run *theBuildCab.bat* script again to rebuild the CAB using the modified INF file.

CHAPTER 6

Mobilizing BusinessObjects

Overview

This tutorial shows you how to mobilize a BusinessObjects Web Intelligence report.

It assumes you:

- Know how to log in to Mobile Web Studio.
- Know how to create and approve an application in Mobile Web Studio.
- Are familiar with administration of BusinessObjects
- Have BusinessObjects XI installed

Creating a mobile BusinessObjects application

In this tutorial, you will create a BusinessObjects drill-down application that you can deploy to your mobile device. You are creating a BusinessObjects application using a Web Intelligence report.

❖ Creating the BusinessObjects Web application

1. From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.
2. Click the down arrow to the right of Add, and select Web Element.

3 In the New Web Element window:

   a In the Address field, enter the BusinessObjects InfoView URL. For example:

   `http://`***`hostname.domain`***`:`***`port`***`/businessobjects/enterprise11/desktoplaunch/InfoView/logon/logon.do`

   For example, if your machine name is “lab2k”, your domain is “sybase.com,” and your port number is “8080,” enter:

```
http://lab2k.sybase.com:8080/businessobjects/enterprise11/
desktoplaunch/InfoView/logon/logon.do
```

   b Select Advanced Navigation.

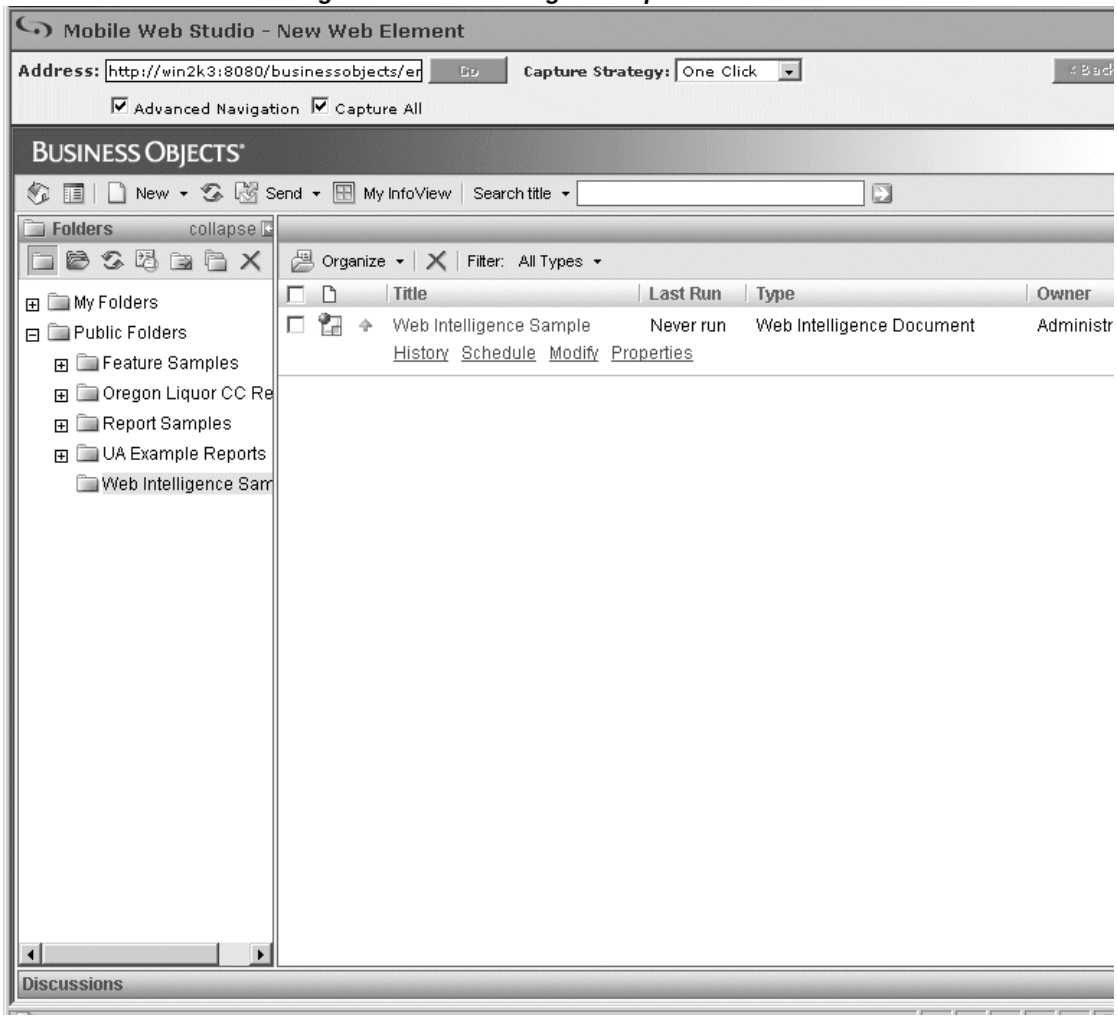
   c Click Go.

4 When the BusinessObjects Login page loads, enter your BusinessObjects XI user name, port, and password. Accept the defaults for System and Authentication.

   Click Log On.

5 In the left pane of the BusinessObjects window, expand the Public Folders by clicking on the +. Expand the Web Intelligence Samples folder, and click on Web Intelligence Sample in the right pane to open it.

Figure 6-1: Web Intelligence report

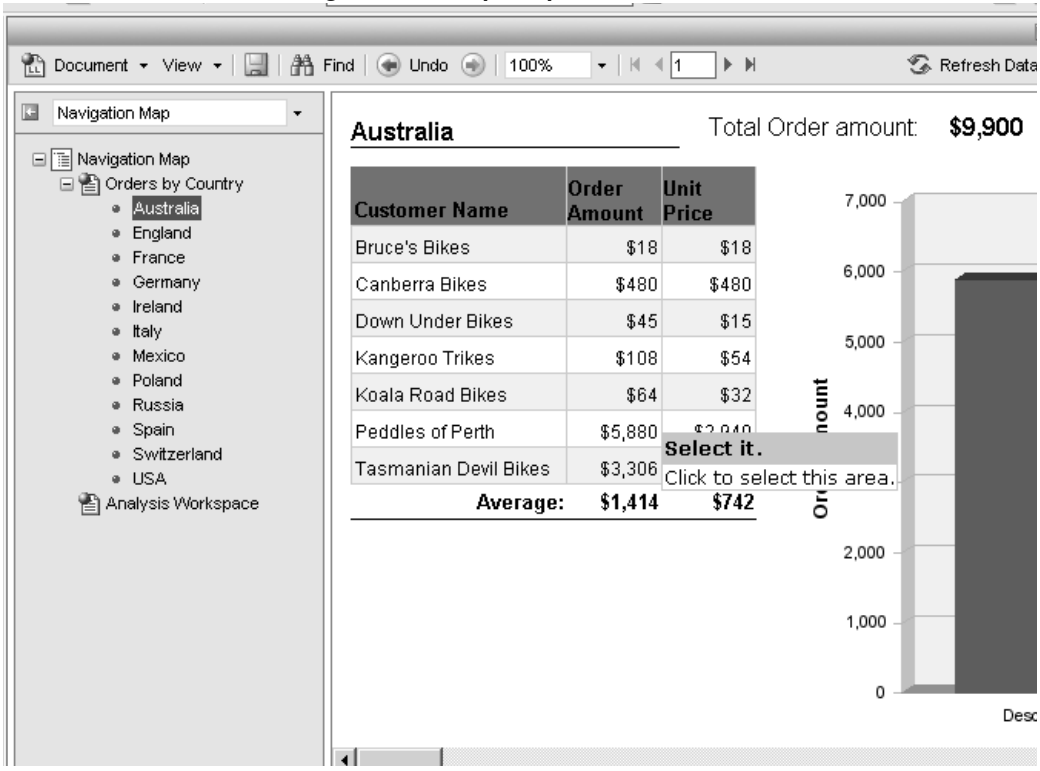


6 Open the Navigation Map and select the report you want to mobilize. From the Capture Strategy drop-down list, select One-click.

Click Next.

7 Place the cursor over the area of the report you want to mobilize. When you see the floating yellow box with the Select It message, click the mouse.

Figure 6-2: Sample report



8 In the next window, select “Rewrite CCL Using a Predefined Template,” then, from the drop-down list, select Use BusinessObjects XI Template.

Click Apply.

9 The New Web Element displays various selectable areas. Click the Select button next to the area you want to mobilize.

10 When the Select button is highlighted, click Next.

11 In the Define window, in Define Record Layout, select Records Contain Labels. Click Next.

12 In the Filter window, click Next.

13 In the next window, in Element Name, enter a name for the application. Click Finish.

14 In the Application Builder, click Save.

15 On the Finish window, make these entries (otherwise accept the defaults):

Roles tab Click Add All to add all roles to this application.

Mobile tab Select Make Available to Disconnected Mobile Devices.

16 Click Finish to save the application, then click OK to confirm.

17 Click Close in the upper-right corner to close the Application Builder window.

18 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The new BusinessObjects application displays in the detail pane.

19 In Mobile Web Studio, right-click the application you just created in the detail pane, and select Approval Status | Approved.

Click OK.

20 Select Approved from the Application Manager Status menu. You see the newly approved application in the detail pane.

You can now deploy the BusinessObjects application to your mobile device following the procedures outlined in Chapter 11, “Deploying Applications to BlackBerry Devices,” and Chapter 10, “Deploying Applications to Mobile Devices.”

CHAPTER 7

Creating a Domino Application

This chapter describes how to create a multipage Domino application by creating three applications and then linking them.

Overview

This tutorial assumes:

- You know how to log in to Mobile Web Studio.
- You know how to create and approve applications in Mobile Web Studio.
- You configured the Domino connection as described in the *Unwired Accelerator Administration Guide*, in Chapter 3, "Post-installation Configuration."

These tasks are described in these sections:

- Creating the Domino list application (Demo_HelpDesk_List)
- Creating the Domino update application (Demo_HelpDesk_Update)
- Creating the Domino insert application (Demo_HelpDesk_Insert)
- Linking the applications

Creating a Domino application

In this tutorial, you set up three applications—a master list application, and two drill-down applications (update and insert)—that extract data from the Domino database. You can then link all the elements to create a mobile application that can be deployed to a mobile device.

❖ **Creating and approving the list application**

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.

2 Click the down arrow to the right of Add, and select Domino Element.

3 On the Domino Element Definition window, make sure the Domino URL option is selected.

Enter the Domino connection information:

Note The fields you see in this window vary according to the connection type you are using This tutorial uses the default connection type, "websession." See the *Unwired Accelerator Developer's Guide* for information about the Domino connection types.

- Database – enter the path for the database file you are using, for example, `NewHelpdeskSystem.nsf`.
- Username – enter your Domino server user name, for example, `UA Group\SYBASE`.
- Password – enter the password that is associated with the user ID file.

  Click Find to populate the Agent and View drop-down lists.
- Application Type – select View.
- View – select a view from the drop-down list, for example, `Help Desk View`.

Click Preview to preview the Domino element.

4 Click Next.

5 In the Define window, in Define Record Layout:

a Select Rows.

b Select Records Contain Labels.

c In Labels are Displayed in Record, enter 1.

d Click Next.

6 In the Filter window, click Next.

7 In the Configure Parameters window, check all parameters with the prefix "_dmno_" that you want to parameterize. Click Next.

8 In the Preview window, in Element Name, enter `Demo_Helpdesk_List`.

Click Finish.

9 In the Application Builder, click Save.

10 On the Finish window, make these entries (otherwise accept the defaults):

Roles tab Click Add All to add all roles to this application.

Mobile tab Select Make Available for Disconnected Mobile Devices.

11 Click Finish to save the application, and click OK to confirm.

12 Click Close in the upper-right corner to close the Application Builder window.

13 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The `Demo_HelpDesk_List` application displays in the detail pane.

14 In Mobile Web Studio, right-click the `Demo_HelpDesk_List` application in the detail pane, and select Approval Status | Approved.

Click OK.

15 Select Approved from the Application Manager Status menu. You see the newly approved `Demo_HelpDesk_List` application in the detail pane.

You have successfully created a list application using the Domino database as a source.

Creating the Domino update application

This section shows how to create the Domino update application (Demo_HelpDesk_Update).

❖ **Creating and approving the Domino update application**

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.

2 Click the down arrow to the right of Add, and select Domino Element.

3 On the Domino Element Definition window, make sure the Domino URL option is selected.

Enter the Domino connection information:

- Database – enter the path for the database file you are using, for example, `NewHelpdeskSystem.nsf`.
- Username – enter your Domino server user name, for example, `UA Group\SYBASE`.
- Password – enter the password that is associated with the user ID file.

  Click Find to populate the Agent and View drop-down lists.
- Application Type – select Agent.
- Agent – select an agent from the drop-down list:

  To pass the parameters to the agent:

  a Parameter Type – select Field Names from View.

  b Select the view you created when you created the Demo_HelpDesk_List application in "Creating and approving the list application."

4 Click Next.

5 In the Define window, click Next.

6 In the Filter window, click Next.

7 In the Configure Parameters window, select any parameters you want to pass to the Domino agent.

All parameters with the prefix "dmno_" are used to parameterize the Domino connection parameters. Click Next.

8 In the Preview window, in Element Name, enter `Demo_Helpdesk_Update`.

Click Finish.

9 In the Application Builder, from the Device Type drop-down list, select PDA, and click Save.

10 On the Finish window, make these entries (otherwise accept the defaults):

Content tab In Name, enter `Demo_HelpDesk_Update`.

Roles tab Click Add All to add all roles to this application.

11 Click Finish to save the application, and click OK to confirm.

12 Click Close in the upper-right corner to close the Application Builder window.

13 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The `Demo_HelpDesk_Update` application displays in the detail pane.

14 In Mobile Web Studio, right-click the `Demo_HelpDesk_Update` application in the detail pane, and select Approval Status | Approved.

Click OK.

15 Select Approved from the Application Manager Status menu. You see the newly approved `Demo_HelpDesk_Update` application in the detail pane.

You have successfully created an update application using the Domino database as a source.

Creating the Domino insert application

❖ **Creating and approving the Domino insert application**

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.

2 Click the down arrow to the right of Add, and select Domino Element.

3 On the Domino Element Definition window, make sure the Domino URL option is selected.

Enter:

- Database – enter the path for the database file you are using, for example, `NewHelpdeskSystem.nsf`.
- Username – enter your Domino server user name, for example, `UA Group\SYBASE`.

- Password – enter the password that is associated with the user ID file.

  Click Find to populate the Agent and View drop-down lists.

- Application Type – select Agent from the drop-down list.
- Agent – select an agent from the drop-down list:

  To pass the parameters to the agent, select Input Parameter and enter values as name value pairs. For example, `param1=value1,param2=value2` and so on.

Click Preview. A preview of the Domino application appears in the Preview pane.

4 Click Next.

5 In the Define window, click Next.

6 In the Filter window, click Next.

7 In the Configure Parameters window, all parameters are displayed as selected. Select any parameters you want to pass to the Domino agent

All parameters with the prefix “dmno_” are used to parameterize the Domino connection parameters. Click Next.

8 In the Preview window, in Element Name, enter `Demo_Helpdesk_Insert`.

Click Finish.

9 In the Application Builder, from the Device Type drop-down list, select PDA, and click Save.

10 On the Finish window, make these entries (otherwise accept the defaults):

Content tab In Name, enter `Demo_HelpDesk_Insert`.

Roles tab Click Add All to add all roles to this application.

11 Click Finish to save the application, and click OK to confirm.

12 Click Close in the upper-right corner to close the Application Builder window.

13 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The `Demo_HelpDesk_Insert` application displays in the detail pane.

14 In Mobile Web Studio, right-click the `Demo_HelpDesk_Insert` application in the detail pane, and select Approval Status | Approved.

Click OK.

15 Select Approved from the Application Manager Status menu. You see the newly approved `Demo_HelpDesk_Insert` application in the detail pane.

You have successfully created an insert application using the Domino database as a source.

Linking the applications

This section shows how to link the master application (list) with the update and insert applications.

1 In Mobile Web Studio, select Build | Applications | Approved from the Application Manager Status menu. You see the Demo_HelpDesk_List application in the detail pane.

2 Double-click the Demo_HelpDesk_list application. The Application Builder window opens.

3 In the Application Builder, click Properties. The Properties Editor window opens.

4 In Properties Editor:

- In Linked Applications, click Add.
- In the Find Application window, click Search. A list of approved applications displays in the Results pane.
- Select the Demo_HelpDesk_Update application and click Add. The Find Application window closes and an entry displays in the Linked Applications section.
- In the Action Type drop-down, select Update.
- In Display_Name, enter a name, for example, `HelpDeskUpdate`. This is the name that displays on te BlackBerry offline client menu options.
- Now link the insert application. In Linked Applications, click Add.
- In the Find Application window, click Search. A list of approved applications displays in the Results pane.
- Select the Demo_HelpDesk_Insert application and click Add. The Find Application window closes and an entry is seen in the Linked Applications section.

- In the Action Type drop-down, select Insert.
- In Display_Name, enter a name, for example, `HelpDeskInsert`.
- Click OK.

5 In the Application Builder, click Save.

6 Click OK in the confirmation pop-up window.

7 Click Close in the upper-right corner to close the Application Builder. The Domino application is now available for synchronization with your mobile device.

Now you are ready to deploy the Domino application to your mobile devices. See Chapter 10, “Deploying Applications to Mobile Devices” and Chapter 11, “Deploying Applications to BlackBerry Devices.”

CHAPTER 8

Creating a Remedy Application

This chapter describes how to create and mobilize a Remedy application with linked parameters.

Overview

This tutorial shows you how to use UA to mobilize Remedy applications based on Web services that are exposed by the Remedy mid-tier server. Mobilization of Remedy applications is accomplished mainly through the Web services component of UA by accessing the corresponding WSDL URLs published by the Remedy mid-tier server.

The tasks to complete this tutorial are described in these sections:

- Creating the Remedy Application (GetHelpDeskList)
- Creating the Update Application (HelpDesk_Modify_Service)
- Creating the Insert Application (CreateHelpDeskCase)
- Linking the applications
- Deploying

This tutorial assumes you:

- Know how to log in to Mobile Web Studio
- Know how to create and approve an application in Mobile Web Studio
- Are familiar with Remedy Web Services creation through the Remedy Administrator

- Are familiar with how to defined the Method, Type, Qualification, Input and Output Mappings for a Remedy Web service
- Have the Remedy Web Service component and mid-tier server installed
- Have the Remedy HelpDesk sample application that is included with the Remedy installation CD installed
- Know how to use linked parameters

Creating a Remedy application

In this tutorial, you will create a master Remedy application and link it to an update and insert application. The resulting application will allow you to mobilize the data in the Remedy server and deliver it to your handheld devices.

Note The data used in this tutorial may vary, as the information is specific to your particular Remedy installation.

Creating the master application

❖ **Creating the GetHelpDeskList application**

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.

2 Click the down arrow to the right of Add, and select Web Service Element.

3 On the Web Service Element Definition window, make sure the WSDL option is selected.

   a In WSDL URL, enter `http://ITSM60Demo.sybase.com/arsys/WSDL/public/ITSM60Demo/HelpDesk_Query_Service`.

   Note You must replace the strings *ITSM60Demo* with the host name of the mid-tier server specific to your own installation.

   Click Get Method.

b Select `HelpDesk_Query_ServicePortType:GetListBy_Requester_FullName` from the drop-down list.

c Next to Content XSLT, click Create. The Web Services Output XSLT Template Customization window appears. Enter:

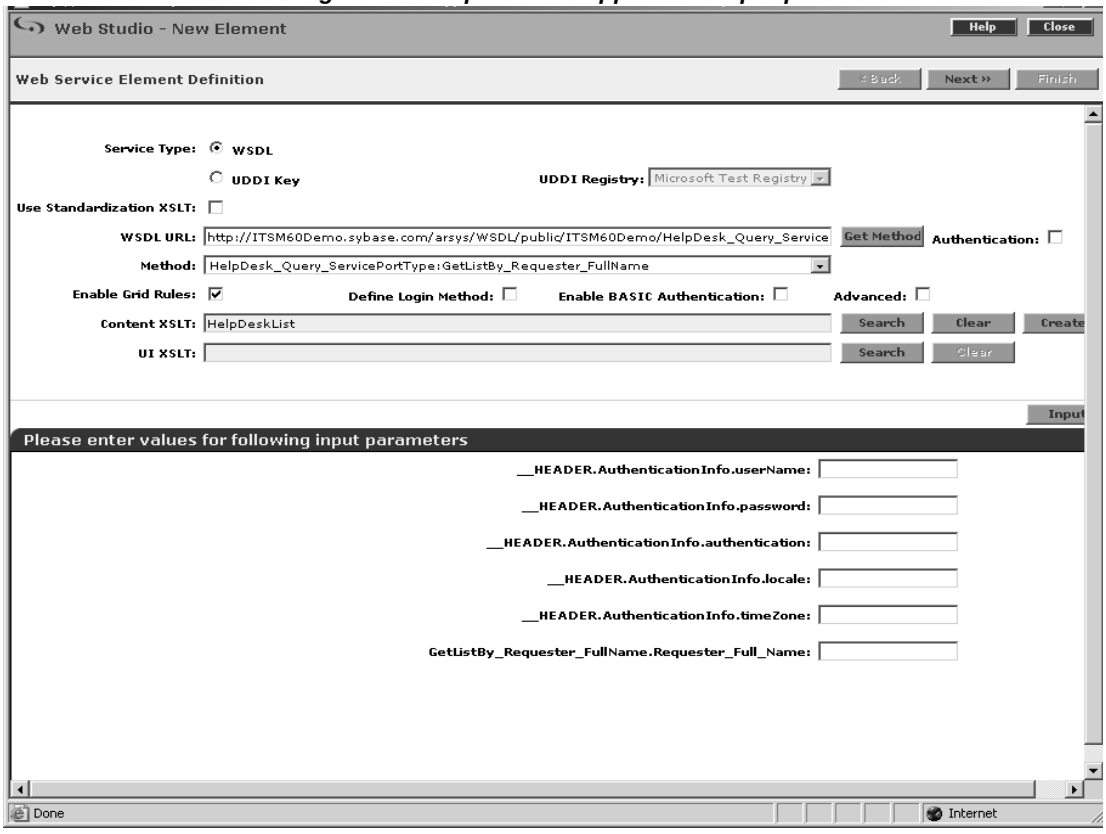
- Name – enter `HelpDeskList`.
- Roles – Click Add All to add all the roles to Assigned Roles.
- Update Output Display Name – click Select.

  The Select Node Level window appears. Select getListValues, and click OK.

d In the Web Services Output XSLT Template Customization Window, click Save, then click OK in the confirmation pop-up window.

e In the Web Service Element Definition window, click Input and enter the values specific to your Remedy installation in the Input Parameters panel:

Figure 8-1: HelpDeskList application input parameters



Click Next.

f A pop-up window asks if you want to proceed even though the parameters are not complete. Click OK.

4 In the Define window, select Records Contain Labels and click Next.

5 In the Filter window, click Next.

6 In the Configure Parameters window, under Variable, select the boxes for:

- _HEADER.AuthenticationInfo.userName
- GetListBy_Requester_Fullname.Requester._Full_Name

Click Next.

7 The Preview window displays. In Element Name, enter HelpDeskList and click Finish.

8 In the Application Builder, click Save.

9 On the Finish window, make these entries (otherwise accept the defaults):

Roles tab Click Add All to add all roles to this application.

Mobile tab Select Make Available to Disconnected Mobile Devices

10 Click Finish to save the application, and click OK to confirm.

11 Click Close in the upper-right corner to close the Application Builder window.

12 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The `HelpDeskList` application displays in the detail pane.

13 In Mobile Web Studio, right-click the `HelpDeskList` application in the detail pane, and select Approval Status | Approved.

Click OK.

14 Select Approved from the Application Manager Status menu. You see the newly approved `HelpDeskList` application in the detail pane.

You have successfully created the primary Remedy application.

Creating the update application

In this procedure, you will create an application called ModifyHelpDeskCase, which allows you to update designated fields according to the specified Case ID on your handheld device. When you save the updates, they are cached until the next synchronization. The updates are then pushed to the UA server, then applied to the Remedy through the associated Web service method.

❖ Creating the ModifyHelpDeskCase application

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.

2 Click the down arrow to the right of Add, and select Web Service Element.

3 On the Web Service Element Definition window, make sure the WSDL option is selected.

   a In WSDL URL, enter:
   `http://`***`ITSM60Demo`***`.sybase.com/arsys/WSDL/public/`***`ITSM60Demo`***`/HelpDesk_Modify_Service.`

   Note You must replace the strings *ITSM60Demo* with the host name of the mid-tier server specific to your own installation.

   b Click Get Method.

   c In the Web Service Element Definition window, click Input and enter the values for the fields you want to update, for example Username, Password, Status, Description, and so on.

   Click Next.

   d A pop-up window asks if you want to proceed even though the parameters are not complete. Click OK.

4 In the Define window, select Records Contain Labels and click Next.

5 In the Configure Parameters window, configure the input parameters, and click Next.

6 The Preview window displays. In Element Name, enter ModifyHelpDeskCase and click Finish.

7 In the Application Builder, click Save.

8 On the Finish window, make these entries (otherwise accept the defaults):

   Roles tab Click Add All to add all roles to this application.

9 Click Finish to save the application, and click OK to confirm.

10 Click Close in the upper-right corner to close the Application Builder window.

11 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The `ModifyHelpDeskCase` application displays in the detail pane.

12 In Mobile Web Studio, right-click the `ModifyHelpDeskCase` application in the detail pane, and select Approval Status | Approved.

   Click OK.

13 Select Approved from the Application Manager Status menu. You see the newly approved `ModifyHelpDeskCase` application in the detail pane.

You have successfully created the update Remedy application.

Creating the insert application

In this procedure, you will create an application called CreateHelpDeskCase, which will allow you to create a new help desk case by entering values in the required fields. A new case ID number is returned once you have successfully created the application.

❖ **Creating the CreateHelpDeskCase application**

1 From Mobile Web Studio, select Applications in the left pane, select New in the Application Manager Status menu, and click the New button to launch Application Builder.

2 Click the down arrow to the right of Add, and select Web Service Element.

3 On the Web Service Element Definition window, make sure the WSDL option is selected.

   a In WSDL URL, enter `http://`***`ITSM60Demo`***`.sybase.com/arsys/WSDL/public/`***`ITSM60Demo`***`/HelpDesk_Submit_Service`.

   Note You must replace the strings *ITSM60Demo* with the host name of the mid-tier server specific to your own installation.

   b Click Get Method.

   c Next to Content XSLT, click Create. The Web Services Output XSLT Template Customization window appears. Enter:

   - Name – enter `HelpDeskCreate`.
   - Roles – Click Add All to add all the roles to Assigned Roles.
   - Update Output Display Name – click Select.

     The Select Node Level window appears. Select "/," and click OK.

   d In the Web Service Element Definition window, click Input and enter the values for the fields you want to update.

Click Next.

e A pop-up window asks if you want to proceed even though the parameters are not complete. Click OK.

f In the Define window, select Records Contain Labels. Click Next.

g In the Configure Parameters window, configure the input parameters. Click Next.

h The Preview window displays. In Element Name, enter `CreateHelpDeskCase`.

i Click Finish.

4 In the Application Builder, click Save.

5 On the Finish window, make these entries (otherwise accept the defaults):

Roles tab Click Add All to add all roles to this application.

6 Click Finish to save the application, and click OK to confirm.

7 Click Close in the upper-right corner to close the Application Builder window.

8 When you return to the Mobile Web Studio main window, select New from the Application Builder Status menu. The `CreateHelpDeskCase` application displays in the detail pane.

9 In Mobile Web Studio, right-click the `CreateHelpDeskCase` application in the detail pane, and select Approval Status | Approved.

Click OK.

10 Select Approved from the Application Manager Status menu. You see the newly approved `CreateHelpDeskCase` application in the detail pane.

You have successfully created the Remedy application.

Linking the applications

Once you have created the master, update, and insert applications, you must link the ModifyHelpDeskCase and CreateHelpDeskCase applications to the GetHelpDeskList master application.

❖ **Linking the update applications to the master application**

1 In Applications | Approved, select the GetHelpDeskList application, and click Edit.

2 In the Application Builder, click Properties.

3 In the Properties Editor window, click the Mobile tab.

4 Under Linked Applications, click Add.

5 In the Find Application window, click Search.

6 From the Results pane, select the ModifyHelpDeskCase application, and click Add.

7 You return to the Properties Editor window, where you see the ModifyHelpDeskCase application listed as a linked application. Enter:

- Action Type – select Update from the drop-down list.
- Menu Level – accept the default value of 1.

8 Link the CreateHelpDeskCase application using the same procedure shown in steps 1–7. Enter:

- Action Type – select Insert from the drop-down list.
- Menu Level – accept the default value of 1.

9 Click OK.

10 In the Application Builder, click Save.

11 Click Close in the upper-right corner of the Application Builder.

Linking the parameters

When setting up a Remedy system like HelpDesk, you typically populate the Remedy data store with standard values for Remedy fields such as Category, Type, Item, and so on. These values may be specific and relevant to your enterprise, so Sybase recommends that you utilize these values automatically when creating mobile applications in UA.

After verifying the WSDL is accessible, you will link the ModifyHelpDeskCase and CreateHelpDeskCase applications to the CTI values you defined when you created the Remedy Web Service in the previous section of this tutorial.

UA includes an application called *RemedyMenuSelections.xml*, located in the *%UA80\tomcat\webapps\onepage\portlets\jsp\custom\remedy* directory, which contains three elements:

- RemedyCTIList – this element uses a Web service like the one you created in “Creating the Remedy Web Service” on page 106 to obtain the Category, Type, and Item selection values from the Remedy system.
- RemedyRSDList – this element uses a similarly created Web service to obtain the Region, Site, and Department selection values.

You can also create additional elements to obtain values from other Web services you may have created.

Since these values may not change frequently in your Remedy system, set a Content Cache Interval value for this application in UA to ensure that the performance of the update and insert Remedy UA applications that link to this linked parameter application do not suffer a degradation in performance.

Each of these elements use a specific UA template to transform the Web service output parameters as needed by UA. Templates for use with a Category, Type, Item Web service (RemedyCTI); and a Region, Site, Department Web service (RemedyRSD), are already included in your UA installation. You can copy and easily modify these templates to apply to other similar selection value lists.

Creating the Remedy Web Service

This section shows how to create a Remedy Web service for obtaining category, type, and item data.

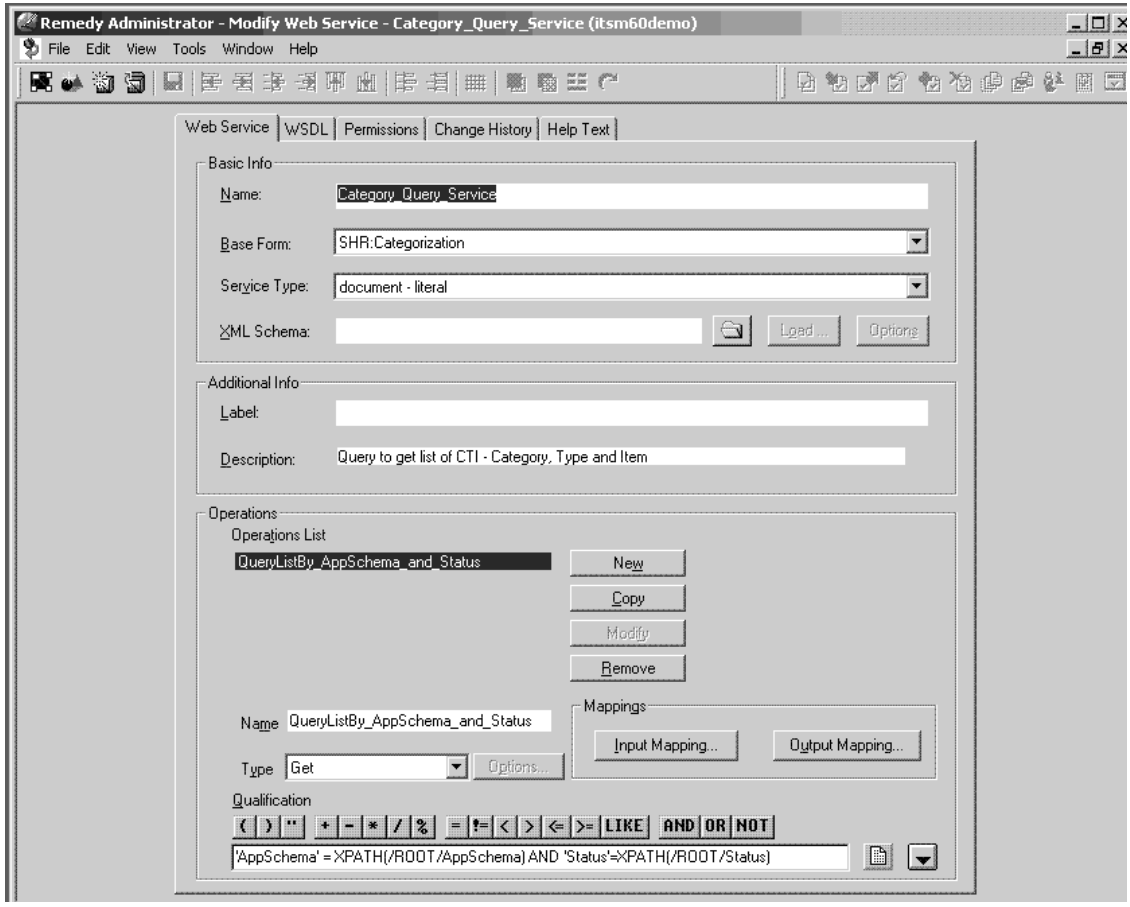
❖ **Obtaining the CTI values**

1 Open the Remedy Administrator wizard and enter your login information.

2 Select Servers | Host |Web Services, where Host is the name of the Remedy server. Right-click Web Services, and select New Web Service.

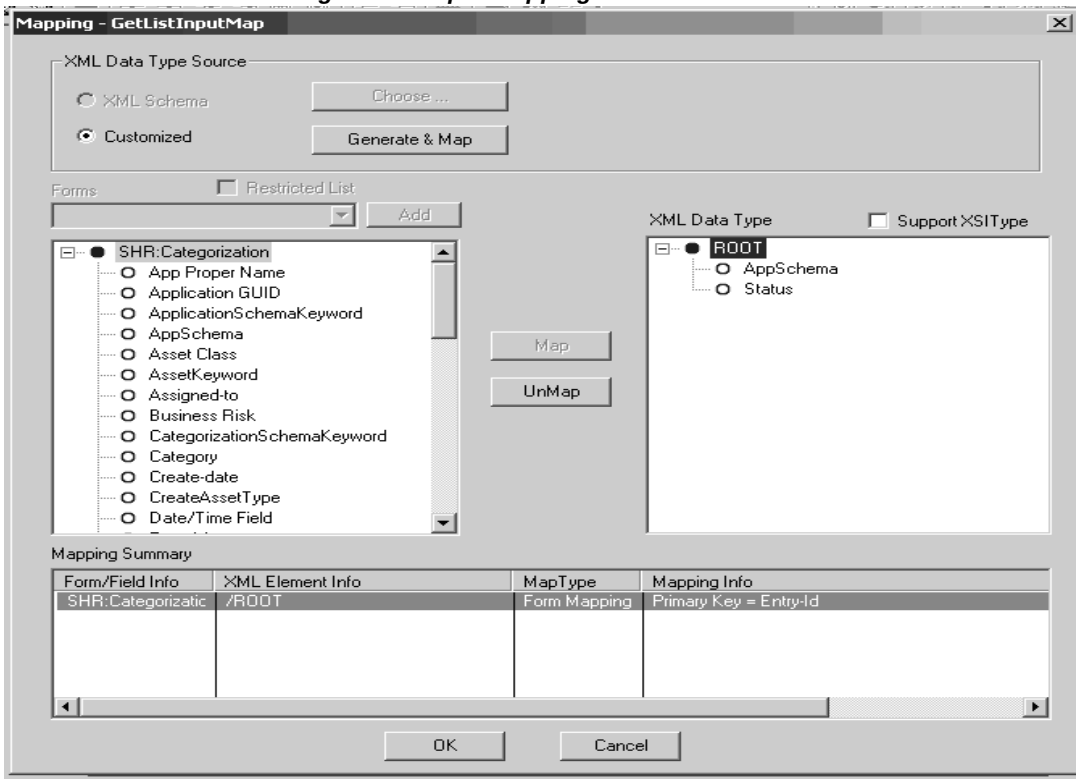
3 In the New Web Service window, make the following entries:

Figure 8-2: New Web Service window



4 Click Input Mapping to define the input mapping as shown.

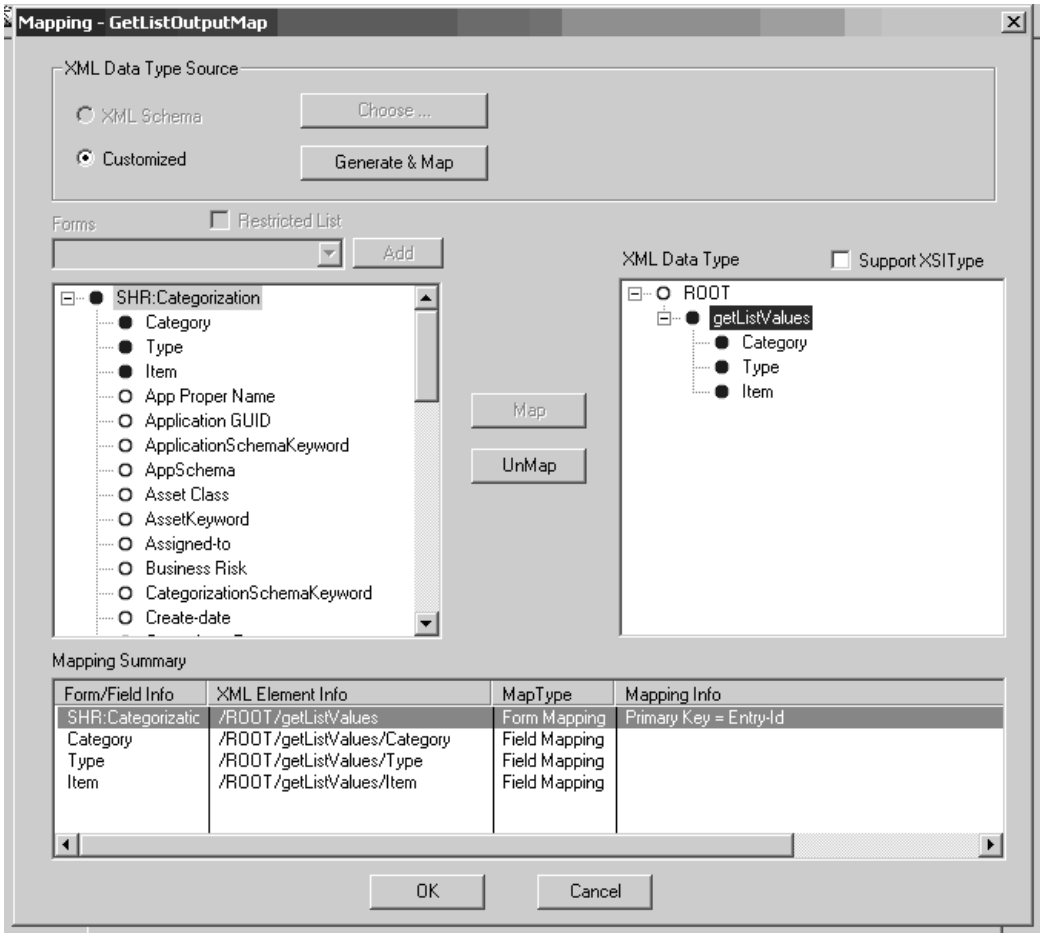
Figure 8-3: Input mapping window



Click OK.

5 Click Output Mapping to define the output mapping as shown.

Figure 8-4: Output mapping window



The ROOT element can be left unmapped, but the getListValues entry is mapped to the SHR:Categorization entry in the form. Category, Type, and Item are mapped to the corresponding entries in the form.

Click OK.

6 Click the Permissions tab to assign the appropriate permissions to the Web Service. From the No Permission list, select Public and click Add.

7 From the main menu, select File | Save | Web Service to save the Web Service.

8 Verify the WSDL is accessible from the Remedy midtier server.

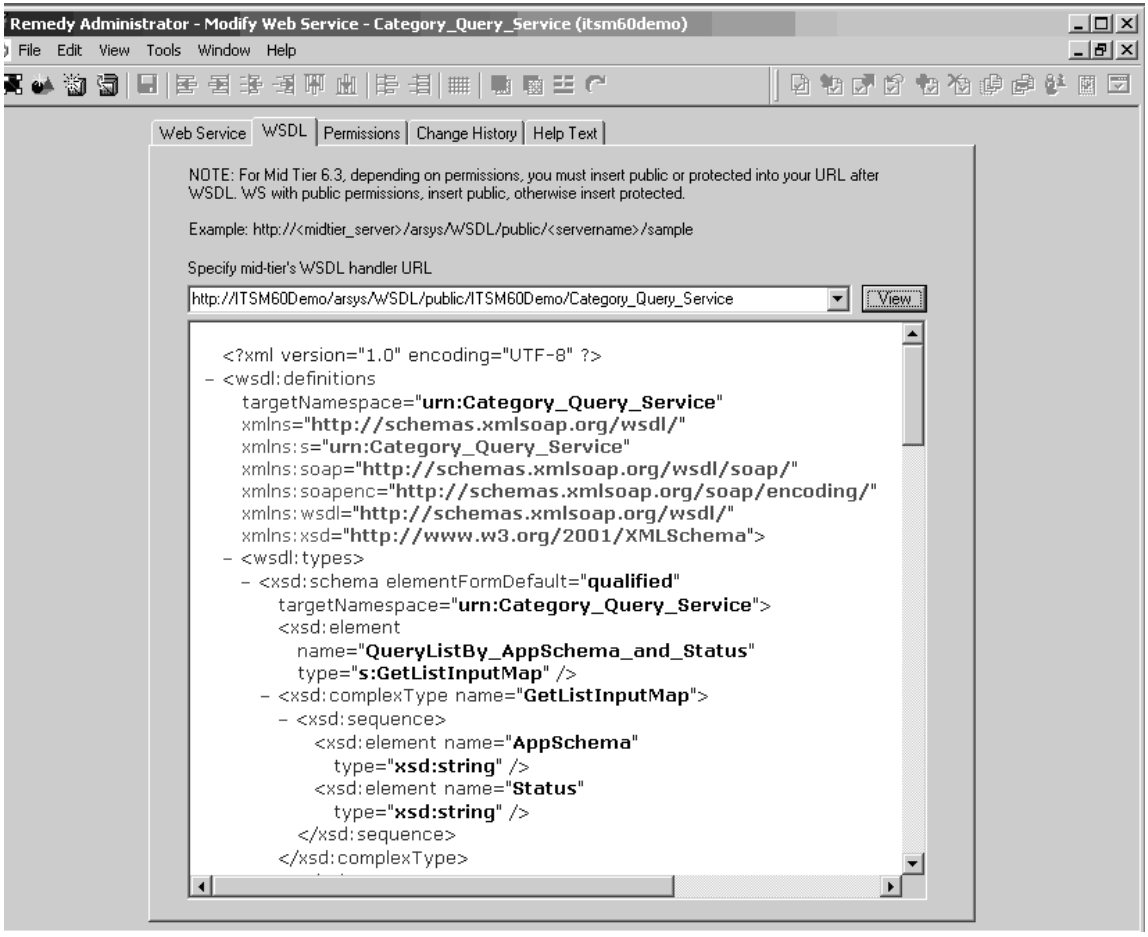
a Click the WSDL tab.

b Edit the WSDL URL field according to the instructions on the tab.

c Click View.

d A WSDL similar to the one shown below displays.

Figure 8-5: Remedy WSDL file



Now you can use this Web Service in Unwired Accelerator to obtain and link the Category, Type, and Item values automatically to your Remedy UA insert and update applications.

❖ **Importing the UA linked parameter application**

1. In Mobile Web Studio, select Build | Applications.
2. In the toolbar above the detail pane, click Import/Export.
3. In the Export/Input window, click the Import tab.
4. Click Browse to navigate to the *%UA80%\tomcat\webapps\onepage\portlets\jsp\custom\remedy* direcotry, and double-click the *RemedyMenuSelections.xml* file.

   Click Upload, then click OK in the confirmation pop-up window.
5. Click Import.
6. Click OK in the pop-up window that confirms the import was successful. You see see the `RemedyMenuSelections` application in the list of approved applications.
7. In the Approved applications detail view, right-click the RemedyMenuSelections application and select Edit.
8. In the Application Builder, in the left pane, you see the RemedyMenuSelections application contains three elements. Right-click the RemedyRSDList element and select Edit | Definition. The Web Service Element window appears.
9. In WSDL URL, you must change the server names and port numbers to match your installation of the Remedy server and Remedy midtier server. Click Finish.
10. Repeat steps 8 and 9 for each element contained in the RemedyMenuSelections application.
11. After editing the definitions of each of the three elements, click Save in the Application Builder window to save the changes.

❖ **Linking the parameters in the ModifyHelpDeskCase and CreateHelpDeskCase applications**

1. In Mobile Web Studio, select the ModifyHelpDeskCase application, and click Edit.
2. In the Application Builder, click Params.
3. In the Configure Parameters window, click Next.
4. For the Category parameter, under Display Type, select Linked.
5. Next to the Category parameter, click Add.

6 In the Find Portlet window, click Search.

7 In the Results pane, select the Category Query Web service you created in Remedy and click Add.

8 In the Configure Parameters window, click Save.

This procedure enables the ModifyHelpDeskCase and CreateHelpDeskCase application to automatically show the defined CTI values in the relevant fields as drop-down lists.

Now you can deploy the Remedy application and try it out. See Chapter 11, “Deploying Applications to BlackBerry Devices.”

For additional information, see *RemedyHelpDeskSample.doc* located in *%UA80%\tomcat\webapps\onepage\portlets\jsp\custom\remedy*, which documents the Remedy sample included in UA.

CHAPTER 9

Creating a Multipage Mobile Charting Application

This chapter shows how to create a mobile application from a chart.

Overview

You can create drill-down charts that use server-side click-across to link multiple charting applications together into a cohesive application.

Note Unwired Accelerator introduces limited device detection for charting applications. To support viewing charting applications on mobile devices with small screens, Unwired Accelerator uses screen resolution detection to alter the resulting chart dimensions. Charting application dimensions are resized only if the device sends the resolution in the HTTP request headers. If there are no resolution headers, the charting application uses the dimensions defined in the charting wizard.

Charting tutorial

To illustrate the use of drill-down charts using server-side click-across, this section contains procedures to create a table of data with the stock market's most active stocks and use drill-down charting to get the quote page with details of a selected stock. The information is presented in a bar chart.

To create a drill-down charting application for 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.

This procedure assumes you:

- Know how to log in to Mobile Web Studio.
- Know how to create and approve an application in Mobile Web Studio.
- Have set the Split window property (SPLITENABLED) to "true" in the *global.properties.xml* file located in *%SYBASE%\tomcat\webapps\onepage\config*.

❖ **Creating the StockDetails application**

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, then click the New button to launch the Application Builder.
3. Click Add.
4. When the New Web Element window appears, in Address, enter `http://finance.yahoo.com/`, and click Go.
5. When the Yahoo Finance Web site displays, type `SY` in the Enter Symbol field below the Today in Finance title bar, and click GO.
6. Verify that One Click is selected as the Format, and click Next.
7. Place the cursor over the Last Trade cell and click the mouse. The window refreshes displaying various presentation possibilities.
8. Click Select next to the first option, then click Next.
9. On the Split window, click Next to bypass the options.
10. On the Define window, click Next to bypass the options.
11. On the Filter window, in the Add Filter Rule section, add a rule to include only rows 1, 4, 5, and 8:
    - In the left-most drop-down list, select "Include record(s)."
    - Make sure the second drop-down list is set to "number."
    - In the text box, enter `1,4,5,8` to indicate you only want to include these records.

- Click Add. In Preview, the records Last Trade, Prev Close, Open, and 1 Yr. Target Est. are highlighted in blue, and a new rule is added under Current Filter Rules.

12 Click Next.

13 On the Configure Parameters window, select the Variable option for the "s" parameter and complete these options for that parameter (accept the defaults for all others):

- Display Name – enter `symbol`.
- Default Value – accept `SY`.
- Type – select None.

Click Next.

14 On the Window Preview, enter `StockDetails` (no space) for the element name and click Finish.

15 In the Application Builder, click Save to create the application.

16 On the Finish window, make this change (accept the default for all others):

Content tab

- Name – enter `StockDetails` (no space).
- In Context – select this option.

Click Finish, and click OK to confirm.

17 Click Close to close the Application Builder.

18 In the Mobile Web Studio main window, select New from the Application Manager Status menu, then right-click the StockDetails application in the details pane and select Create Chart. The Create Chart window displays.

19 Under Choose a Selection, click Select, then click Next.

20 Complete the following options in the Select Chart Type window:

- Chart Title – enter `StockDetails`.
- Chart Height – change this value to 300, since PDA screens are only 320 pixels high.
- Chart Width – change this value to 200, since PDA screens are only 240 pixels wide.
- Font Name – accept the default of SansSerif.
- Chart Legend – select None from the drop-down list.

- Chart Output – accept the default of JPEG.

  Note If you are using Pocket Internet Explorer, you can use Flash. Otherwise, you must use JPEG.

- Chart Type – select Bar Chart.

Click Next to continue.

21 On the Create Chart window, look for the Chart Parameter section, and set these options:

- X-Axis Label – enter `Time` to establish a label for the row.
- Y-Axis Label – enter `Value` to establish a label for the column.
- Category Labels – select "Column 1" from the drop-down list.
- Show Values – select this option.
- Series 1 Data – select "Column 2" from the drop-down list.
- Series 1 Name – accept the default, since the legend is set to None.
- Series 1 Color – select "Blue" from the drop-down list.

Click Next to continue.

22 In the Preview window, you see a stock details bar chart in blue, with labels for Value and Time.

Click Finish to save the application, and click OK to confirm.

23 In Mobile Web Studio, click New under Application Manager Status. The StockDetails application displays in the detail pane.

24 Approve the application:

- Right-click the StockDetails application in the detail pane, and select Approval Status | Approved.

25 Select Approved under Application Manager Status and verify the StockDetails application displays.

❖ Creating the Most Actives stock application

1 In Mobile Web Studio, select Applications from the Build menu in the left pane, select New from the Application Manager Status menu, then click the New button to launch the Application Builder.

2 Click Add.

3 When the New Web Element window appears, enter `http://finance.yahoo.com/` in the Location field and click Go or press Enter.

4 When the Yahoo Finance Web site displays, click the Most Actives link, located in the left column below Market Summaries and above Investing.

Note When the Stock Market is closed, the previous day's most active stocks display.

5 Verify that One Click is selected as the Format, and click Next.

6 Perform a one-click capture on the Most Actives stock table by clicking on Symbol (the cursor flag provides instructions). The window displays various presentation styles.

7 Click Select to the left of the first grid, and then click Next.

8 On the Split window, notice that Column 3 includes several values—trade and time. Split Column 3 into separate columns to accommodate each value:

- In the left drop-down list, select Column No..
- Enter `3` in the text box.
- In the second drop-down list, select Space.
- Click Add. The Preview section shows the changes, and a new rule is added.

Click Next.

9 On the Define window, click Next to bypass the option.

10 On the Filter window, in the Add Filter Rule section, set the following:

- "Include record(s)" and "number", enter `2-6` in the text field, and click Add;
- "Include field(s)" and "number", enter `1,3` in the text field, and click Add.

Click Next.

11 On the Configure Parameters window, click Next to bypass the options.

12 In Window Preview window, enter `Most Actives` for the Element Name, and click Finish.

13 On the Application Builder, click Save to create the application.

14 In the Finish window, change:

Content tab

- Name – enter `Most Actives`.
- In Context – select this option.

Click Finish, then click OK to confirm.

15 Click Close to close the Application Builder.

16 On the Mobile Web Studio main window, select New from the Application Manager Status menu.

17 Right-click the Most Actives application in the details pane and select Create Chart.

18 On the Create Chart window, click Select, and click Next.

19 Complete the following options in the Select Chart Type window:

- Chart Title – enter `Most Actives`.
- Chart Height – change this value to 300, since PDA screens are only 320 pixels high.
- Chart Width – change this value to 200, since PDA screens are only 240 pixels wide.
- Font Name – accept the default of SansSerif.
- Chart Legend – select None from the drop-down list.
- Chart Output – accept the default of JPEG.
- Chart Type – select Bar Chart.
- Create An Image Map – select this option. Several additional options display when this option is selected.
- Category Event Name – enter `symbol`, which is the name of the parameter in the StockDetails application.

  Accept the default options that are selected below the Category Event Name.
- Series Event Name – enter `nothing`. While this is a required value, we are not using it, hence the value “nothing”.

  Accept the default options that are selected below the Series Event Name.
- Client Side – make sure this option is not selected (unchecked).

20 Click Find Application.

21 When the Search window displays, click Search, select the StockDetails application in the Results pane, then click Add. The Name, Resource ID, and Window ID of the associated application display.

22 Click Next to continue.

23 In the Chart Parameters window, complete these options:

- X-Axis Label – enter `Stock` to establish a label for the row.
- Y-Axis Label – enter `Volume` to establish a label for the column.
- Category Labels – select Column 1 from the drop-down list.
- Show Values – select this option.
- Series 1 Data – select "Column 2" from the drop-down list.
- Series 1 Name – accept the default, since the legend is set to None.
- Series 1 Color – select "Blue" from the drop-down list.

Click Next to continue. The Most Actives chart displays in the Preview window.

24 In the Preview window, click Finish.

25 In the Mobile Web Studio main window, right-click the Most Actives application in the detail view and select Approval Status | Approved.

26 When you see the confirmation that the application was saved successfully, click OK.

❖ **Previewing the drill-down chart application**

1 Select Approved from the Application Manager Status menu.

2 Select the Most Actives application in the detail pane and click Preview.

3 Click one of the bars in the bar chart. You see the StockDetails charting application with a bar chart for the selected company.

Note The Most Actives application is used in "Creating the Most Actives stock application" on page 116.

CHAPTER 10

Deploying Applications to Mobile Devices

This chapter describes how to deploy applications from Mobile Web Studio to mobile devices.

Deploying to a PDA tutorial

This tutorial shows you how to deploy mobile applications to a PDA. This tutorial uses the `employeeSales` application you developed in Chapter 4, “Creating a Multipage Mobile Application with Transaction Support.”

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 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.

Publishing the mobile application

In this step, publish the employeeSales application. Once published, the application is available for use on a PDA. The application enables you to view sales figures for each employee, by region and customer identifier.

❖ **Deploying applications to online devices in 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 in 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 employeeSales application, and click Add.
- Click Save.
- For Name, enter employeeSales, 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 employeeSales 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 employeeSales 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 employeeSales 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 employeeSales 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 employeeSales 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.

Trying out the employeeSales application

In this step, try out the employeeSales application from a mobile device, such as Palm OS or PocketPC, or a mobile device simulator.

❖ **Using the mobile application on a PDA**

1 Perform a synchronization to your mobile device.

2 On the mobile device, you should see the employeeSales in the subscribed group list.

3 Click the employeeSales link to view the mobile application.

4 Click one of the customer identifiers in the cust_id column to access detailed customer order information. For example, click 136 to see detailed sales information about one of Chin's customers.

5 Return to the main page.

6 From the drop-down list, select Western to access records in the Western region.

7 Try another record.

CHAPTER 11

Deploying Applications to BlackBerry Devices

This chapter provides special instructions for working with applications to be deployed to the RIM BlackBerry, for use in online or offline modes.

Overview

Unwired Accelerator enables you to create applications specifically tailored for use on BlackBerry devices. For the applications you create in Mobile Web Studio, you can specify the columns to display in list view, and in the drill-down detail view; and you can customize the BlackBerry display template for all applications or for particular applications. You can also create applications using non-grid data as a source.

Note Unwired Accelerator has limited support for non-tabular data.

When applications are deployed to the BlackBerry device, you can easily synchronize application data, delete unwanted applications and data, update data, view high-level list and drill-down detail data, and sort data.

BlackBerry applications tutorial

This tutorial shows you how to use Mobile Web Studio to set up applications for use on BlackBerry devices. Topics include:

This tutorial assumes you:

- Know how to log in to Mobile Web Studio
- Know how to create and approve an application in Mobile Web Studio
- Have created the employeeSales application described in "Mobile applications with transaction support tutorial" on page 41.
- Have created the SSCAMaster application described in "Creating multiple-page applications" on page 27.

Setting up applications for the BlackBerry device

This describes how to set up applications to be accessible on the BlackBerry device, in offline mode and online mode.

❖ **Setting up applications for BlackBerry in offline mode**

1 In Mobile Web Studio, select Applications from the Build menu in the left pane, select Approved under Application Manager, and select the employeeSales application you created in "Mobile applications with transaction support tutorial" on page 41.

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 device during synchronization.

Note Select "Make Available for Disconnected Mobile Devices" only for the master application, not its linked applications.

5 Click OK and save the application.

6 In the Application Builder, click Preview. The employeeSales application displays in the Preview panel.

7 Click Close to exit Application Builder.

8 On the Mobile Web Studio window, there is now a check mark for the application in the Offline column. This column shows whether an application has been activated for BlackBerry offline.

To try out the application on your BlackBerry device or simulator, see "Retrieving applications on the BlackBerry device" on page 132.

❖ **Setting applications for BlackBerry in online mode**

1 In Mobile Web Studio, create a page for the SSCAMaster application:

- Select Pages in the left pane, and click the New button.
- Click Add.
- Click Search.
- Select SSCAMaster, and click Add.
- Click Save.
- In the Save Page window:
  - Name – enter SSCAMaster
  - Click Add All to select all roles.
  - Click OK to save, then click OK to confirm.
- Click Close to exit the Page Builder window.
- Approve the page (click New under Page Manager, right-click Currency, 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 SSCAMaster, then click Add.
- Click Save.
- In the Save Page Group window:
  - Name – enter SSCAMaster
  - Select Add All for both Navigation Styles and Roles.
  - Click OK to save, then OK to confirm.
- Click Close to exit the Page Group Builder window.

- Approve the Page Group (click New under Page Group Manager, right-click SSCAMaster; select Status | Approved, then click OK to confirm).

3 Deploy the page group:

- Select Page Group, and Approved.
- Select SSCAMaster in the detail pane.
- Click Update, then 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`***`:`***`port`***`/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.

c Log out of Portal Interface.

Establishing list and detail settings

This section describes how to set up list and detail settings for applications that are to be deployed to a BlackBerry device.

❖ **Setting up list and details settings for an application**

1 In Mobile Web Studio, select Applications from the Build menu in the left pane, select Approved under Application Manager, and select the SSCAMaster application.

2 Click Edit. The Application Builder displays.

3 In the Application Builder, select Properties.The Properties Editor opens.

4 Select the List/Detail tab. The List/Detail tab is used to identify the columns to include in the list view and in the detail view. By default, all columns are in the Selected list, indicating all columns will be displayed in the list and detail views on the BlackBerry.

5 In the Detail view, change the order of the columns to display by selecting Dept Head, and using the up arrow to move it to the first position, after Dept Name. The first entry in the Selected list is the first column to display on the BlackBerry screen.

6 To change the display width of a column, in List View, select the column you want to change the width for, and enter a number in the Width box, for example, 5, for five characters wide.

Click the plus button.

7 In the Device Config section, select the size of your mobile device, or click Find to find a predefined template for your device.

8 Click OK to save and close the Properties Editor window.

9 In Application Builder, click Save to save the changes to the application, and OK to confirm.

10 Close Application Builder.

Customizing online BlackBerry templates

This section describes how to customize templates for applications that are to be deployed to a BlackBerry device and used in online mode. You can modify the basic BlackBerry template, or create various templates.

Note You can convert unstructured data to a grid format using JSP, and then apply the BlackBerry template.

❖ Customizing BlackBerry templates for online mode

Note Menu items vary according to the version of the BlackBerry wizard you are using. This procedure shows examples using version 3.7.

---

1 In Mobile Web Studio, select Templates in the left pane, and Approved under Template Manager Status.

2 Click the New button to start Template Upload.

3 On the Template Upload, click Wizard, and select BlackBerry Online.

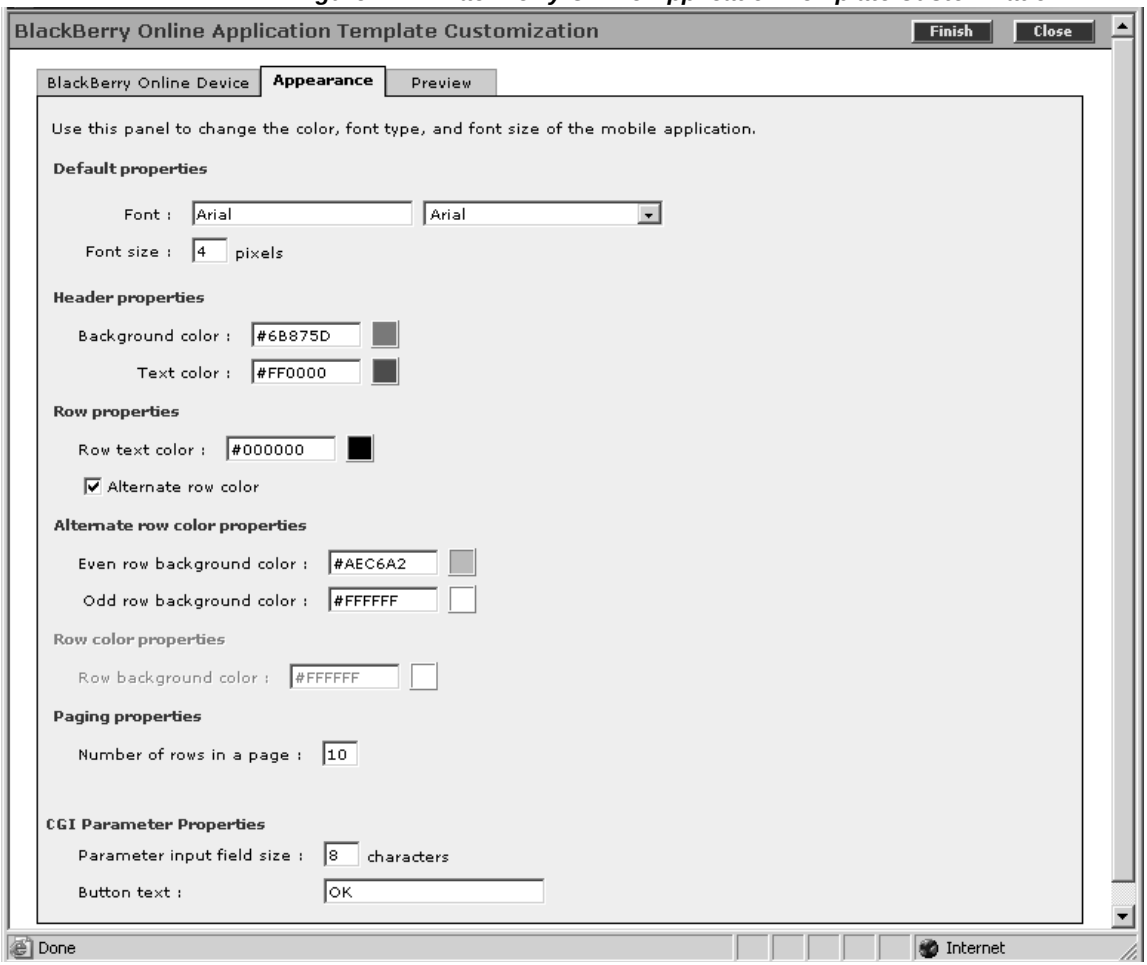
4 In the BlackBerry Application Template Customization editor, make these selections to change the header appearance (otherwise accept the defaults):

BlackBerry Device tab In Name, enter `BlackBerry2`.

Appearance tab Under Default Properties, make these changes to the font appearance:

- Change Font Size to 4.
- Under Header Properties, enter #FF0000 for Text Color.
- Under CGI Parameters Properties, change Ok to OK.

Figure 11-1: BlackBerry Online Application Template Customization



5 Click the Preview tab to preview the changes.

6 Apply the BlackBerry2 template to applications that you plan to deploy to the BlackBerry device. Once you apply the template to an application and synchronize your BlackBerry device, the new template is used on the application.

Click Finish.

7 On Template Upload, select Add All roles.

8 Click Save, and OK to confirm.

BlackBerry device tutorial

This tutorial shows how to use the Unwired Accelerator client on the BlackBerry device. Topics include:

This tutorial assumes you:

- Have the M-Business Anywhere server and client installed and configured as described in the *Unwired Accelerator Installation Guide* (or, have the BlackBerry Enterprise Server and Desktop client software installed and configured).

  Alternatively, you can use the RIM BlackBerry simulator with these tutorials. See the *Unwired Accelerator Installation Guide* for information about obtaining the simulator from Research in Motion.

- Have the Unwired Accelerator offline client software installed on your BlackBerry device or BlackBerry simulator as described in the *Unwired Accelerator Installation Guide*

- Have an Unwired Accelerator user account, based on `masuper`/`m8super`, set up for the BlackBerry device as described in the *Unwired Accelerator Installation Guide*

Retrieving applications on the BlackBerry device

This section describes how to retrieve applications on the BlackBerry device by synchronizing with M-Business Anywhere server or BlackBerry Enterprise Server. You can synchronize all applications or individual applications.

Note You can synchronize applications that have structured grid data, and unstructured grid data. In some cases, applications with unstructured grid data may not display on the BlackBerry device. See "Handling non-grid data" on page 133 for additional information.

❖ **Retrieving applications on a BlackBerry device**

1 The Unwired Accelerator BlackBerry client must be running on the device (you should see the UA icon included in the application menu).

2 Select the UA icon to access the Unwired Accelerator offline client.

3 Select a user profile by selecting Profiles from the trackwheel menu, selecting a profile, setting it as active, and selecting Close from the trackwheel menu to exit. This example uses the `masuper`/`m8super` account.

4 Select Refresh All Apps from the trackwheel menu.

A list of available applications displays.

5 Select the SSCAMaster application, select Change Option from the trackwheel menu, and select the application. A check mark displays to the left of the application. (If you have multiple applications, you can repeat this step to select several individual applications to refresh).

6 Select Refresh App from the trackwheel menu. The selected applications are synchronized. During the synchronization process, status messages display to keep you informed of the progress.

7 Select OK when the Refresh App Completed messages displays. The selected applications appear under My UA Applications.

8 Select SSCAMaster and select Open from the trackwheel menu.

The list view displays when the application is running. The two columns you elected to display in list view—Dept Name and Dept Head—are shown.

9 To view more details of a column, select a row, and select "Details" from the trackwheel menu.

Handling non-grid data

This section describes how to handle applications that use non-grid (or unstructured) data, such as a graphic, a PDF file, or a search window. After synchronization, when you select an application that uses non-grid data, the BlackBerry client launches the BlackBerry browser to display its contents. The BlackBerry client requires online connectivity to retrieve the applications content. If you are offline, establish connectivity and then retrieve the application.

Note Unwired Accelerator has limited support for non-grid data.

❖ **Handling non-grid data**

1 If you open a non-grid application, the BlackBerry browser launches within the BlackBerry client so you can view the application.

2 When you are finished, select Close from the trackwheel menu. The BlackBerry browser closes, and you to return to the BlackBerry client.

Deleting applications and data

This section describes how to clear persistent applications and data that are stored in memory on the device, not the source applications or data. This is useful to free memory and space.

❖ **Clearing out applications and data**

1 From the UA main menu, select Delete Data from the trackwheel menu.

2 Select Delete All.

Note You can use the Refresh All Apps option later to retrieve the Unwired Accelerator applications, using procedures in “Retrieving applications on the BlackBerry device” on page 132.

3 Select Yes to confirm.

Using drill-down applications on the BlackBerry device

This section describes how to use drill-down applications on a BlackBerry device.

❖ **Using drill-down applications on BlackBerry**

Drill-down applications are also known as server-side click-across applications.

1 In Mobile Web Studio, select Applications from the Build menu in the left pane, select Approved under Application Manager, and select SSCAMaster.

2 In Mobile Web Studio, set the SSCAMaster application to be available for disconnected mobile devices:

   a In Mobile Web Studio, select Applications from the Build menu in the left pane, select Approved under Application Manager, and select the SSCAMaster application.

   b Click Edit.

   c In Application Builder, click Properties.

   d In Properties Editor, click the Mobile tab.

   e Select “Make Available for Disconnected Mobile Devices.”

   Click OK.

   f In Application Builder click Save.

3 On the BlackBerry device, refresh the application as described in “Retrieving applications on the BlackBerry device” on page 132.

4 Select a record, and select “Click Thru” on the trackwheel menu. (If an application does not show the Click Thru menu option, it was not created as a server-side click-across application).

   The list view displays the first available link on the application. A breadcrumb trail displays at the top of the grid.

   You can also perform a “Click Thru” from the detail view of a grid row. This displays as an underline text.

Sorting on the BlackBerry device

This section shows how to sort within an application on the BlackBerry device.

❖ **Sorting in BlackBerry applications**

1. In Mobile Web Studio, select Applications from the Build menu in the left pane, select Approved under Application Manager, and select the SSCAMaster application.
2. If you have not already done so, in Mobile Web Studio, set the SSCAMaster application to be available for disconnected mobile devices.
3. On the BlackBerry device, refresh the application.
4. Access a record, and display the trackwheel menu.
5. Under Sort Column, select "Dept Name." The first time you sort a selected column, it sorts in ascending order. The next time you sort the same column, it sorts in descending order.
6. Select "Dept Head" to resort.

Updating applications on the BlackBerry device

This section describes how to update an application from the BlackBerry device.

❖ **Updating applications on BlackBerry**

1. In Mobile Web Studio, select Applications from the Build menu in the left pane, select Approved under Application Manager, and select the employeeSales application.
2. In Mobile Web Studio, set the application to be available for disconnected mobile devices.

   Note Select "Make Available for Disconnected Mobile Devices" only for the master application, not its linked applications. For example, you would select "Make Available for Disconnected Mobile Devices" for the employeeSales application (the master application), but not the customerOrder application (its linked, or "child" application).

3. On the BlackBerry device, refresh the application.
4. Open the application.
5. Highlight a record, and select Click Thru from the trackwheel menu.
6. Highlight a line item, and select Edit from the trackwheel menu.

7 On the Update Record window, select Clear Field from the trackwheel menu, and enter `10` to change the Quantity value.

If you had set up data validation, as described in "Creating an update application" on page 43, you would receive a custom error message if you tried to enter an invalid entry.

8 Select Save from the track wheel menu, and save the change.

9 Select Send Update to update the UA server with the changes.

10 Open the Sales application again, and check the record and line item. You see the new value in the quantity column.

Note The changes made affect only the persistent data on the device, not in the source data. The next time you refresh this application, the changes take effect on the Unwired Accelerator server.

CHAPTER 12

Troubleshooting

This chapter provides troubleshooting information.

Server-side click-across

This section discusses troubleshooting for the server-side click-across feature.

Secondary application executes with incorrect parameter

Sometimes when you define a link between two applications (called server-side click-across), and then click a link in the first application that executes the second application, the first application receives incorrect data, or no data.

This may indicate a naming problem for an event associated with the initial application selected. The event name must match the parameter display name in the target application. If the name is incorrect, the target application is executed with a default value for its input parameters, which may result in unexpected behavior.

In the secondary application, check the display name by editing the application and clicking on the Param button. You can view all the required parameters in the wizard.

1 In Mobile Web Studio, select Applications from the left pane, Approved in the Application Manager Status menu. The approved applications display in the detail pane.

2 Click the secondary application in the detail pane and click Edit in the toolbar.

3 Click the Param button to display the application's parameters.

4 Check to make sure the event name matches the display name for the target application.

Application links

This section discusses troubleshooting the application links.

Transaction link does not display

If you create an application that is linked to an update application, but the update or edit link does not display, this typically indicates the update application does not have "Update" selected in the "Kind" drop-down list on the Configure Parameters window. You must select at least one variable to "update" in the update application in order for the update or edit link to display.

See "Creating an update application" on page 43.

Mobile Applications

This section discusses troubleshooting mobile applications.

Deployed mobile application does not appear in PDA

If the deployed mobile application does not appear in the mobile device, check the following:

- Verify that the mobile application is deployed to the group. In Mobile Web Studio, select Manage | M-Business | Groups and check the groups.

- Verify that the M-Business username/password and server properties are set correctly in M-Business Client. M-Business Clients use M-Business users, not the Mobile Web Studio users. See the *Unwired Accelerator Installation Guide* for configuration information.
- Verify that the user belongs to the group that contains the mobile application.

Mobile application is blank in M-Business Client

If the mobile application shows nothing on the mobile device, do a PDA preview on the mobile application to verify that all the data displays properly.

JVM Sync error on BlackBerry device

If you receive a JVM sync error, and a message indicating the element is too big, then try the following:

- Try resyncing. See "Retrieving applications on the BlackBerry device" on page 132 for information about synchronizing.
- Try selecting a single application, and resyncing. See "Retrieving applications on the BlackBerry device" on page 132 for information about synchronizing.
- Try deleting all applications, using the Delete Data option on the trackwheel, and then resyncing. See "Deleting applications and data" on page 133 for information about clearing the BlackBerry device's memory.
- If the problem persists, try deleting the application and recreating it using a smaller dataset (for example, you could modify the SQL query to limit the results set), or including only required functionality.

Glossary

API

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.

ASP

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.

adapter

A component that provides an interface between an internal application and external applications or messaging systems. An adapter detects events and validates event contents.

Application Builder

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).

channel

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.

click across

An Unwired Accelerator feature that enables you to connect related or unrelated applications in a flow using events.

client/server

A network architecture in which one or more computers (servers) accept requests for services from one or more workstations (clients).

This may also refer to a back-end application (server) that accepts requests for information from a front-end application (client).

Configure Parameters window 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.

connection pooling 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.

connectionless communications Communications that do not require a dedicated connection or session between applications.

continuous capture An Unwired Accelerator feature that enables you to capture a set of Web pages from a remote site and define how to extract the content for display.

Continuous capture window Application Builder window used to capture a set of Web pages from a remote site and define how to extract the content for display.

Define window Application Builder window used to define the grid layout of an application.

enterprise 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.

EP An acronym for enterprise portal. An enterprise portal integrates all aspects of an organization's IT infrastructure and offers customers, partners, vendors, and employees a broad array of resources and services, including personalized information, online purchasing, e-mail, forums, search engines, and product support.

event 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.

event definition A set of criteria that are used to determine the contents of events.

Filter window 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.

Finish window Application Builder window used to configure the application for use.

grid rules The Unwired Accelerator feature for manipulating the content and format of an application for display on a mobile device.

HTTP Hyper Text 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.

HTTPS The secure version of HTTP.

Internet A global network connecting millions of computers.

intranet A private network within an organization.

JDBC JDBC is a data access interface based on **ODBC** and used with the **Java** programming language.

J2EE Sun software: Java 2 platform, Enterprise Edition.

Java 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.

LDAP 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.

M-Business Anywhere 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.

metadata Data that describes other data. Any file or database that holds information about another database's structure, attributes, processing, or changes.

Mobile Web Studio 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.

New element window Application Builder window used to create the element of your choice, including elements for Web, XML, HTML, JSP, database, document, and so forth.

ODBC 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.

Split window 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.

SOAP 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.

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.

SOAP was developed by Microsoft, DevelopMentor, and Userland Software and has been proposed to the Internet Engineering Task Force (IETF) as a standard.

SQL Structured Query Language.

SSL 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.

server A computer or software package that provides specific capabilities to client software running on other computers.

servlet 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.

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.

sockets A portable standard for network application providers on TCP/IP networks.

stored procedure 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.

style sheet General term for software that transforms XML documents based on one XML vocabulary into XML documents based on a different XML vocabulary. Example style sheets are Java Server Pages (JSPs) and XSLT style sheets.

TCP/IP 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.

Unwired Accelerator 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.

Window preview window Application Builder window used to view the element and give it a name.

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

A

B

C

D

E

F

G

H

I

J

L

M

N

P

R

S

T

U

W