SYBASE®

Feature Guide

Unwired Accelerator

6.5

DOCUMENT ID: DC00112-01-0650-01

LAST REVISED: January 2005

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

Contents

About This Book

Audience

This book is for developers using Sybase® Unwired Accelerator version 6.5 to develop and deploy mobile applications.

How to use this book

Use this book to learn about the features in this release:

- Chapter 1, "Introduction," gives an overview of Unwired Accelerator features.
- Chapter 2, "Developer Features," describes new developer functionality in Unwired Accelerator Mobile Web Studio.

To learn about Portal Interface features, see the *Portal Interface User's Guide*.

To use some procedures that illustrate some of the features in Unwired Accelerator, see the *Mobile Application Development Tutorial*.

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.

Enterprise Portal and Unwired Accelerator online documentation The Unwired Accelerator documentation set includes:

- The Unwired Accelerator *Quick Start Guide* walks you through two simple scenarios to mobilize a Web application and a database, and to deploy the results to either a PDA or BlackBerry device.
- The *Mobile Application Development Tutorial* contains tutorials for developing applications and deploying them to mobile devices.
- The *Enterprise Portal Developer's Guide* includes development topics for Unwired Accelerator components, Portal Interface portlets, and Java Template Framework pages.

- 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.
- The *Unwired Accelerator Feature Guide* (this book) provides an overview of features included in Unwired Accelerator 6.5.

jConnect™ for JDBC™ documents Unwired Accelerator 6.5 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.

Other sources of information

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

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

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

  Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.
- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

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

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

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

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

2 Select Products from the navigation bar on the left.

3 Select a product name from the product list and click Go.

4 Select the Certification Report filter, specify a time frame, and click Go.

5 Click a Certification Report title to display the report.

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

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

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

2 Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

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

1 Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.

2 Select EBFs/Maintenance. 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:

Key	Definition
commands and methods	When used in descriptive text, this font indicates keywords such as: • Command names used in descriptive text • C++ and Java method or class names used in descriptive text • Java package names used in descriptive text
variable *filename*	Italic font indicates: • Program variables, such as *myCounter* • Parts of input text that must be substituted, for example: *Server*.log • File names
%SYBASE%	Variable used to represent the Sybase Unwired Accelerator installation directory on Windows systems.
$SYBASE	Variable used to represent the Sybase Unwired Accelerator installation directory on UNIX systems.
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 a GUI interface, 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

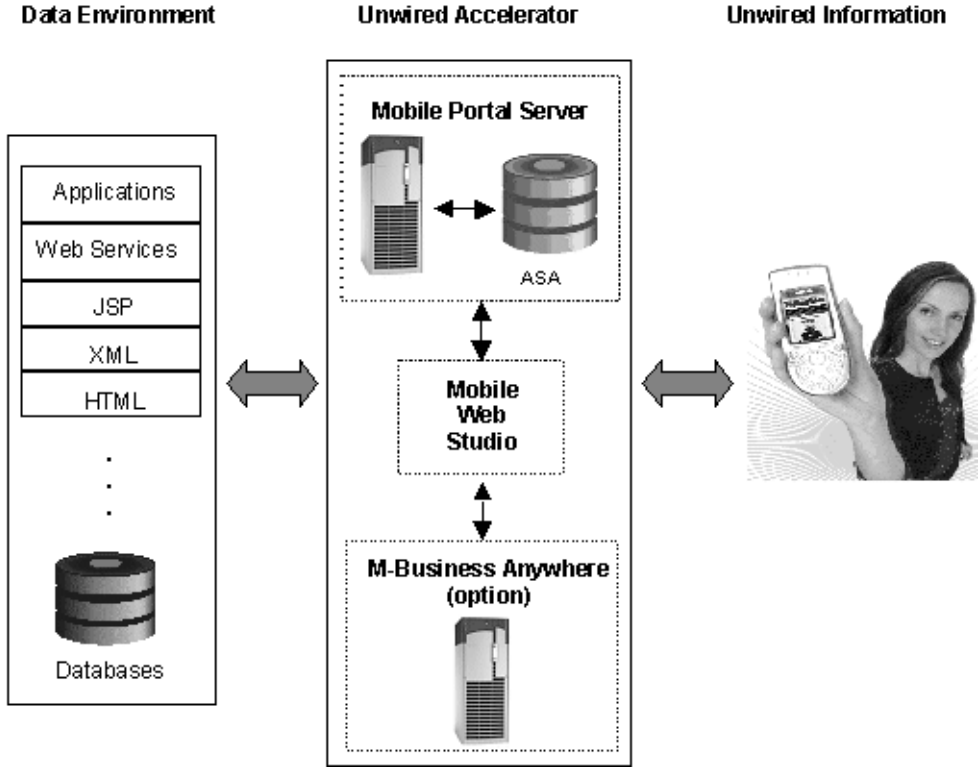
Introduction

This chapter introduces Sybase Unwired Accelerator version 6.5.

Architectural overview

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

Figure 1-1: Architectural overview



Mobile Web Studio	Mobile Web Studio is a Web-based rapid development tool that uses patented technology for creating powerful and interactive mobile Web applications or for mobilizing existing Web applications or data sources like databases, XML, Web Services, HTML, JavaServer Pages (JSPs), and Active Server Pages (ASPs). Mobile Web Studio is a visual development tool with simple-to-use point-and-click functionality.
Mobile Portal Server	Mobile Portal Server delivers Web applications and content in a connected or disconnected environment, coordinating the communication between M-Business Anywhere and legacy back-end systems in an occasionally connected environment. Based on a scalable and distributed architecture, the Mobile Portal Server runs in the J2EE compliant Web container, Tomcat.
M-Business Anywhere	M-Business Anywhere is a reliable, scalable, and secure platform for delivering Web-based applications and content to mobile devices in an occasionally connected model.

Content Capture

Content Capture is a Sybase-patented technology for repurposing Web-based applications and content using an object-oriented scripting language. The Sybase capture engine analyzes Web content and stores the feature definition, or signature. At runtime, the Content Capture engine uses pattern matching to locate the same signature. Therefore, even if content has moved or changed, the engine is still likely to find the information.

Unwired Accelerator features

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

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

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

End-user features

- Users have online, or always-available, connected access to their applications and data from these devices:
  - PocketPC Internet Explorer running Windows CE 2003
  - Motorola MPx200 (Windows Mobile 2002 and 2003 OS)
  - HandSpring Treo 600 (PalmOne OS)
  - Sony Ericsson P900 (Symbian OS)
  - BlackBerry
- Users have offline, or occasionally connected, access to their applications and data from these devices:

    - PocketPC
    - Palm
    - Symbian
    - BlackBerry
- Alerts – schedule or externally trigger application content for processing, including writing content to an e-mail message, SMS, the database, or disk. See “Alerts” on page 71.

Developer features

Unwired Accelerator makes mobile application development easy with these features:

- Application Builder
    - Create new Web-based mobile applications using a point-and-click interface.
    - Mobilize existing Web content, databases, and Web services without programming, including the ability to link, or chain, content from multiple Web pages together using continuous capture. See “Continuous capture” on page 39.
    - Preview new applications.
    - Deploy a single application to multiple devices.
    - Intelligent capturing and repurposing of Web content.
    - Transaction management and conflict resolution. See “Conflict resolution” on page 38.
- Composite applications
    - Create composite applications by linking multiple disparate data sources together
    - Define workflow and communication between applications by defining events and listeners

    See the *Mobile Application Development Tutorial* for information about composite applications.
- Templates

    - Define templates for controlling the appearance of applications. See “Customizing mobile application templates” on page 59.
    - Define navigation styles. See “Navigation styles” on page 27.
- Personalization
    - Define adapters for personalizing content for each user. See the *Enterprise Portal Developer's Guide*.
- Page Builder
    - Create page layout
    - Create and edit pages, and populate pages with applications
- Life cycle management
    - Control the life cycle of applications and pages
    - Use versioning to track application changes

Security features

Unwired Accelerator security features include:

- Authentication – single sign-on – LDAP, user name and password authentication.
- Role-based authorization – you can grant roles for security to a single user or a group of users
- Data confidentiality and integrity – SSL.

Administrator features

Unwired Accelerator makes traditional system administration activities and tasks that control the behavior, content, and appearance of portals easy with features that include:

- Web-based administration
    - User and role management.
    - M-Business server functions management.
    - Page management.

    - Point-and-click deployment.
    - The ability to import and export objects. See “Importing and exporting objects” on page 52.
- Alerts
    - Create alerts based on content or errors
    - Adapters – SMS, e-mail, database, file system, and so on

  See “Alerts” on page 71.
- The ability to define access to functions using roles
- Scalable and distributed architecture

See the *Enterprise Portal Developer's Guide* for more information about these features.

CHAPTER 2

Developer Features

This chapter describes Unwired Accelerator (UA) developer features that are not documented in the *Enterprise Portal Developer's Guide*.

Multiple co-brands

Unwired Accelerator allows you to create multiple co-brands (portals) hosted on one Unwired Accelerator installation. Multiple co-brands allow you to create different looks for different audiences. For example, you might want to use different languages for different audiences.

Each co-brand, or resource, has a row in the resources table, and that resource has a unique resource_id assigned to it when the resource is created.

The users table has a resource_id foreign key reference that previously indicated the resource to which a user belonged. This reference remains, but currently indicates only the default resource to which the user belongs if no resource is explicitly specified in the Portal Interface URL.

See the *Enterprise Portal Developer's Guide*, Chapter 14, "Creating Co-brands," for detailed instructions on using co-brands.

Saving and editing applications

When you create an element in the Application Builder and save the application for the first time, you see the Finish window. The options for saving the application display on seven tabs—Content, Roles, Presentation, List/Detail, Administration, Answers Anywhere, and Statistics. You can also edit an application any time by clicking Properties in the Application Builder. This section describes the available options when you save or edit an application.

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

- Name – a unique name for the application that contains this element.
- Content Type – the type of content the application displays. You can select these option from the drop-down list:
    - Application/msword
    - Application/pdf
    - Application/x-msexcel
    - Application/x-mspowerpoint

    - image/gif
    - image/jpeg
    - text/html
    - text/plain
    - text/rtf
    - text/xml
    - Add New
    - XSL template – displays only when you select “text/xml” as the Content Type. Select an existing XSL template, or create a new one to apply to the application’s content.
    - Output Type – displays only when you select “text/xml” as the Content Type. Select the type of output in which the application’s content displays.
- Charset – the character set in which the application’s content displays.
- Content Cache Interval – the interval at which the content of an application is refreshed for the defined source. If this value is not Real Time, the content is held by the cache for the duration of the specified interval. All requests made during this interval receive the cached content.
    - The default is Real Time, which means that application content is retrieved from its source on every playback request.
    - Specify custom values by selecting Add New from the drop-down list. Select the minute/second drop-down list and enter a value in the text box.
- Parameter – create an invisible parameter to which you can assign click-across events.
- Secure – whether this is a secure application.

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

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

- Linked Applications – allows you to update the application's source content from a PDA and link the source application with the update application.
- No URL Stretch – disables URL stretching (rewriting) on played-back application. If you are having problems with broken images in a played-back application, this can sometimes solve the problem.
- Mobile Offline – if you are using a BlackBerry device, this option makes the application viewable on BlackBerry. See "Setting up Applications to be Accessible on the BlackBerry Device" in the *Mobile Application Development Tutorial*.

Roles tab Users with roles in the Assigned Roles list are the only users that can access the application. Select roles from the Available Roles list and click Add to move the selected roles to the Assigned Roles list.

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

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

  See the *Enterprise Portal Developer's Guide*, Chapter 8, "Building Templates" for information on creating help files.
- Refresh Interval – the interval at which the application's content is refreshed. The browser automatically makes a request for the application content when this interval runs down to zero.
  - The default is Daily, which translates into 1440 minutes.
  - Specify custom values by selecting Add New from the drop-down list. Two additional fields display where you enter the new custom value and select whether that value is in minutes or seconds.

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

  You can create a copy of the default setup page and add custom validation code near the top of the JavaScript function called `doPersonalizeOk<%=windowId%>`.

- Default Size – select the default size at which you want this application to display only if Display Within IFrame is not selected. Unselect Display Within IFrame to select from Show All, Large, Medium, Small, or Add New.

  When you select Add New, another field appears where you enter the size in pixels, for example “500.”

- Display Direction – the direction in which the application’s content displays—LTR (left to right) or RTL (right to left). See “Application content display direction” on page 26 for more details.

- Display Within IFrame – displays the application in an inline frame that displays inside of another HTML document or displays directly on a page with other elements flowing around it.

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

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

  - No Popup – enables this application’s links to display inline; that is, links do not display in a separate pop-up window.
  - Height – the height at which the application displays. Select New to add a new height in pixels. See “Custom application height” on page 26 for details.

- Editable – whether the application can be edited by users in the roles to view the application.

- Show Last Refresh – display the latest refresh image when the application displays. By default, this option is turned on.

- Alert – schedule or externally trigger application content for processing.

List/Detail tab This 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.

See the *Mobile Application Development Tutorial* for information about using the List/Detail tab.

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

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

- Category/Subcategory – a category and subcategory that best describe the application.
- Description/Notes – an optional description or note.
- Active – indicates whether the application is active, which is the default. Unselect the option to inactivate the application. Only applications with an approved status and marked as active are available to users.
- Versioning – creates a new version of the application each time you save it. Each “Save” creates a new version number.

Answers Anywhere The Answers Anywhere option enables you to access application content using natural language queries using a browser or mobile device.

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

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

- Field Synonyms – alternate names for the assigned field, such as "boss" or "supervisor" for the "manager" field.

- Parameter Synonyms
  - Parameters – list of all exposed parameters in the application (the ones you selected in the Parameters window).
  - Assigned Synonyms – assign alternate names to parameters. For example, a Yahoo stock parameter is called "s," which has no real meaning, so you can assign a synonym that has more meaning, such as "ticker" or "symbol."

- Check Conflicts – this button launches a window that alerts you to any potential conflicts in synonyms with other applications. Only synonyms that are used to identify an application are checked. In other words, application synonyms are tested against application synonyms and field synonyms from other applications, and field synonyms are tested against the application synonyms and field synonyms of other applications. Parameter synonyms are not checked for conflicts because they are not used to identify applications.

- Default Filter Field – specify which field to use as a default when dynamically creating filter rules.

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

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

- Custom Error Message – you can customize an error message to be sent in case of an error during the processing of a query.

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

Statistics The Statistics tab displays the user that created the application, creation and modification dates, and other reference information. For example, the resource ID is displayed. Users are grouped under resources. Each resource corresponds to an Unwired Accelerator co-brand.

Answers Anywhere

Answers Anywhere is a Java J2SE-based solution that allows you to design and run agent networks so that users can send simple text-based queries and receive text or HTML responses back. Applications in Approved status are used in query resolution. You can query the portal using:

- Desktop or mobile browsers
- BlackBerry devices
- E-mail
- SMS messaging on SMS-enabled mobile phones or PDAs

See “Setting up Natural Language Search” in the *Mobile Applications Development Tutorial* for information about searching using these methods.

The Answers Anywhere Web application contains:

- Web application SDK JAR files
- Agent Network *wired.opal* file, which is an XML file that stores the information about the UA agent network. This file is located in *%SYBASE%\UA65\tomcat\webapps\dejima\dbdsUnwired*.
- Synonyms files, such as *application.synonym*, *call.synonym*, and *send.synonym*. These files contain synonyms for various agents and entities in the network. These files are located in *%SYBASE%\UA65\tomcat\webapps\dejima\dbdsUnwired*.
- Library and properties folders, located in *%SYBASE%\UA65\tomcat\webapps\dejima*.
- The *uadejima.properties* file, located in *%SYBASE%\UA65\tomcat\webapps\dejima\WEB-INF\classes* which contains the property settings for configuring the e-mail server and user account the e-mail interface uses, as well as the location of the Agent Network’s *wired.opal* file.

- Connection Cache – Answers Anywhere uses the portaldb connection cache.

See “Setting up Natural Language Search” in the *Mobile Applications Development Tutorial* for information about creating an Answers Anywhere application and establishing synonyms.

Answers Anywhere Agent Network

The Answers Anywhere Agent Network is the mechanism that processes queries on behalf of UA. When a query is received by UA and passed to the Agent Network it is analyzed and broken into its constituent components in this way:

1 The Agent Network examines the command portion of the query to identify whether the query should be handled by the execute or send agent. This is done by comparing the command to the synonyms for each agent.

2 The Agent Network examines the application_name portion of the query and attempts to locate the application in the UA database. To locate the application, the Agent Network inspects up to four parts of each application: the application name, application synonyms, field names and field synonyms. The field names and field synonyms only apply if the application contains grid structured data.

3 The Agent Network processes the remainder of the query for parameter and/or filter information.

While this sequence of events is taking place the Agent Network is also building a structured XML representation of the query, known as the Intent XML, which is returned to UA. If the Agent Network successfully understands and processes the query, the Intent XML contains all the information the UA needs to execute the desired application, including any parameter and data filtering information specified in the query. See “Intent XML” on page 18.

Other language support

UA 6.5 Ask Anywhere integration supports simplified Chinese queries, as well as English queries. The Agent Network loaded determines what language is supported when using the Ask UA feature. UA 6.5 includes Agent Networks for English and simplified Chinese. To support simplified Chinese queries, you can change the Agent Network.

❖ **Changing the Agent Network to Simplified Chinese**

1 Open the *uadejima.properties* file located in *%UA65%\tomcat\webapps\dejima\WEB-INF\classes*.

2 Search for `default.opal=/dbdsUnwired/wired.opal`.

3 Change the default.opal parameter to:

```
default.opal=/dbdsUnwired_zh/wired.opal
```

Language syntax

Answers Anywhere Agent Networks are usually designed for very specific applications. The Agent Network created for Unwired Accelerator is a very generic Agent network that is used for all Unwired Accelerator applications. Queries must conform to some basic syntax rules.

Syntax

The basic language syntax is as follows, although you can mix the order of the elements:

command application_name parameters and/or record filters

No matter how the question is constructed, you must supply at least two pieces of information: (1) the action or command, and (2) the UA application on which to apply the action. The optional parts are specifying parameters or recording filters which affects the content of an application that will be returned to the user.

Synonyms are used in commands, applications, parameters and fields. Synonyms are alternative words that the Answers Anywhere search mechanism uses to carry out the search. For example, if the field “auto” has “car,” “truck,” and “vehicle” defined as synonyms, a query like “find cars with make Acura and color blue” would include the field “auto” in its search for application data.

Commands

The UA Answers Anywhere agent network uses two explicit commands—execute and send.

The execute agent processes queries that are requests for information that are returned to the user who created the query.

The send agent processes queries that are for collaborative efforts, such as sending the response content to a person other than the user who initiated the query. An example is “send John Doe’s phone number to jane.doe@domain.com.” In this case, the results of the query are sent to jane.doe@domain.com instead of to the user who initiated the query.

The synonyms defined for execute and send are in Table 2-1.

Table 2-1: Synonyms for execute and send

Command	Synonyms
execute	invoke, play, run, get, what, when, how, why, who, find
send	give

Application Name

Application name is the name of the application in Unwired Accelerator. Note that this is not the element name in the application builder. You can also define synonyms for application names.

Parameters and filters

Unwired Accelerator applications can have parameters to retrieve dynamic content and use filters to retrieve only the records that are of interest if the application contains grid/structured data.

The explicit word for parameter is “parameter.” If your application has more than one parameter, you need to specify the parameter display name or the display name’s synonyms for each; for example, “find cars with make Acura and color blue.”

The explicit word for filter is “filter.” If you use the explicit word for filtering, the default field name (column name) defined in the application will be used to apply the record filter. Otherwise, you can specify the field name or its synonyms and the value to filter out unwanted records. If the fields are assigned in the application, the field names can be used to identify the application. If that is the case, Unwired Accelerator will only return that field’s content in the structured data.

Intent XML

Intent XML is the information returned by Answers Anywhere. Here is an example of Intent XML:

```
<DejimaInterpretation>
   <Input><![CDATA[find info for wall]]></Input>
   <Understood><![CDATA[find info for wall]]></Understood>
   <System>
    <Execute explicit=“true”>
    <Application>
    <Header>
     <IdList match="info">
      <id>251</id>
     </IdList>
    </Header>
    <Parameter explicit="true"><![CDATA[wall]]></Parameter>
   </Application>
 </Execute>
 </System>
</DejimaInterpretation>
```

This example shows that the original query was “find info for wall.” Using the agent network, the Answers Anywhere runtime resolved this query using the Execute, Application, and Header agents. In the agent network for UA Answers Anywhere, the Header agent is connected to a data source that is linked to the portal database through the portaldb connection cache. The data source is used to determine whether there is an available application that matches the query. In this example, application 251 (the application ID) has been determined to be the relevant application.

Another feature of the Intent XML is the determination of parameters in the query string. The parameter XML node is present in the Intent XML whenever the Parameter agent in the agent network determines there is a parameter. Using this information, the parameter value can be passed along to the application as a parameter during application playback. Certain synonyms for parameters have already been specified in the agent network. These are:

- param
- arg

If multiple applications match the query, the Intent XML contains several id nodes as shown in the example below:

```
<DejimaInterpretation>
<Input><![CDATA[get test]]></Input>
```

```
<Understood><![CDATA[get test]]></Understood>
<System>
    <Execute explicit="true">
        <Application>
            <Header>
                <IdList match="test">
                    <id>281</id>
                    <id>211</id>
                </IdList>
            </Header>
        </Application>
    </Execute>
</System>
</DejimaInterpretation>
```

Both applications 211 and 281 matched the query. When at least one matching application is located, the matching value is placed in the match attribute of the IdList node. This match value is then tested against the field headers and field header synonyms to determine if a grid filter rule must be dynamically generated and executed against the application content. If the match attribute is not determined to be a field header or a field header synonym, the dynamic rule is not generated.

Resolving ambiguity

Ambiguity is a state in which there is more than one possible meaning in your query. For example, if you want to query the portal for John Doe's employee information, and you enter `Show employee info for John Doe`, the network cannot determine which agent should handle the request because you did not enter "show" into the agent network as a synonym for either the Execute or Send agent.

If there is ambiguity in your query and the agent network cannot resolve the query, you receive a list of applications from which you can select the one you want to use to execute the query.

If you are using multiple co-brands in your portal, Sybase recommends that you limit the number of applications you co-host within a single co-brand and use alternative co-brands to separate related applications to reduce the possibility of ambiguities.

When creating an Answers Anywhere application, click the “Check Conflicts” button to check for potential ambiguities with other applications in that particular co-brand. See “Answers Anywhere” on page 12.

Setting up Answers Anywhere for e-mail

To enable users to query UA using e-mail, you must first create and configure an e-mail account on your mail system for UA to check. For example, you can create an account called “askua.” You must then edit the *uadejima.properties* file to contain the inbound/outbound server name, as well as the account information for the e-mail box UA is checking. See uadejima.properties below for information about configuring the properties.

uadejima.properties

Use the *uadejima.properties* file to configure Answers Anywhere so users can query the portal using e-mail. This file is located in *%UA65%\tomcat\webapps\dejima\WEB-INF\classes*. Table 2-2 lists the properties you can set and a short description of each property.

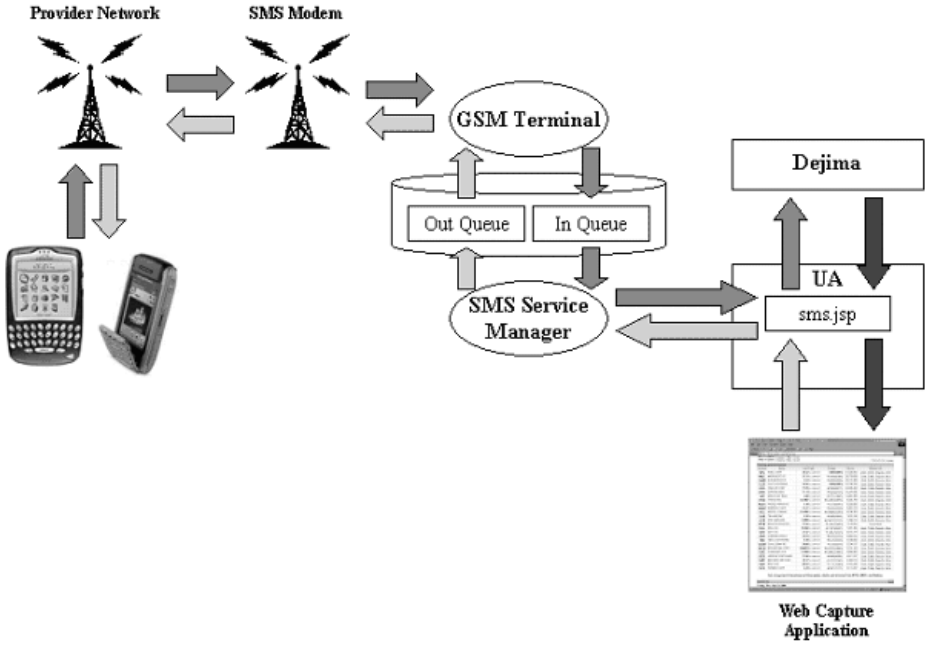
Table 2-2: uadejima.properties property settings

Property name	Description
smtp.host	Outbound e-mail server.
pop.host	Inbound e-mail server.
smtp.auth	true – authenticates outgoing e-mail. false – the default setting. Does not authenticate outgoing e-mail.
email.id	User account on the inbound e-mail server that UA reads from.
email.pwd	Password for UA to use to access the inbound mailbox.
ua.host	Host and domain name of the machine where UA is installed. For example, if the machine where UA is installed is named "machine" and the domain is "sybase.com," enter: `machine.sybase.com`
ua.port	The port UA is running on. For example, `4040`.
ua.rid	The resource ID used for Answers Anywhere integration. ua.rid is the default resource ID is no resource ID is specified in the client used to query Answers Anywhere.
default.opal	The location of the Answers Anywhere network. For example, *%UA65%\tomcat\webapps\dejima\dbds Unwired\wired.opal*.
role.check	Set to true to use role verification. Set to false if role verification is not needed. **Note** When you set role.check to false, role-based security is bypassed and an anonymous request can return data from any application, including applications that might contain sensitive information. If the portal contains data and applications that should not be publicly available to all users, set role.check to true.

Using SMS messaging

Users with SMS-enabled mobile phones or PDAs can send queries to a mobile phone or modem attached to a computer. Mobile modem control software on the computer then retrieves the message from the phone/modem and passes it to the Dejima components for processing. When the request is processed and a response is created, the modem control software sends the response back to the phone number that sent the request.

Figure 2-1: SMS messaging with UA Answers Anywhere



CellularModemController (CMC)

The CellularModemController (CMC) retrieves SMS messages from the phone/modem and passes them to the Answers Anywhere components for processing. When the request has been processed and a response created, the CMC sends the response back to the phone number that sent the request.

Note If you set up multiple co-brands in your portal, you must also configure different instances of CMC with each CMC instance configured to access the appropriate co-brand (RID).

Maintaining an access log

CMC enables you to maintain an access log that contains a subset of information for each SMS request received, such as:

- Requesting phone number
- Timestamp of request
- Request contents
- Response contents
- Response status

You can select which of these fields to record in the access log file. You can configure CMC to place the access log data in flat files or in a database file.

Administrative utility

The CMC includes an Web application administrative utility that allows you to:

- View and modify the configuration properties associate with the CMC, including:
  - Serial communication line where the modem is attached
  - Baud rate to use with the serial communication line
  - Frequency of polling to look for new messages
  - Maximum number of SMS messages to return when a response exceeds the 160-character limit of an SMS message.
- Contents of the files containing the white and black lists phone numbers.
- Access log configuration

- Error message viewing for the CMC process

Configuring the CMC

CMC uses the *CMCConfig.properties* file, located in *%SYBASE%\UA65\cmc\classes\com\sybase\cellmodem*. This file contains initial values that you can override with command line options. You can run the configuration file with the `run -help` command to see the details of all options. At the command prompt, enter:

```
run -help
```

See Table 2-3 for some of the more common options.

Table 2-3: CMCConfig.properties file settings

Option	Description	Default value
accessLog.File	Name of the file where access log data is kept.	None. If no file is specified, no access log is maintained.
databaseDriver	Fully qualified class name of a java.sql.Driver implementation for connecting to a database.	com.sybase.jdbc2.jdbc.SybDriver
databasePwd	Password to use when connecting to the database.	SQL
databaseURL	JDBC URL to the database containing tables used by CMC.	None. If this option is not specified, CMC does not attempt to connect to, or use any database.
databaseUser	User name to use when connecting to the database.	dba
httpURL	This is the HTTP URL that the HTTPClientMessageHandler class sends requests to with the request body message. This is the access point for Answers Anywhere processing. The sms.jsp page looks for an optional rid=<*N*> query parameter, where <*N*> is the RID number. For example, http://machine.sybase.com:4040/dejima/sms.jsp?rid=211. If this is not specified, the default RID configured in *uadejima.properties* is used.	http://localhost:4040/dejima.sms.jsp

Option	Description	Default value
instanceName	This is the name by which this instance of the CMC process and its modem are known. When you have more than one CMC, you must assign each CMC a distinct name so the configuration data is kept separate and the respective access logs can be distinguished. When CMC is using a database, this name is used in searches and inserts.	DejimaServer
logLevel	This determines the minimum severity level of messages to be placed in the logFile. Valid values are described in java.util.logging.Level.	WARNING
roleBasedAccess	This is a Boolean property. When set to true, CMC attempts to match a sender's phone number to a registered cell number in the user.cell_phone column of the portaldatabase. If a match is found, the login_name for that user is passed as a a parameter to the HTTPUrl for the Web application to use in applying access control.	false **Note** Use this option in conjunction with the *uadejima.properties* file role.based property. See Table 2-2 on page 21.

Security with e-mail, and SMS based request handling

E-mail requests are serviced for all registered UA users whose "from" e-mail ID matches the registration e-mail ID in the system. SMS requests are serviced for all registered UA users whose SMS phone number matches the phone number of a user registered in the UA system.

Role-based access to the executed application applies once it is established that the user is a valid UA user. Security roles are defined when you build the application in Mobile Web Studio. Nonregistered users' requests are serviced only for applications that have the "everyone" or "guest" roles.

The administrator can also set up an explicit list of phone numbers for people who are allowed access (a "white list") to CMC services. In the same way, the administrator can set up an explicit list (a "black list") that blocks access from certain phone numbers. If the user's number is found on the allowed list, processing of the query completes, if the user's number is not found, a message is sent that indicates the user does not have permission to access that particular service.

Custom application height

When you save an application, the Presentation tab in the Mobile Web Studio – New Application Finish window contains a Height option that lets you select the height at which an application displays.

The Height option is available only when Display Within IFrame is selected.

When the Height option is available, you can select from several predefined heights or select Add New. When you select Add New, an input field appears where you enter the application height as a numeric value, in pixels. The field accepts only numeric values.

❖ **Customizing an application's height**

1 Log in to Mobile Web Studio and create a new application. See the *Mobile Application Development Tutorial* for instructions on creating a new application.

2 When you finish creating the application, click Save in the Application Builder.

3 In the Mobile Web Studio – New Application Finish window, complete the options on the Content and Roles tabs.

4 On the Presentation tab, select Display Within IFrame.

5 Select Add New from the Height drop-down list.

6 In the input field that appears to the right of the Height option, enter the numeric value, in pixels, of the height you want for the application; for example, `200`.

7 When you complete the remaining options, click Finish.

Application content display direction

Unwired Accelerator lets you control the display direction of the content when you save an application or edit its properties in the Application Builder.

❖ **Customizing application content direction**

1 Log in to Mobile Web Studio, and create a new application. See the *Mobile Application Development Tutorial* for instructions on creating a new application.

2 When you finish creating the application, click Save in the Application Builder.

3 In the Mobile Web Studio – New Application Finish window, complete the options on the Content and Roles tabs.

4 On the Presentation tab, select Display Direction.

5 Select LTR (left to right) or RTL (right to left) from drop-down list.

6 When you complete the remaining options, click Finish.

You can change the content direction of an application any time by editing the application.

Navigation styles

Unwired Accelerator includes several navigation styles that allow you to format application content to display properly on a mobile device. These styles display in Mobile Web Studio only when you save a new page group or edit an existing page group; that is, the navigation style options are not available when you save or edit applications or pages.

Note Navigation styles also display in Portal Interface when you click the Manage Pages icon, and then click Edit next to a page group, to display the Save Page Group dialog box.

The *navstyles.xml* configuration file is located in *%SYBASE%\UA65\onepage\fw\properties*. The default navigation styles provided in UA 6.5 are:

- Full HTML and frame support (desktop browsers). This is the default.
- PocketPC/PDA HTML with frames like Pocket IE on PocketPC 2003.
- MbusinessClient/PDA HTML without frames, like the M-Business Anywhere client.
- Phone-WML – phones with Wireless Markup Language browser.
- Phone-HTML – phones with HTML browser.

For more information, see “Choosing a navigation style” in Chapter 11 “Building Page Groups,” in the *Developer’s Guide*.

Content style

Content style (device type) determines the application playback presentation. Each content style has a template associated with it. The default template for each of the content styles is defined in *oem.xml*. The Device Type drop-down list in the Application Builder displays the list of content styles. On application playback, the UWP applies the template associated to the content style and the content style is mapped to a UserAgent.

Note For Wireless Access Point (WAP) Phones, you can define a custom template for application playback by implementing an interface.

UserAgentMapping

UserAgentMapping.xml, located in *%SYBASE%\UA65\tomcat\webapps\onepage\fw\properties*, maps the UserAgents to the navigation and content styles. Several UserAgent mappings are defined by default. For example:

```
<UserAgentMapping>
  <UserAgent clientpattern="Mozilla">
    <NavStyle name="Default"/>
     <Content name="portal" type="text/html"/>
   </UserAgent>
<UserAgent clientpattern="MSIE" platform="PPC">
 <NavStyle name="PocketPC/PDA"/>
  <Content name="portal" type="text/html"/>
  <ResolutionHeader name="UA-pixels"/>
   </UserAgent>
```

The clientpattern and platform attribute values are from the client's User-Agent HTTP header. The User-Agent header usually contains the platform and browser information. In device detection, Unwired Accelerator matches the platform substring against the User-Agent header first.

The name attribute value in the NavStyle element is from *navstyles.xml*. This specifies the Navigation Style the UserAgent supports. The name attribute value in Content element is from *oem.xml*. This specifies the Content Style that application playback should use for the UserAgent.

❖ **Defining a custom template for Content Style**

1 Implement the interface *com.onepage.fw.uwp.interfaces.uwp.UWPMobileTemplate*:

```
String getTemplate
```

This returns the template string name, or if there are any errors, NULL. The template string name should match one of the names listed in the Template Manager in Mobile Web Studio.

2 Compile it. All the required JARs are in *WEB-INF\lib*.

Check the XML data by going to:

```
http://host.domain.com/onepage/servlet/UWPServlet?w
id=<wid>&content=text/xml
```

<*wid*> is the application ID you see in the application properties in Application Builder.

3 Move the implementation in the *WEB-INF\classes* directory.

4 Set "WAPCustomTemplate:<*class package*>" in the name attribute of the Content element. For example:

```
<Content name="WAPCustomTemplate:com.sybase.template.MyTemplate"
```

Defining a custom template example

In this example, the OpenWave Basic template is returned. The OpenWave Basic template is located in *%SYBASE%\portlets\templates\mobile\openwave.xsl*.

```
package com.sybase.ep.utils;

import com.onepage.fw.uwp.interfaces.uwp.*;
import javax.servlet.http.*;
import java.io.*;

/**
 * Sample custom template for openwave browsers
 */
public class OpenWaveTemplate implements
UWPMobileTemplate
{
    public String getTemplate (UWPRequest request)
```

```
        {
            HttpServletRequest req =
    (HttpServletRequest)request.GetServletRequest();
            String agent = req.getHeader("User-Agent");
            if (agent.indexOf ("UP.Browser") >= 0) {
                return "OpenWave Basic";
            }
            else {
                return null;
            }
    }
```

For this example, the custom template is used on OpenWave browsers, so you must modify the content in *UserAgentMapping.xml* to say:

```
<UserAgent clientpattern="UP.Browser" platform="UP.Browser">
    <NavStyle name="Phone-WML"/>
    <Content name="WAPCustomTemplate:com.sybase.ep.utils.OpenWaveTemplate"
type="text/vnd.wap.wml"/>
</UserAgent>
```

Multiple template support

In Unwired Accelerator, you can associate multiple templates with a single application.

Each template is associated with a particular viewer, such as a Web browser or a mobile device. The following basic templates are provided:

- Portal – template used for standard browsers, such as Internet Explorer. The default template is OP 1-1 Basic.

  To preview application templates, click Template in Application Builder, click Search in the Find Template window, select the template in the Results pane, then click Preview.

- PDA – template used for mobile devices. The default is PDA Grid Basic.

  To preview a PDA template in Application Builder click Preview.

- BlackBerry – template used for BlackBerry devices. The default is BlackBerry.

To preview a BlackBerry template in Application Builder, select BlackBerry for Device Type, click on the "v" button next to the Template, select Mobile, then click Preview.

- WAP-HTML – template used for WAP phone clients that support HTML.

  To preview a WAP-HTML template in Application Builder, click Preview.

- WAP-WML – template used for WAP phone clients that support WML 2.0.

  To preview a WAP-WML template in Application Builder, click Preview.

You can change templates on an application-by-application basis.

Note The default templates are defined in *%SYBASE%\UA65\tomcat\webapps\onepage\config\oem.xml*, within the `<ViewerList></ViewerList>` section.

❖ **Selecting a different template**

1 When creating an application in Application Builder, select the Device Type—Portal, PDA, WAP-HTML, WAP-WML, or BlackBerry.

2 Click Template. The Template finder window displays.

3 Using the template finder, select the required template.

4 Click OK to select the template.

Application update links

When you create an application to display on a PDA or BlackBerry client application, and you want to give users the ability to update the application's source content from that PDA, link the source application with an update application, which acts as a queue to store updates on a PDA until the PDA is resynchronized with the local host computer on which Unwired Accelerator is installed.

First, you must create the update application, then you must create the primary application with a link to the update application (see Chapter 5, “Creating a Multipage Mobile Application with Transaction Support” in the *Mobile Application Development Tutorial* for instructions on creating primary and update applications). For conflict resolution to work correctly on your update, you must also create a verification application and link the update application to it. The verification application must return a grid data set using a select statement based on the values of one of CGI parameters used as the search parameter. It must also include a header row that defines the columns of the grid. The column names must match the update parameter in the update application.

Note If conflict resolution is not an issue for you, there is no need to create a verification application. All updates take effect regardless of whether or not the value to be updated is current.

Mobile applications

Accessing mobile applications in connected mode

You can access mobile applications any time using your Smartphone, PDA, or BlackBerry device. The URL for accessing mobile applications is:

```
http://machine.domain/onepage/mp.jsp
```

For example, if the name of the machine where Unwired Accelerator is installed is named “ua,” and your domain is “sybase.com,” enter:

```
http://ua.sybase.com/onepage/mp.jsp
```

Users are grouped under resources. Each resource corresponds to an Unwired Accelerator co-brand. The index JSP (*mpindex.jsp*) accesses resource ID (RID) rid=21 from desktop browsers such as Internet Explorer. For more information about resource IDs, see Chapter 14, “Creating Co-brands” in the *Enterprise Portal Developer’s Guide*.

In the Welcome window, enter your mobile portal user name and password. The next window (the home page) displays the navigation tree with mobile applications displayed as links. The navigation tree shows the names of the page groups, pages, and applications. Only those page groups that support the navigation style of the requesting device are listed in the home page. See “Navigation styles” on page 27. To see the content of an application, click the appropriate link in the home page.

Running applications in disconnected mode

You can run applications in disconnected (offline) mode with the Anywhere Client 5.5 for Symbian P900, PocketPC, and Palm devices. Install the M-Business Anywhere Client browser, and configure the server for sync operations. See your M-Business Anywhere documentation for instructions.

Note The only types of disconnected mode applications that you can use on the P900 are spidered Web based applications.

M-Business Anywhere

Use the M-Business option on Mobile Web Studio to administer a subset of M-Business features, including:

- Managing channels – create, edit, and delete M-Business Anywhere channels for Web content, organized by categories, or public channels.
- Managing group applications – create, edit, and delete M-Business Anywhere groups to manage how you assign channels to your users.
- Managing users – create, edit, and delete M-Business Anywhere users from Mobile Web Studio.

The *Mobile Application Development Tutorial* contains sample procedures that illustrate how to use the M-Business Anywhere functionality.

BlackBerry device

Using the BlackBerry browser on devices connected with an MDS module residing in the BlackBerry Enterprise Server (BES), you can have a mobile portal. The BlackBerry client container application provides includes applications that can be used even when there is no connectivity. The client container application uses the BlackBerry Java Development Environment (JDE), which includes a complete set of APIs and tools that allows developers to develop custom Java application for the BlackBerry Wireless Handheld. After installation and deployment, the Custom Client Application for UA icon displays on the screen of your BlackBerry device.

BlackBerry device trackwheel menu

The UA application uses the standard BlackBerry device navigation model. The UA applications trackwheel has these menu items:

- Open – allows you to open an application on the device.
- Delete – allows you to delete an application on the device.
- Settings – allows you to configure and save information for connecting to the UA server.
- Sync – allows you to sync a single application content to the device.
- Sync All – forces the sync of all available application content to device.
- Logs – shows transaction logs from previous synchronization.
- Delete data – deletes logged data and all user settings.
- About – shows a screen with the Client Container Application information and version.
- Close – closes the J2ME application and returns you to the program desktop screen.

When you display an application, these menu items are also seen:

- Details – shows the details of an application.
- Click Thru – allows you to click through links in an application.
- Sort – allows you to sort columns in an application.
- Edit – allows you to edit applications.

Connection settings for synchronization

Select the Settings menu to go to the Connection Settings window. Enter:

- User name – this is the user name you use for logging in to UA.
- Password – the password you use to log in to UA.
- Resource ID – a drop-down list with names of the registered resources and co-brands.
- UA Server Name – fully qualified name of the machine where UA is installed.
- UA Server Port – port number used to connect to UA.

See “Retrieving applications on the BlackBerry device” in the *Mobile Application Development Tutorial* for the procedure for retrieving applications on your BlackBerry device.

BlackBerry Application Template Customization

You can customize the presentation of data on your mobile device by using the Mobile Application Template Customization window in Mobile Web Studio.

❖ **Launching the BlackBerry Application Template Customization wizard**

1 In Mobile Web Studio select Build | Templates |Approved.

2 Right-click your mobile device from the list, for example BlackBerry, and select New Template.

3 The Mobile Application Template Customization window launches.

The Mobile Application Template Customization window contains four tabs with the following options:

BlackBerry device tab Use the Device tab to name your device and assign user roles.

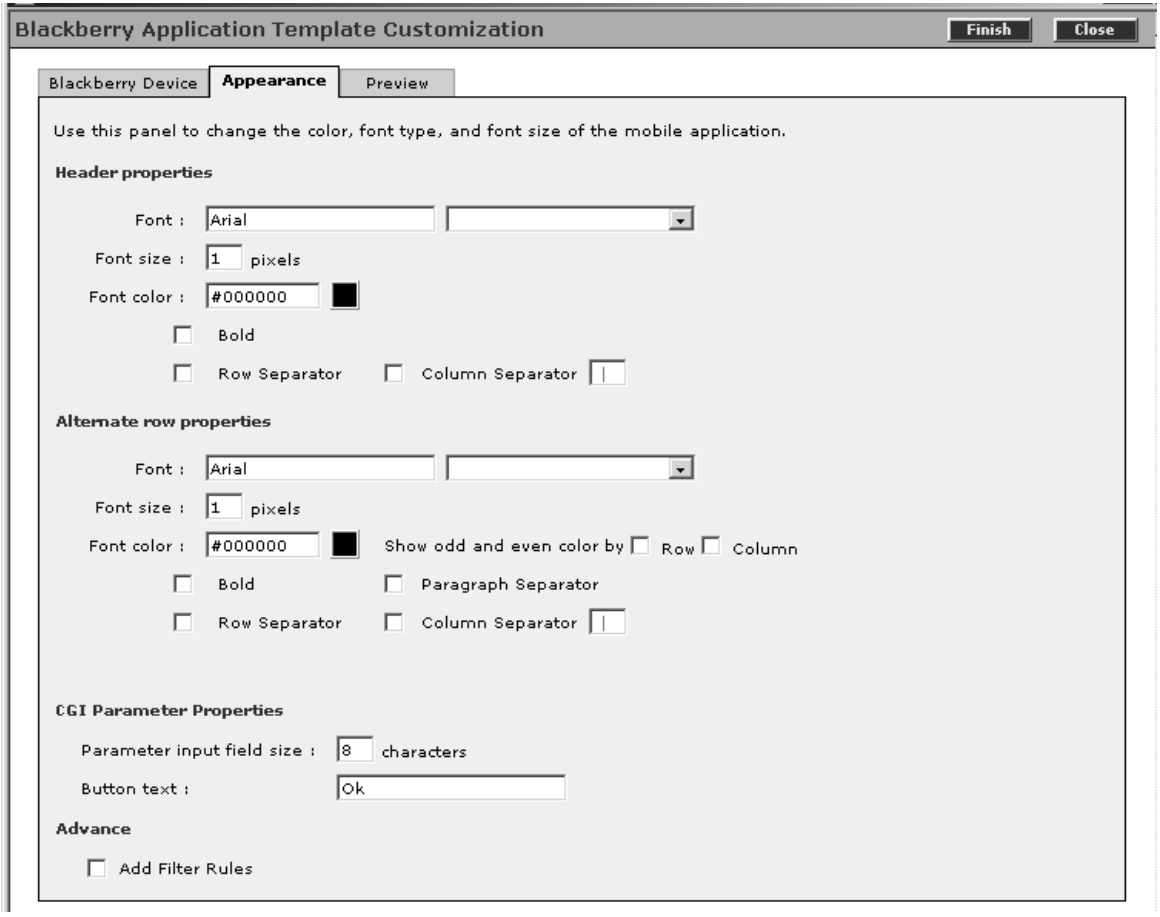
Appearance tab Use this tab to customize the properties of the presentation for your mobile device. The options are:

- Header Properties – apply only to the first row of data.
  - Font – select from the drop-down list.
  - Font size – enter the desired font size in pixels.
  - Font color – select the color box to the right to see a selection of colors.
  - Bold – select for bold.
  - Row Separator – select to add a horizontal spacing line.
  - Column Separator – select to add a vertical spacing line.
- Alternate Row Properties – applies to all other rows of data (not the header row).
  - Font – select from the drop-down list.
  - Font size – enter the desired font size in pixels.

- Font color – select the color box to the right to see a selection of colors.
    - Show odd and even color by
        - Row, or
        - Column
    - Even Col Color – select the color box to the right to choose the color for the even columns.
    - Odd Col Color – select the color box to the right to choose the color for the odd columns.
- Bold – select for bold.
- Row Separator – select to add a horizontal spacing line.
- Paragraph Separator – select to put a space between each row.
- Column Separator – select to add a vertical spacing line.

- CGI Parameter Properties
    - Parameter Input Field Size
    - Button Text
- Advance
    - Add Filter Rules

Figure 2-2: Mobile application template customization



Filter tab This tab appears only if you selected Add Filter Rules in the Appearance tab. Select Enable Filter For List Data.

- Filter Value in Between String
- Filter Value in End of String

Preview tab Use this tab to preview the data presentation. The preview is not an exact representation of what you will see on your mobile device.

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

Update and conflict resolution

You can perform connected (online) or disconnected (offline) updates of records from PDA or BlackBerry client applications. The *Mobile Application Development Tutorial* includes instructions for performing an update.

Having multiple copies of the same data on multiple mobile devices on which multiple users can update and perform sync operations introduces the problem of one user's updates cancelling modifications made by other users. Unwired Accelerator provides concurrency control and conflict resolution to solve this problem.

Updating records from PDA applications

When you make changes on record data from your PDA device while running applications in disconnected mode, you can check the status of your update operation. You can confirm whether the update operation succeeded, and if not, you can find out why it failed. The Mobile Portal Server maintains a log of all update requests it receives from the M-Business server during sync operations. Logs are created in the portaldatabase.

When you perform sync operations, update logs are sent to the PDA device. Click Log in the application pages to see a list of all sync log records. The List page of the logs shows only a subset of the fields from the log record. To see more details, click any column of the log record to display the detail page.

Use the Delete button on the detail page to delete log records. Delete log records periodically, otherwise, the list grows and increases your sync times. When you delete a log record, it is immediately removed from the device database and no longer appears in the log record list. However, the records are not deleted from the portal server until the next sync operation.

Conflict resolution

Unwired Accelerator uses optimistic concurrency control. This allows users to make changes to records on their local devices, but at sync time, the mobile portal server rejects an update operation if the data submitted for modification has already been modified by another user in the enterprise. For example, if four users change the same record and perform the sync operation, only the first user's sync is executed, and the other three users' update requests are rejected.

During sync, the PDA devices return the original value (the value before the change) of the updated record columns, the new value for the columns, and all the search parameter values defined in the updated application, to the linked application of the mobile portal server. See “Application update links” on page 31.

Continuous capture

The continuous capture feature enables application developers to remotely capture a set of Web pages that are connected (or related) to each other, and define how to extract and display the content from the various pages within Mobile Web Studio. You can navigate through this set of pages using embedded hyperlinks, can preserve the original Web site’s page structure, and can redefine how to present the captured pages.

Continuous capture allows you more control over content and presentation of captured Web sites. For example, you can extract tabular data and present it in a structured, grid-oriented layout, and build drill-down applications. This allows you to create applications that use click-across functionality more effectively, and also build applications that can then be captured as a whole and delivered to a mobile device.

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

Capture and playback enhancements

UA 6.5 has several new options for capture and playback.

Note These options work only with English language character sets.

Navigation-time ActiveX (NavACX)

Because of the numerous technologies that are used, you cannot always navigate to every possible page and successfully create applications from all Web accessible sources. You may not be able to navigate to some sites or pages within sites. You may also not be able to create an application using the exact piece of information that is viewable on a Web page. Especially challenging are Web sites that use JavaScript to generate some or all of an HTML page, and sites that use frame-based solutions.

To help overcome this challenge, a new option, NavACX, is available when performing a capture. You can now select from these formatting options:

- CapALL
- NavACX
- ACX

You can select none, one, two, or all three of these options.

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

To use this functionality, when creating a new application, in the Mobile Web Studio – New Web Element window, select NavACX from the format options drop-down list to the right of the Format drop-down list.

Note Selecting NavACX does not automatically result in server-side ActiveX being used at playback. Server-side ActiveX is used only when either DOM or ACX is also selected.

Document object model (DOM)

DOM uses server-side ActiveX during playback, and allows you to locate content on a Web page that uses the HTML layout more explicitly. The DOM strategy is an alternative capture strategy that makes use of the Document Object Model used by many browsers when displaying HTML page content. This Document Object Model and its published API is used by UA in conjunction with server-side ActiveX technology to query, modify, and extract desired page content. When you select the DOM format, you see a sequence of captures from the page. The captures start from the HTML feature closest to the selected item, and roll outwards. For example, clicking on some text embedded in a row of a table, embedded in a page, results in the text being displayed, then the row, then the table, and then the page. The actual "unwinding" depends on the actual HTML structure of the page.

This capture strategy is most effective when the content of the site is static and stickiness of the content is not an issue. Often, it is the most effective approach when defining the second and subsequent captures in a Continuous Capture portlet.

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

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

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

@OP tag usage in URLs

You can use @OP tags as part of the URL to be altered at playback, and these are substituted just as they are for @OP tags in other situations, for example as part of SQL query strings. If a URL contains query parameters, the normal parameter handling code allows you to mark one or more of these query parameters so that their values can be substituted at portal playback.

You cannot, however, use the @OP tag to replace any part of the scheme, host, or port portion of the URL. For example, you cannot use the @OP tag this way:

```
http://@OP["host"="machine.sybase.com"]:4040/path/index.html
```

You can use the @OP tag in the URL this way:

```
http://machine.sybase.com:4040/@OP["path"="whatever"]/index.html
```

The @OP tag must contain a suitable default value, as this is used to determine the initial navigation point.

Improved JSESSIONID handling

Many Web sites (especially those using J2EE application servers) use the JSESSIONID approach to managing sessions. This involves sending a cookie (JSESSIONID) during the login process, and the browser generating a URL request containing the string ';jsessionid=*nnnnnnnn*' in return. The cookie value and the URL value must match.

Cookies are sent dynamically, so to prevent future playbacks from generating URL requests with the wrong ";jsessionid=" string embedded, when the playback engine sees that a URL has the string ";jsessionid=" as part of the URL, it automatically replaces the value of the JSESSIONID parameter with the value sent in the JSESSIONID cookie. This ensures that URLs sent in CCL playback are consistent with the actual JSESSIONID value.

New @OP cookie tags

In some cases, session values are included in URLs, but not through the ";jsessionid=" approach. A mechanism has been introduced that allows you to replace tagged sections of the CCL with the content of cookies in the current session cookie JAR. This works by allowing you to define runtime substitution of parameter values.

The @OP cookie tag allows you to specify runtime replacement of query parameters using the value of a specified cookie. You can do this by using this syntax:

```
@OPCOOKIE["<cookie name>"]
```

where <*cookie name*> is replaced by the name of the specified cookie. For example:

```
@OPCOOKIE["JSESSIONID"]
```

There is an alternative form, @OPCOOKIEENC, which URL encodes the cookie value when substituted. The syntax is otherwise the same.

File element

Unwired Accelerator lets you create application elements that are based on files. The file must contain data arranged in a format that can be mapped to grid data (see "Grid rules" on page 53). For example:

- Delimited columns – a file that has data separated by commas, tabs, semi-colons, colons, or spaces. Typically, this is a file with a *.csv* extension, which you can create when you save the contents of a spreadsheet application.
- Fixed-size columns – a file with data arranged in columns of a fixed size. These files can be saved and uploaded with a *.txt* extension.

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

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

❖ **Creating a delimited file element**

This example shows how to create an application from a file source that uses commas to separate the data. You can also create a file that uses tabs, semi-colons, colons, or spaces to separate data.

1 Use a spreadsheet application to create a four-column table in a spreadsheet program with the following content:

Col1	Col2	Col3	Col4
1	2	3	4
999	888	777	666

2 Save the content as a comma-delimited file in a temporary directory, with the name *data.csv*; for example, *C:\temp\data.csv*.

3 Log in to Web Studio, select Build Applications, select New from the Application Manager Status menu, then click New to launch Application Builder.

4 Click the arrow to the right of the Add button and select File Element. You see the New Element window, which allows you to identify the file to upload from your local machine to the Unwired Accelerator server.

5 Complete the File Element Definition options:

- File Name – enter the file name of your source data (*C:\TMP\data.csv*).
- Delimited, Fixed Width – reflects how source file content is arranged—data separated by a delimiter (such as a comma), or data arranged in fixed-sized columns.

  For this example, select Delimited.
- Destination Location – accept the default location, which is created automatically.
- File Name – leave blank. This field is automatically filled in once you upload the file.
- Delimiter – select the delimiter used in the source data file. Select "Comma."
- Merge Repeated Delimiters – whether repeated delimiters should be replaced with a single delimiter. This is useful if your source data includes blank columns. For this example, leave this unselected.
- Text Quoter – specifies the quote character that is used to define a section of "as is" text that may include delimiters. Accept the default.

Click Upload to the right of the Destination Location field. When the confirmation displays that the file was successfully uploaded, click OK.

6 Click Preview to see the result.

7 Click Next.

8 In the next three windows (Split, Define, and Filter) click Next to bypass the fields, which are not necessary for this example.

9 In the Window Preview window, enter `fedata` for the file element.

10 In the Continuous Capture window, click Finish.

11 In the Application Builder, click Save.

12 In the Finish window, under the Content tab, in the Name field, enter `fedata`.

Accept all the defaults, and click Finish.

13 In the confirmation pop-up, click OK.

❖ **Creating a fixed-column file element**

This section describes how to create a file element using fixed column width data; that is, the source file data is arranged in columns of fixed length.

1 Use a text editor to create a four-column table with the following content:

Col1	Col2	Col3	Col4
1	2	3	4
99	44	1	12

2 Verify that the number of spaces between each column is the same; for this example put 9 spaces between each column.

3 Format the text using a monospaced font—for this example, select Lucinda Console.

4 Save the content as a text file (**.txt*) in a temporary directory, with the name *data.txt*; for example, *C:\temp\data.txt.* Exit the text editor.

5 Log in to Web Studio.

6 Select Applications from the Build menu in the left pane, select New from Application Manager Status menu, then click the New button to launch Application Builder.

7 Click the arrow to the right of the Add button and select File Element. You see the New Element window, which allows you to identify the file to upload from your local machine to the Unwired Accelerator server.

8 Complete the File Element Definition options:

- File Name – enter the file name of your source data (*C:\temp\data.txt*), or click Browse to locate the file.
- Delimited, Fixed Width – reflects how the source file content is arranged—data separated by a delimiter (such as a comma), or data arranged in fixed-sized columns.

  For this example, select Fixed Width.
- Destination Location – accept the default location, which is created automatically.

- File Name – leave blank. This field is automatically filled in once you upload the file.
- Columns (comma separated) – indicates where to place the columns for the data source; for this example, `0, 9, 18, 27` (this indicates that each column has nine spaces between it and the next).

Click Upload to the right of the Destination Location field. The program creates a *\PortalDocs* subdirectory beneath the directory you have specified as the temporary directory on the local machine. The file you upload is saved in a subdirectory beneath *\PortalDocs* that has your user name; for example, *x:\temp\PortalDocs\opsuper*, where "*x*" is the drive where the temporary directory is located, and "*opsuper*" is the name of the user that uploaded the file.

When the confirmation displays that the file was successfully uploaded, click OK.

9 Click Preview to see the result.

10 Click Next.

11 In the next three windows, click Next to bypass the fields (Split, Define, and Filter), which are not necessary for this example.

12 In the Window Preview window, enter `fixed width` for the file element name.

13 In the Continuous Capture window, click Finish.

14 In the Application Builder, click Save, complete the options, then click Finish.

15 In the Finish window, under Content tab, in the Name field, enter: `fixed width.`

Click Finish.

16 In the pop-up confirmation window, click OK.

Database element JNDI connection cache option

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

When you create a database element, you can use one of the default JNDI resources included with Unwired Accelerator, or you can create a new resource.

Using a default JNDI data source resource

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

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

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

Creating a new JNDI data source resource

To use a new JNDI data source resource, an entry must exist for the connection cache in the <Resource name> section of *server.xml* file located in *%SYBASE%\UA65\tomcat\conf*.

server.xml

Here is the entry in *server.xml* for the portal database:

```
<Resource name="jdbc/portaldb" auth="Container"
    type="javax.sql.DataSource"/>
        <ResourceParams name="jdbc/portaldb">
            <parameter>
                <name>driverClassName</name>
                <value>com.sybase.jdbc2.jdbc.SybDriver</value>
            </parameter>
            <parameter>
                <name>driverName</name>
                <value>jdbc:sybase:Tds:localhost:4747?
```

```
            ServiceName=portaldatabase</value>
        </parameter>
        <parameter>
            <name>user</name>
            <value>dba</value>
        </parameter>
        <parameter>
            <name>password</name>
            <value>SQL</value>
        </parameter>
        <parameter>
            <name>maxActive</name>
            <value>20</value>
        </parameter>
        <parameter>
            <name>maxIdle</name>
            <value>10</value>
        </parameter>
        <parameter>
            <name>maxWait</name>
            <value>20000</value>
        </parameter>
</ResourceParams>
```

On Tomcat, *server.xml* is located in *x:\UA65\tomcat\conf\server.xml* on Windows, where "*x*" is the drive where Unwired Accelerator is installed. To create a new entry, see the instructions at http://jakarta.apache.org/tomcat/tomcat-4.1-doc/jndi-datasource-examples-howto.html.

Creating a database element using a JNDI data source resource

This example uses the Unwired Accelerator sample database to illustrate how to use a JNDI data source resource in the definition of a database application.

❖ **Creating a database element using a connection cache reference**

1 Log in to Web Studio, select Build | Applications, then click New to access the Application Builder.

2 When the Application Builder displays, click the down-arrow to the right of the Add button, and select Database Element.

3 When the Web Studio – New Web Element window appears, select the Connection Cache option.

4 Complete these fields:

- Conn Cache Name – from the drop-down list, select the name of the connection cache for the sample database:

  ```
  sampledb
  ```

- SQL Query String – enter this SQL statement as one line:

```
select c.lname, p.name, si.quantity, si.id, si.prod_id from customer
c, sales_order s, sales_order_items si, product p where c.id =
s.cust_id and s.id = si.id and si.prod_id = p.id
```

5 Click Preview. You see the data that corresponds to the SQL query.

6 Click Next.

7 On the next three windows (Split, Define, and, Filter), click Next to bypass the options. See the "Data-capable post-processing options" section in Chapter 5 of the *Enterprise Portal Developer's Guide* for descriptions of these options.

8 In the Window Preview, enter `Sales Update` for the Element Name, then click Next.

9 In the Continuous Capture window, click Finish. Do not click Continue in this window.

10 Back in the Application Builder, click Save and complete the options on the Web Studio – New Application Finish window.

11 In the Finish window, under the Content tab, in the Name field, enter `salesup`.

Accept all the other defaults, and click Finish.

12 In the pop-up confirmation, click OK.

Server-side click-across applications

Unwired Accelerator click-across functionality lets you create events that start when a user clicks the content in an application table. For example, a user can click a grid value in one application to refresh another application's content on the page using the grid value as a parameter to be applied to the refreshed application.

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

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

When you define click-across events you can now place the event on a subset. For example, when you define a click-across event, you can enter "2-" in the Row field, which means that every record from record 2 to the last record is included in the event, and the header record is skipped.

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

General click-across improvements

You can define an event that takes its input value from any cell in the grid, not just from the cell in which the event is defined. The source cell can even be in a hidden column.

To display information from an underlying data store, and build a drill-down application that enables users to select a particular item for more information, the item's key can be a generated identifier used to look up the details. However, you may want to display a more meaningful description of the grid item. To support this, build a grid containing the generated identifier in one column, and the description in another column. By hiding the first column, and building an event on the second column, you can create a compact application that enables you to easily drill down to the details section.

See "Grid rules" on page 53 for more details on hiding grid data.

Continuous capture versus server-side click-across

Continuous capture applications and applications that contain server-side click-across events provide similar results—the ability to create an application with embedded links that display new content when clicked. For example, an application may have a link that returns more detailed information about an item on the top-level application when you click the link. However, there are differences between these two approaches.

Continuous capture allows you to create a set of related pages, based on one source Web site, which is defined within a single application. Individual pages are accessed through links on the top-level page. As users click links, the source Web site context is maintained, including cookies, and content is extracted to each page using the continuous capture information stored when the application was created. Continuous capture works for remote Web site capture.

In contrast, server-side click-across is built around separate, independent applications. The content of each application is defined separately, and context, such as cookies, is maintained separately. This means that the context is not preserved as you move from application to application. Each step establishes a completely separate session with the back-end Web server. Because each application is separate, you need not build them from the same Web site.

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

Charts

Unwired Accelerator 6.5 allows you to create drill-down charts using the charting post-processing wizard. Drill-down charts use server-side click-across to link multiple charts together into a cohesive application.

To create one drill-down charting application from two chart applications, there must be a relationship between the category or series names of the first chart and the parameters used in defining the second chart's data source.

See the *Mobile Application Development Tutorial*, which contains a tutorial that illustrates the use of drill-down charts using server-side click-across. The tutorial contains procedures to create a table of data with the stock market's most active stocks and uses drill-down charting to display the quote page with details of a selected stock.

Importing and exporting objects

Mobile Web Studio allows you to import and export single applications, catalogs, composite applications, pages groups, pages, and agents to Portal Interface. To export or import an application, it must use a nonmobile template.

Note See "Importing, Exporting, and Deploying Portal Objects," Chapter 13 in the Enterprise Portal Developer's Guide for general instructions on using import and export functionality.

Restrictions

XML validation XML validation must be turned off for the import and export functionality to work:

1 Open *global.properties.xml* in a text editor. The default location of this file is *x:\UA65\tomcat\webapps\onepage\config*, where "*x*" is the drive on which the product is installed.

2 Find <Property name="XmlValidation"> and verify that the value is set to "off." If it is not, change it.

3 Save the file and close the text editor.

4 Restart the application server.

Multiple element applications To export an application to the Mobile Web Studio, which is used with M-Business Server to form the Unwired Accelerator product, the application can have only one element. Multiple-element applications are not compatible as mobile applications.

Templates The JSP template is the only valid template that can be used for exporting or deploying an application to M-Business Server.

Grid rules

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

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

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

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

Grid rules may be useful in situations such as these:

- Executive dashboard – create a concise financial report from general data. For example, use a Web page of financial report data that targets a wide audience as the source, create rules that eliminate unnecessary rows and columns, and add “bottom-line” information for an executive audience.
- Aggregate data – create aggregate data from various raw data sources. For example, use raw data accessible in a database as the source, and create rules that combine and present the data in a form that is more useful to a particular audience.
- Graphical presentation – create a chart or Flash presentation of data. For example, use an existing Web report as the source, and create rules that restructure the report for graphical presentation without manipulating the data. The grid rules provide flexibility for extracting data that can be used in the charting layer.

“include” options

When you capture or retrieve grid-based data, the program uses a default “include” logic that assumes you want to include all of the data you captured or retrieved. This logic remains, even when you specify to exclude records or fields, insert record or fields, or edit records.

However, when you select any “include” option from the first Add Filter Rule drop-down list, the default “include all” no longer applies, and you must explicitly specify the records to include.

Therefore, to hide records or fields from the display, first use Include Records to specify the records to include, then use Include Hidden * to specify the record or field to hide.

Warning! If you select Include Hidden Records without explicitly specifying the records to include, the final preview of the portlet is blank. Because you have not explicitly stated which rows of data to include, the only data that is actually there is hidden, resulting in a blank display.

Using grid rules

This section describes grid-rule functionality that is useful for creating mobile applications. Topics include:

- Insert rules – hidden records or fields
- Edit rules – images and values
- Using @OP tags with grid rules

Insert rules – hidden records or fields

When inserting either records or fields, you can specify to insert a hidden record (or field). Such an insertion marks the record (or field) as hidden, so it does not display in the application. However, its contents can be used for event definitions.

Edit rules – images and values

When you are editing records, there are two options for defining content of target cells:

- Specify an image to be inserted into the cell.
- Specify an explicit value, or a value based on a formula, that can be applied to a range of cells in the grid.

When selecting an image type, you can enter a path to the image relative to the application server hosting Mobile Web Studio. For example, for the *icon_arrow.gif* under the *images* directory in the onepage Web application directory, enter: */onepage/images/icon_arrow.gif*. You can also specify a full HTTP URL reference.

When selecting a value type, you can use a syntax that allows you to specify ranges of existing cells, and perform functions (nested if required) on these cells. The syntax is:

`<command>(args[, ...])`

where `<command>` is the operation to perform, and `args` are the arguments to use.

Commands include:

- sum – sums the arrays of cells specified.
- avg – calculates the average of arrays of cells.
- int – returns the integer portion of argument.
- diff – determines the difference between two arguments.
- div – divides two arguments.
- concatenate – concatenates all the specified cells.
- concatenate2 – concatenates all the specified cells, inserting pad characters.
- min – determines minimum of specified arrays of cells.
- max – determines maximum of specified arrays of cells.

Note These commands are described in detail in the sections that follow.

Arguments include:

- Scalar – specifies a single-element value (either some literal text, or cell content).
- Array – specifies a range of cells, which can be one- or two-dimensional.

To specify single-cell content, use this syntax:

```
$R<x>F<y>
```

where <***x***> and <***y***> are indexes into the cells. For example, $R4F5 specifies record 4, field 5.

You can use the special value 0 as an index to indicate the current value. For example, if you are editing record 6, field 9, $R0F9 refers to record 6, field 9. This is especially useful for rules that edit a number of records or fields at once; it enables you to use a single rule to perform the same operation on a range of cells.

Array arguments use the single-cell syntax, combined so that a range of cells is specified. That is:

```
$R<x>F<y>[:R<x'>F<y'>]
```

If the second cell specifier is not present, the array consists of a single cell; if the second cell is specified, the array consists of all the cells in the indicated range. For example, $R5F2:R6F3 indicates records R5F2, R6F2, R5F3, R6F3. Again, use 0 as an index to use the current record or field.

Note If a function is expecting an array argument, a scalar is accepted and treated as a single cell array. However, if a function is expecting a scalar argument, an array argument cannot be accepted.

For all functions, you can use the return value of another function as a valid argument.

sum

Description

This function sums all arguments, and returns a single result.

Arguments

There can be one or more array arguments. Summation is carried across all the arrays specified (or, if scalars, by simply adding the scalar value to the current summation). Non-numeric values are ignored. The result is returned as a floating point value.

Examples

```
=sum($R3F4:R7F6)

=sum($R6F4, $R10F8)

=sum(5, $R6F7)
```

avg

Description

This function sums all arguments, divides the result by the number summed, and returns a single result.

Arguments

There can be one or more array arguments. Summation is carried across all the arrays specified (or, if scalars, by simply adding the scalar value to the current summation). Non-numeric values are ignored, and not added to the count of items summed. The result is returned as a floating point value.

Examples

```
=avg($R3F4:R7F6)
=avg($R6F4, $R10F8)
=avg(5, $R6F7)
```

int

Description

This function returns the integer portion of a single scalar argument.

Arguments

There can be only one scalar argument.

Examples

```
=int(2.5)
=int(=sum($R6F4:R10F8))
```

diff

Description

This function subtracts the second argument from the first argument.

Arguments

There can be only two scalar arguments.

Examples

```
=diff($R3F4, $R7F6)
=diff($R3F4, $R7F6)
=diff(5, 3)
```

div

Description

This function divides the first argument by the second.

Arguments

There can be only two scalar arguments.

Examples

```
=div($R3F4, $R7F6)
=div($R6F4, 5)
=div(10, 2)
```

concatenate

Description

This function concatenates all arguments, and returns a single result.

Arguments

There can be one or more array arguments. Concatenation is carried across all the arrays specified (or, if scalars, by simply concatenating the scalar value to the current result).

Examples

```
=concatenate($R3F4:R7F6)
=concatenate($R6F4, $R10F8)
=concatenate(5, $R6F7)
```

concatenate2

Description

This function concatenates the second and subsequent arguments, using the first argument as a pad, and returns a single result.

Arguments

The first argument is scalar, and specifies the pad string to be used; there can be one or more subsequent array arguments. Concatenation is carried across all the arrays specified (or, if scalars, by simply concatenating the scalar value to the current result). The pad value is inserted between all concatenated values (but not appended or prefixed).

Examples

```
=concatenate2(abc, $R3F4:R7F6)
=concatenate(-, $R6F4, $R10F8
```

min

Description

This function determines the minimum value across all the arguments, and returns a single result.

Arguments

There can be one or more array arguments. Determination is carried across all the arrays specified (or scalar values, if used). Non-numeric values are ignored. The result is returned as a floating point value.

E.xamples

```
=min($R3F4:R7F6)
=min($R6F4, $R10F8)
=min(5, $R6F7)
```

max

Description

This function determines the maximum value across all the arguments, and returns a single result.

Arguments

There can be one or more array arguments. Determination is carried across all the arrays specified (or scalar values, if used). Non-numeric values are ignored. The result is returned as a floating point value.

Examples

```
=max($R3F4:R7F6)
=max($R6F4, $R10F8)
=max(5, $R6F7)
```

Using @OP tags with grid rules

Another useful feature is the ability to use @OP tags when defining grid rules. At runtime, the @OP tag is replaced with a value passed to the application as an input parameter. @OP tags are useful for building SQL queries, because they enable an application to use queries with parameters. Several tutorials in the *Mobile Application Development Tutorial* use the @OP parameter tag with SQL queries.

You can use the @OP tags in grid rules enabling applications to use input parameters as a way of altering the runtime behavior of the rules associated with the application.

Customizing mobile application templates

When you deploy mobile applications to a mobile device such as PDAs (Palm OS, PocketPC, etc) or BlackBerry device, the mobile device displays the embedded data in grid form. You can alter the appearance of these grids using the Mobile Application Template Customization feature, which provides options for presenting data on the PDA, including various color schemes, number of rows to display on each page, and so forth.

The *Mobile Application Development Tutorial* includes sample procedures for customizing mobile application templates for both PDAs and BlackBerry devices.

Template parameters for customizing mobile application templates include:

Device Change the template name, mobile application resolution, and assigned roles. See “Device tab” on page 60.

Appearance Change the font color, font type, and font size for the mobile application. See “Appearance tab” on page 60.

Column Set the display length for column cells. See “Column tab” on page 62.

Paging Specify the number of data rows to display on the grid, and define the appearance of the paging buttons. See “Paging tab” on page 63.

Search Indicate whether to show the search mechanism, and to specify the appearance of the search buttons. See “Search tab” on page 64.

Update Indicate whether to update cell data, and specify the appearance of the update buttons. See “Update tab” on page 66.

Breadcrumbs Change the appearance of the indicator used to guide the user through the user interface. See “Breadcrumbs tab” on page 68.

Business Logic Add your custom JavaScript functions that are called when each grid data cell is drawn. See “Business Logic tab” on page 69.

Preview View a graphical representation of your mobile application template settings. See “Preview tab” on page 71.

Device tab

The Device tab enables you to change the template name, the width and height of the mobile application, and the assigned roles.

Table 2-4: Device tab parameters

Parameter	Description
Name	The template name. If the application is new, the default template name is PDA Grid Basic. This template contains all the default properties. To create a new template, enter a new name. All changes you make are associated with the new template name when you save the template. To use another template, enter the template name.
Resolution Width	The width of the mobile application in pixels. The default is 225 pixels.
Resolution Height	The height of the mobile application in pixels. The default is 150 pixels.
Roles	Select one or more roles from the Available Roles list, and click Add to add them to the Assigned Roles list, which defines what roles can access this mobile application application.

Appearance tab

The Appearance tab enables you to change content color, font type, and font size.

Table 2-5: Appearance tab parameter

Parameter	Description
Default properties	
Font	The font used in the mobile application. The default is Arial.
Font Size	The font size, in pixels, used in the mobile application. The default is 3 pixels.
Font Color	The font color used in the mobile application. The default is #6B875D.
Color	The color used in the mobile application. The default is #000000.
Header properties	
Background Color	The background color used in the grid's header. The default is #6B875D.
Text Color	The text color used in the grid's header. The default is #FFFFFF.
Row properties	
Row Text Color	The text color used in the grid's rows. The default is #000000.
Alternate Row Color	Indicates whether you want alternate row colors on the PDA. Alternating row colors may make the grid text easier to read. If this box is selected, alternate row colors are used. If the box is not selected, the default row color is white.
Alternate row color properties	
Even Row Background Color	The background color used for even-numbered grid rows (applicable only if you selected the Alternate Row Color). The default is #AEC6A2.
Odd Row Background Color	The background color used for odd-numbered grid rows (applicable only if you selected the Alternate Row Color). The default is #FFFFFF
Row color properties	
Row Background Color	The background color used for grid rows (applicable only if you did not select the Alternate row color box). The default is #FFFFFF.
Margin properties	
Left Margin Size	The width of the left margin of each cell, in pixels. The default is 5 pixels.
Top Margin Size	The size of the top margin of each cell, in pixels. The default is 5 pixels.

Note You can change the grid color property by entering a color Hexadecimal code (such as #6B875D) in the text box, or you can click the color palette next to the color properties, and select a color.

Column tab

The Column tab sets the maximum length of each column cell data string. Certain column data cells have long data strings, which might make the data difficult to read in the grid format. The column customization feature enables you to control the maximum characters to display for each column cell.

The depth column properties refer to grid data. For example, Depth 1 is the initial grid data the user sees. If a user clicks a link in the first grid data that goes to another list of grid data, the second list of grid data is Depth 2 of the grid data, and so on, to Depth 5.

You can set the length of each column in each depth by selecting a column from the drop-down list, and entering the maximum number of characters that can display in that column.

For example, on Depth 1, you may want to set column 1 to display 5 characters. Select “column 1” from the drop-down list, and enter “5” in the text box. Move the mouse so the focus is changed to a different field to save the entry for that column. Repeat the procedure to set column 2, and so on.

Table 2-6: Column tab parameters

Parameter	Description
Enable Column Truncating	Indicates whether to enable the column truncating feature. If enabled, long strings of data are truncated according to the number of characters you specify to display. If disabled, the entire data string is displayed. By default, column truncating is disabled.
Depth 1 – 5 column properties	
Column #	Select each column from the drop-down list for depth 1 – 5, and assign the column width.
Chars	For each column, enter the column width in characters, for example, enter 4 to indicate four characters wide.

Paging tab

The Paging tab specifies whether to enable paging, the number of grid rows to display at one time, and the appearance of the paging buttons if paging is enabled. Paging enables you to organize data in pages that you can access easily.

Table 2-7: Paging tab parameters

Parameter	Description
Enable Paging	Indicates whether to enable the paging feature. If paging is enabled, data is organized into pages that can be accessed by selecting page buttons. If paging is disabled, all the data is displayed. By default, paging is enabled.
Paging properties	
Number of Rows in a Page	The maximum number of rows in the grid display allowed on each page. The default is 7.
Paging button font properties	
Font	The font type used in the grid. The default is Arial.
Font Size	The font size used in the grid. The default is 10 pixels.
Paging button properties	
Active Button Color	The color of the active, or selected, paging button. The default is #6B875D.
Active Button Text Color	The text color of the active paging button. The default is #FFFFFF.
Inactive Button Color	The color of any inactive, or unselected, paging buttons. The default is #EFEFEF.
Inactive Button Text Color	The text color of any inactive paging buttons. The default is #808080.

Note You can change the grid color property by entering a color Hexadecimal code (such as #6B875D) in the text box, or you can click the color palette next to the color properties, and select a color.

Search tab

The Search tab indicates whether to enable search, whether to show the search mechanism, and how to customize the search buttons. The search mechanism includes the search field, the Find, Close, and Back buttons, the column drop-down list, the criteria drop-down list, and the All-Level check box. The search mechanism can be very useful, but it does use several rows of a PDA's limited display. Search is enabled by default.

Table 2-8: Search tab parameters

Parameter	Description
Enable Search	Indicates whether to enable the search feature. If enabled, users can use the search mechanism to search for data. If disabled, users cannot search. By default, search is enabled.
Search properties	
Show Search Option	Indicates whether the search mechanism is displayed. If enabled, the column drop-down list, the search criteria drop-down list, and the All Depth Level check box display. By default, show search is enabled. If disabled, the search mechanism does not display, and you must set defaults for search column number, search criteria, and search all level. For mobile applications, the Search button appears to the right of the paging buttons. When you click the Search button, the column drop-down list, the search criteria drop-down list, and the All Depth Level check box display.
Default Search Column #	If the Show Search Option is disabled, select a default search column number from the drop-down list. The default is 1.
Default Search Criteria	If the Show Search Option is disabled, select a default search criteria from the drop-down list. The default is “like.”
Default Search All Level	If Show Search Option is disabled, indicate whether to search all levels. The default is to not search all levels.
All Level Text	You can change the text “all” next to the check box by entering different text in the text box to the right of “All text level:”
Use Images	Indicate whether to use images or text for search buttons. By default, images are used.
Image button paths	
Image Location	If images are used for buttons, specify the image file location. By default, button images are stored in: `SYBASE/UA65/tomcat/webapps /onepage/portlets/templates/ mobile/images` To customize images, provide the correct image file location.

Parameter	Description
Search Button Image	Identifies the Search button image; for example, *btn_search_6B875D.gif*. A preview of the button displays to the right of the input field.
Find Button Image	Identifies the Find button image; for example, *btn_find_6B875D.gif*. A preview of the button displays to the right of the input field.
Close Button Image	Identifies the Close button image; for example, *btn_close_6B875D.gif*. A preview of the button displays to the right of the input field.
Back Button Image	Identifies the Find button image; for example, *btn_back_6B875D.gif*. A preview of the button displays to the right of the input field.
Text button properties	
Button Color	Enter or select the button color to use for search buttons, using information in the Note, below. The default is #6B875D.
Search Button Text	Use the default Search button text or type new text.
Find Button Text	Use the default Find button text or type new text.
Close Button Text	Use the default Close button text or type new text.
Back Button Text	Use the default Back button text or type new text.
Text button font properties	
Font	The font type used for the search buttons. The default is Arial.
Font Color	The font color used for the search buttons. The default is #FFFFFF.
Font Size	The font size used for the search buttons. The default is 10 pixels.

Note You can change the grid color property by entering a color Hexadecimal code (such as #6B875D) in the text box, or you can click the color palette next to the color properties, and select a color.

Update tab

The Update tab indicates whether users can update cell data from the PDA, and enables you to customize the update buttons.

Table 2-9: Update tab parameters

Parameter	Description
Enable Update	Indicates whether to enable the update feature. If enabled, users can update cell data. If disabled, users cannot update cell data. By default, update is enabled.
Use Images	Indicates whether to use images or text for update buttons. By default, images are used.
Image path properties	
Image Location	If images are used for buttons, specify the image file location. By default, button images are stored in: `SYBASE/infoediton/tomcat/webapps /onepage/portlets/templates/ mobile/images` To customize images, provide the correct image file location.
Edit Button Image	Identifies the Edit button image; for example, *btn_edit_6B875D.gif*. A preview of the button displays to the right of the input field.
New Button Image	Identifies the New button image; for example, *btn_new_6B875D.gif*. A preview of the button displays to the right of the input field.
Save Button Image	Identifies the Save button image; for example, *btn_save_6B875D.gif*. A preview of the button displays to the right of the input field.
Done Button Image	Identifies the Done button image; for example, *btn_done_6B875D.gif*. A preview of the button displays to the right of the input field.
Text button font properties	
Font	The font type used for the update buttons. The default is Arial.
Font Color	The font color used for the update buttons. The default is #FFFFFF.
Font Size	The font size used for the update buttons. The default is 10 pixels.
Text Button properties	
Button Color	Enter or select the button color to use for update buttons, using information in the Note, below.
Edit Button Text	Use the default Edit button text or type new text.
New Button Text	Use the default New button text or type new text.
Save Button Text	Use the default Save button text or type new text.
Done Button Text	Use the default Done button text or type new text.

Parameter	Description
Enable Update JavaScript	Indicates whether to enable the JavaScript function. This enables you to add a JavaScript update validation call. If enabled, you can access the text area under Update JavaScript Properties.
Update JavaScript Properties	Enter an update function call using JavaScript. The function is called when an update is submitted from the PDA. See "Sample update function call using JavaScript," below, for example code.

Note You can change the grid color property by entering a color Hexadecimal code (such as #6B875D) in the text box, or you can click on the color palette next to the color properties, and select a color.

Sample update function call using JavaScript

This is a sample function that is called when a user submits an update from a PDA:

```
// This function will be called when save is click
// arr : An array of updated input fields
function updateValidation(arr)
{
    // arr is an array of the modified updated values
    for (var ii=0; ii < arr.length; ii++)
    {
        // The input name is receivd from arr[ii][0]
        // The input value is received from arr[ii][1\
    }
}
```

In this example, comments are added to assist users on the data returned. An array is passed to the "updateValidation" JavaScript function which contains a list of all the input data within the update form, including the input name and its previous or updated values. You can add validations or change values within the array. You need not return the array since it is referenced within the main code.

Breadcrumbs tab

The Breadcrumbs tab enables you to change the appearance of the breadcrumb, or indicator, used to guide the user through the user interface.

Table 2-10: Breadcrumbs tab parameters

Parameter	Description
Enable Display of Breadcrumbs	Indicates whether to enable use of breadcrumbs. If selected, breadcrumbs are enabled, otherwise, breadcrumbs are disabled. By default, breadcrumbs are enabled.
Root Page Text	The text used to identify the initial page of data displayed. The default is Main.
Use Images	Indicates whether to use an image or text as the breadcrumb indicator. If selected, an image is used, if not selected, a text indicator is used. By default, images are used.
Image path properties	
Image Location	If a breadcrumb image is used, specify the image file location. By default, the breadcrumb image is stored in: `SYBASE/infoediton/tomcat/webapps /onepage/portlets/templates/ mobile/images` To customize images, provide the correct image file location.
Arrow Image Path	If an image is used, identify the Arrow image, for example, *btn_trail_image_6B875D.gif*. A preview of the image displays to the right of the input field.
Arrow text properties	
Arrow Text	If an image is not used, enter the text to use in place of the Arrow image, for example: `->`
Enable Breadcrumb Text Truncating	Indicates whether breadcrumb text should be truncated to save space. If selected, text is truncated, if not selected, text is not truncated. The default is not truncated.
Breadcrumb Text Truncation Length	If breadcrumb truncation is enabled, specify the truncation length. The default is 5 characters.

Business Logic tab

The Business Logic tab enables you to add your own custom JavaScript functions that are called when each grid data cell is drawn.

Table 2-11: Business Logic tab parameters

Parameter	Description
Enable Business Logic	Indicates whether to enable the business logic feature. If enabled, you can add business logic in the form of JavaScript. If disabled, you cannot add business logic. By default, business logic is disabled.
Business Logic JavaScript Function Declarations	Enter a function call using JavaScript. The function is called when an update is submitted from the mobile device. See below for example code.

Sample business logic call using JavaScript

Following is a sample function that is called when cells are displayed on a mobile device.

```
// This function will be called whenever each cell data
// is drawn
// cellValue : value of the data cell
// function evaluateFunction(cellValue)
{
  // cellValue is the value of the cell to be printed
  // The following show an example of changing the data
  // cell font color
  // to red and adding a link to call alertFunction when
  // click on
  // uncomment out to use
  /*
  createElement("A");
  eleHref.href = "#";
  eleHref.onclick = alertFunction;
  eleHref.style.color = "#FF0000";
  eleHref.appendChild( document.createTextNode(
  cellValue ) );
  return eleHref;
  */
  // return string or DHTML object to be replaced
  // return cellValue;
}

// uncomment this out to use
/*

function alertFunction()
{
  // the reference to this is the object declared to
  // this function
```

```
  var obj = this;
  // This will return the text that is clicked on
  var txt = obj.childNodes[0].nodeValue;
  alert("I have clicked on : " + txt);
  // If you want to show status or alert at the
  // bottom of the grid table use the following
  // showCustomData("My text here : " + txt);
}
*/
```

In this example, when the user clicks a cell, the cell content is displayed in red (#FF0000) and the status or alert displays at the bottom of the grid table.

Preview tab

The Preview tab shows what customization property settings look like on a mobile device. You cannot edit this window.

Note This preview is not an exact representation of what you will see on your PDA/WAP devices. The presentation varies in size and color.

Alerts

Alerts allow you to create user notifications based on an application's conditions. For example, you can send an e-mail when the price of a specific stock reaches a specified value.

This section describes how to:

- Create alerts
- Establish alert rules and schedules
- View alert details
- View the generated alert in an e-mail application or on a mobile device

When you create an alert, the corresponding agent can be seen in Mobile Web Studio. Alerts have these attributes:

- Rules – specify the application content that triggers an alert.

- Schedule – specifies when the agent should check the application for alert-producing content.

Alerts are saved based on user name. One alert (agent) definition is created per user, and is the unique ID for this user's agent.

Requirements and restrictions

The following list describes requirements and known limitations of the alert functionality in Unwired Accelerator version 6.5:

- Alerts require that you use Portal Interface with Internet Explorer 5.5 or 6.0. Netscape support is not currently available.
- You can define alerts only on applications created in Mobile Web Studio; that is, you cannot define alerts on personal applications that are created in Portal Interface.
- To define alerts on an application, you must select the Alert option on the Presentation tab when you save the application in Mobile Web Studio.
- Once the Alert option is selected and users set rules on that application in Portal Interface, removing (unselecting) the Alert option on the application in Mobile Web Studio has no effect. That is, you cannot roll back alerts that have been defined on an application by existing users.
- The Portal Interface alert capability uses the agent functionality of Mobile Web Studio. After an alert is defined on an application, an agent is created on the application by a user or by defining a schedule in the Alerts tab of MyInfo functionality. An agent is created in the background with the user's name. If an agent exists in Mobile Web Studio with the same name as a user, the existing Mobile Web Studio agent is overwritten.
- When Portal Interface users who have defined alert rules or schedules are deleted or disabled in Mobile Web Studio the administrator should delete or disable the operation to remove the alert agent associated with the user if one exists.

❖ **Creating alerts**

This example procedure shows you how to create an alert in Mobile Web Studio.

1 Use the sample database application included with the product.

   a Log in to Mobile Web Studio.

b Select Applications from the Build menu on the left pane.

c Click New to access the Application Builder.

d When the Application Builder displays, click the down arrow to the right of the Add button and select Database Element.

e For this example, when the Mobile Web Studio – New Element window appears, select Connection Cache.

f Complete the remaining fields:

- Conn Cache Name – name of the connection cache for the sample database; for this example:

  `sampledb`

- SQL Query String – enter this SQL statement:

```
select c.lname, p.name, si.quantity, si.id, si.prod_id
from customer c, sales_order s, sales_order_items si, product p
where c.id = s.cust_id and s.id = si.id
and si.prod_id = p.id
```

g Click Preview to see the data that corresponds to the SQL query you entered.

2 Click Next.

3 On the next three windows (Split, Define, and Filter), click Next.

4 In the Window Preview, enter `Sales Update` for the Element Name, then click Next.

5 In the Continuous Capture window, click Finish. Do not click Continue in this window.

6 In the Application Builder, click Save and complete these options on the Web Studio – New Application Finish window:

a Content tab – enter `Sales Update` for the application name.

b Roles tab – click Add All to add all roles to the Assigned Roles list.

c Presentation tab – select Alert, Display Within IFrame, and No Popup. Accept the default entries shown for the other options.

d Click Finish.

e In the confirmation pop-up window, click OK.

7 Click Close to exit the Application Builder.

8 Select New from Application Manager Status menu.

9 Right-click the new Sales Update application in the right pane and select Approval Status | Approved.

In the confirmation pop-up window, click OK.

10 Add the new application to a page.

a Select Pages from the Web Studio Build menu.

b Select New from the Page Builder Status menu, then click New.

c When the Page Builder displays, select the full layout icon (the far right layout icon), then click Add.

d When the Search window appears, click Search.

e Select the Sales Update application in the Results pane and click Add.

f In the Page Builder, click Save and complete these options:

- Name – enter `Sales`.
- Type – select Catalog from the drop-down list.
- Active – verify that this option is selected.
- Roles – click Add All to add all roles to the Assigned Roles list.

Click OK.

11 When you see the confirmation that the page was saved, click OK.

12 Click Close to exit the Page Builder.

13 On the Mobile Web Studio main window, select New from the Page Builder Status menu, right-click the Sales page, and select Status | Approved.

When you see the message that the page was saved, click OK.

14 Make the new page and application available to Portal Interface users.

a Select Approved from the Page Builder Status menu.

b Right-click the Sales page in the detail pane and select Update Users.

c When you are prompted to reconfirm that you want to perform an update operation, click OK.

d When a message displays that the update operation succeeded, click OK.

15 Log out of Mobile Web Studio.

16 Alerts do not start until you set up rules for the alert in Portal Interface. To set up rules for the application you created in the previous steps:

a Log in to Portal Interface.

b Click Manage Pages.

c In the Add Page Group window, enter `Information` as the page group name, and click Done.

d Click Add Page.

e In the Add Page window, add the pre-made catalog page "Sales," and click Done.

f In the Sales page, click the "!" icon in the Sales Update application.

g In the Alert window:

- Name – select Playback from the drop-down list.
- Rule Data – enter the text you want to receive as a message in the alert. For this example: `Sales data update`.
- Description – enter a brief description of the alert. For example: `Sales data has been updated`.
- Under Rules:

  Rule Relationship – select "and".
  Select Fields By – select "label".
  Execute Alert When – select "field3", "is greater than", 25 in the three columns, then click +.
  Enter another rule using the same values for Rule Relationship and Select Fields By.
  Execute Alert When – select "field1", "equals(text)", enter Devlin, then click +.

h Click Done.

i In the confirmation pop-up, click OK.

j Click Cancel to close the Alert window.

17 Set up a schedule for the alert:

a Click MyInfo, and click the Alert tab.

b In the Alert window, select Trigger Alert Based on a Schedule.

c Select the Start Time, Recurrence, and Range.

d Select No End Date.

e Send Alerts To – enter the e-mail address where you want to receive the alert.

f Click Done.

g In the pop-up confirmation, click OK.

18 Log out of Portal Interface.

19 Log in to Mobile Web Studio to create a database update application, which can change the quantity of the Tee Shirts for Devlin in the sampledb and trigger an alert.

a Select Applications from the Build menu on the left pane.

b Click New to access the Application Builder.

c When the Application Builder displays, click the down-arrow to the right of the Add button and select Database Element.

d For this example, when the Database Element Definition window appears, select Connection Cache.

e Complete the remaining fields:

- Conn Cache Name – name of the connection cache for the sample database; for this example:

  `sampledb`

- SQL Query String – enter this SQL statement:

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

20 In the Split, Define, Filter windows, and Database Element windows click Next.

21 In the Location: Database Element window, check all three variables (prod_id, id, and quanity).

22 In the Preview Window, in Element Name, enter:

- Element Name – `Devlin`
- id: – `2001`
- prod_id – `300`
- quantity – `25`

23 In the Continuous Capture window, click Finish.

24 In the Application Builder, click Save and complete these options in the Finish window:

a On the Content tab, enter `Devlin` for the application name.

b On the Roles tab, click Add All to add all roles to the Assigned Roles list.

c On the Presentation tab, select Alert, Display Within IFrame, and No Popup. Accept the default entries shown for the other options.

d Click Finish.

e When you see the confirmation message that the application was saved successfully, click OK.

Creating an SMS alert using Mobile Web Studio

This section shows how to create an alert for a Web element that is created from Mobile Web Studio instead of Portal Interface. The alert results are sent to a telephone or mobile device, so the data that is sent must be shortened as much as possible. This example illustrates how to send an alert to a Short Message Service (SMS) subscriber.

SMS is a service that lets cellular phone users send and receive short messages, generally 140 to 160 characters, depending on the cellular phone.

Unlike paging, but similar to e-mail messaging, short messages are stored at SMS centers, which lets you retrieve messages later if you are not available to receive them immediately. SMS messages travel to a cellular phone over the system's control channel, which is separate from the voice channel.

Note SMS was introduced in the Global System for Mobile Communications (GSM), a worldwide standard for digital wireless mobile phones. SMS is now supported by all other digital-based mobile communications systems.

The procedures are:

- "Creating an alert for SMS delivery" on page 78
- "Viewing the alert (telephone or mobile device)" on page 79

❖ Creating an alert for SMS delivery

This procedure describes how to create a stock quote alert for SMS delivery. Since 128 characters is the maximum size for short messages, you must shorten the data to be delivered from eight fields to the one pertinent field.

1. Log in to Mobile Web Studio.
2. Select Applications from the Build menu in the left pane, select New from the Application Manager Status menu, and click New to launch Application Builder.
3. Click the arrow to the right of the Add button and select Web Element. You see the New Web Element window.
4. In Location, enter a URL to capture grid content; for example: `http://finance.yahoo.com`, and click Find.
5. Scroll down the page, and click the "Most Actives" link under Top Stories.
6. In Format, select Grid from the drop-down list, and click Next.
7. Click Select to select the grid that includes "Last Trade" and click Next.
8. In the Split and Define windows, click Next.
9. In the Filter section, set up a rule to exclude most records. Since the stock quote is sent to a telephone or mobile device, which is limited to 128 characters, you want to send only the pertinent data. In Filter:
   a. Create a rule that includes only the Symbol, Name, Last Trade, and Change fields.

      Click Add.

   b. Create a rule to include the record with "MSFT" (Microsoft).

      Click Add.

   c. Click Next to bypass the Parameter window.
10. In Element Name, type `MS stock` and click Next. The Continuous Capture window displays.
11. Click Finish. The Application Builder window displays with the new application.
12. Click Save, and configure the application:
    a. In the Content tab, enter `Quick Stocks` as the name of the application.
    b. In the Roles tab, click Add All to add all roles to the application.

c In the Presentation tab, click Alert, Display Within IFrame, and No Popup.

Note You can also use the Application properties window to add an alert, after you have saved the application.

d In Window Preview, you can view the format.

e Click Finish.

f In the confirmation pop-up, click OK.

13 Click Close to close Application Builder. The new Web element displays in the right pane.

14 On the Mobile Web Studio main window, select New from the Application Builder Status menu, right-click the Quick Stocks application, and select Approval Status | Approved.

When you see the message that the application was saved, click OK.

15 Make the new application available to Portal Interface users:

a Select Approved from the Application Builder Status menu.

b Right-click the Quick Stocks application in the detail pane and select Update Users.

c When you are prompted to reconfirm that you want to perform an update operation, click OK.

d When a message displays that the update operation succeeded, click OK.

Viewing the alert (telephone or mobile device)

When the data matches the rule you established, in this case the stock price crossing a certain threshold, the alert is sent to your telephone or mobile device. For example, you may receive a message similar to this:

```
Last Trade: 17.11
```

❖ **Creating rules and schedules from Mobile Web Studio**

Use this procedure to create a sample rule and schedule for an alert using the Agent Manager.

1 In Web Studio, select Agents from the Manager menu in the left pane, then click New to launch Agent Builder.

2 Select the Schedule tab and establish a schedule for the rule:

   a Click the box next to “Trigger Agent based on a Schedule” to select it.

   b For Time, use the option controls to specify the start time. For this example, use the current time

   c For Recurrence, use the option controls to select the recurrence unit (such as seconds, minutes, hours) and interval. For the tutorial, select Minutes as the unit, and 5 as the interval. The alert schedule recurs every five minutes.

   d For Range, enter today’s date in MM/DD/YY format as the start date. For convenience, use the calendar icon to select the month and date.

      For the end date, click “No end date.” Alternatively, you can click “End by” and specify an end date.

3 Select the Rule tab and establish a rule for the newly created application:

   a Click the box next to “Execute Agent based on a “Rule” to select it.

   b Click the Add icon to find the newly created Application.

   c Click the Find icon once the Rule window launched.

   d Click Search to find the SMS Test Application that you created in “Creating an alert for SMS delivery” on page 78.

   e Select the application from the list to add this newly created Application, and click Add.

   f In “Execute agent when,” select “field2” and “is greater than” from the drop-down lists, and enter a value in the text field that is slightly higher than the current stock quote.

   g Click the plus sign (+) to apply this rule.

   h Click OK to finish.

4 Select the Action tab and set up a destination:

   a Click Add to add the destination, and select Notify | SMS. The Action Details window displays.

   b In SMS To, enter your telephone number.

   c Select your carrier from the drop-down list.

   d In Subject, enter a subject for this message.

   e Select “Include Rule’s Content.” Additional options display.

f Click “Send the Portlet that met the Rule Criteria.”

g Click OK to finish and close the Action Details window.

5 In Agent Builder, click Save to save the agent. Supply a name for the agent, such as `GoingUp`, assign roles to all, and click OK.

6 In Agent Manager | Stopped, select the newly created agent and click Start to run the agent.

Creating an e-mail alert using Mobile Web Studio and Portal Interface

For this example, create a Web element in Application Builder, and then create an alert for the Web element. You can enable alerts through the Application Builder Save window, or on the Application properties window in the Web Studio. Selecting the Alert option enables alerts for the application. Tasks include:

- “Creating an alert for e-mail delivery” on page 81
- “Creating a rule from Portal Interface” on page 82
- “Creating an alert schedule” on page 84
- “Viewing an alert created by a Portal Interface user” on page 85
- “Creating rules and schedules from Mobile Web Studio” on page 79

❖ Creating an alert for e-mail delivery

This section shows how to create a stock quote alert for e-mail delivery.

1 Log in to Web Studio.

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

3 Click the arrow to the right of Add, and select Web Element. You see the New Web Element window.

4 Enter a URL to capture grid content, for example, `http://finance.yahoo.com` and click Find.

5 Scroll down the page and click the “Most Actives” link, located in the middle column under Top Stories.

6 In Format, select Grid from the drop-down list, to display the stock data in grid format.

7 Click Next to display the presentation options.

8 Locate the presentation option that includes the Symbol, Name, and Last Trade columns, click Select, and click Next.

9 Click Next four times to bypass the Split, Define, Filter, and Parameter windows.

10 In Element Name, enter “web” and click Next. The Continuous Capture window displays.

11 Click Finish.The Application Builder window displays with the new application name.

12 Click Save, and configure the application:

a In the Content tab, enter “Stocks” as the name of the application.

b In the Roles tab, click Add All to add all roles to the application.

c In the Presentation tab, click Alert, Display Within IFrame, and No Popup.

d In Window Preview, you can view the format.

e Click Finish. The message “Application saved successfully displays.”

f Click OK.

13 Click Close to close Application Builder. The new Web element displays in the right pane.

14 Deploy the application to the Portal Interface.

❖ Creating a rule from Portal Interface

This section shows how to create a rule for the alert. This rule triggers an alert whenever the Last Trade value for Microsoft’s stock exceeds a set threshold. Use Portal Interface to add a rule to the Web element you created in “Creating an e-mail alert using Mobile Web Studio and Portal Interface” on page 81.

1 Log in to the Portal Interface.

2 Create a page group, a page, and add the “Stocks” application.

3 Click the Alert icon (!) on the application's title bar. The Alert definition window displays.

Note There are no default rules defined for an alert.

4 In the Alert definition window, define two rules for the Web element:

a In Name, select Playback from the drop-down list.

b Leave Rule Data and Description blank.

c Under Rules, define the rule relationship by clicking "and" to indicate multiple conditions.

You can also click "or" to indicate a choice between conditions.

d Define the condition for Select fields by clicking "label" to indicate field labels are used to meet the condition.

Other condition options include "position" to indicate window position, or "xpath" to indicate the same condition from every record.

e For the first rule, use "Execute alert when" to define the condition that triggers the alert:

Select "field 1" from the drop-down list.

Select "contains" from the drop-down list.

Enter `MSFT` (for Microsoft) in the value field, to establish a match for field 1.

Click the plus sign (+) to add the "Execute agent when" rule. The new rule displays under Rules and Rules Summary.

f For the second rule:

Select "field 3" from the drop-down list. Notice some of the other choices for establishing conditions that are available in the list.

Select "is greater than" from the drop-down list.

Enter `26` (to indicate a Last Trade stock price) in the value field, to establish a match for field 3.

Click the plus sign (+) to add the "Execute agent when" rule. The new rule displays under Rules and Rules Summary.

5 Click Done to save the Rule, and click OK in the confirmation window.

6 Click Cancel to close the Alert window.

Creating the schedule (Portal Interface)

This section shows how to create a schedule for the alert. You will use Portal Interface to add the schedule for the rules you set up in “Creating a rule from Portal Interface” on page 82. Alerts do not start until you set up and save a schedule, and start the alert.

Every alert needs a destination to deliver the content. The destination can be:

- An e-mail address. The user’s registered e-mail address is the default destination. You can specify another e-mail address.
- An SMS subscriber. Default SMS carriers include ATT Wireless, Cingular, Nextel, Sprint, T-Mobile, and Verizon.

  You can add additional SMS carriers to the *oem.xml*, using the syntax shown in this example:

```
<DeviceList>
   <DeviceClass name="sms" type="email">
       <DeviceDef name="ATT Wireless"
        address_value="mobile.att.net" />
       <DeviceDef name="Nextel"
        address_value="messaging.nextel.com" />
      <DeviceDef name="Sprint"
       address_value="messaging.sprintpcs.com" />
   </DeviceClass>
</DeviceList>
```

❖ **Creating an alert schedule**

1 From Portal Interface, click the My Info link. The icon looks like a lowercase “i.”

2 In MyInfo, select the Alert tab.

3 Set up the schedule parameters and the alert destination:

   a Click the box next to Trigger Alert Based on a Schedule to select it.

   b For Time, use the option controls to specify the start time. For the tutorial, use the current time.

   c For Recurrence, use the option controls to select the recurrence unit (such as seconds, minutes, hours) and interval. For the tutorial, select Minutes as the unit, and 5 as the interval. This indicates that the alert schedule recurs every five minutes.

   d Click Disable once to indicate that the e-mail or SMS notification should be sent only once, the first time the conditions are met.

e For Range, enter the current date in MM/DD/YYYY format as the start date. For convenience, use the calendar icon to select the month and date.

For the end date, click "No end date." Alternatively, you could click "End by" and specify an end date.

f For Send Alerts to, click Email and enter your e-mail address. Alerts are sent to this e-mail address. You can also click SMS to send the alert to an SMS subscriber, as described in "Creating an SMS alert using Mobile Web Studio" on page 77.

4 Click Done to save the schedule.

5 Click Start. The Start and Stop buttons act as a toggle, starting and stopping the schedule, and therefore, the alert.

6 Click Cancel to close the Alert window.

❖ **Viewing an alert created by a Portal Interface user**

This section shows how to view the rules and schedule for the alert you created, on the Portal Interface, from the Mobile Web Studio.

1 Log in to Mobile Web Studio.

Note Use your portal login name.

2 Select Agents in the left pane.

3 Select Ready under Agent Manager. You see the alert you modified in Portal Interface. The alert has the same name as the Portal Interface account name.

4 Double-click the alert to display the Agent Builder window.

5 Select the Schedule tab to see the schedule for the newly created alert.

6 Select the Rule tab to see the rule list for the newly created alert.

Note You can disable a rule by clicking the box to the left of the Rule ID.

7 From the Rule tab, select a rule and click Edit to see the details of the rule. You can modify the rule from Mobile Web Studio, assuming you have correct permissions, even if it was created on Portal Interface. Click OK to close the edit window.

8 Click Close to close Agent Builder.

❖ **Viewing the alert in e-mail**

This section shows how to view the generated alert in your e-mail application. Wait to perform this step until after the first interval that you set in “Creating an alert schedule” on page 84.

1 Log in to your e-mail account, using the account that you specified in “Creating an alert schedule” on page 84.

2 View your inbox for the alert notice. Use the Date and Time, or Subject columns to identify the alert.

3 Double-click the alert to see details.

Note You may want to stop the alert to keep your e-mail account from filling up. To do so, use the Stop button.

---

global.properties.xml additions and changes

The *global.properties.xml* file is the master configuration file for Mobile Web Studio and Portal Interface. Table 2-12 lists the additions or changes to the *global.properties.xml* file that are not documented in the *Enterprise Portal Developer's Guide*.

Global property group

This group contains general settings, including server names, addresses, mail properties, and portal properties.

Table 2-12: global.properties.xml changes and additions

Property name	Property description
XmlValidation	<Property name="XmlValidation" value="@XML_VALIDATION@" description="(on/off) Enable/Disable runtime xml validation for definitions of portlet, page, application, and template. This property should be turned on in development environment. Set to off to improve performance in production environment" menugroup="10"/>
MBusinessGroup	<PropertyGroup name="MBusinessGroup" descripton="properties for configuring M-Business Server in the Portal and Studio.">

Property name	Property description
MS.Enabled	<Property name="MS.Enabled" value="@MS.ENABLED@" description="(true/false) true to enable MobileStudio. false to it." menugroup="10"/>
MBusinessGroup	<PropertyGroup name="MBusinessGroup" descripton="properties for configuring M-Business Server in the Portal and Studio.">
MB.Enabled	<Property name="MB.Enabled" value="@MB.ENABLED@" description="(true/false) true to enable MBusiness integration Portal and Studio. false to disable it." menugroup="10"/>
MB.AutoRegistration	<Property name="MB.AutoRegistration" value="false" description="(true/false) true to enable auto MBusiness user registration when a user is created in EP. false to disable it." menugroup="10"/>
AuthenticationUsing	<Property name="AuthenticationUsing" value="database" description="Authentication technique to use: CSI/Database/EPSecurity" menugroup="100"/>

Index

A

C

D

E

F

G

I

J

M

N

P

R

S

T

U

X