

Fast Path to Web Mobility with IBM WebSphere Everyplace Mobile Portal

Table of Contents

Overview	2
IBM WebSphere Everyplace Mobile Portal	2
Mobile Portal and XDIME	3
Creating mobile applications with Mobile Portal	4
Getting started with the Studio wizards.....	4
About layouts	7
Defining layouts for your devices	7
Connecting the canvas layout to XDIME	9
Special features of XDIME.....	11
Other helpful features	16
Conclusion	18
Resources	18



Overview

We live in an increasingly mobile world. Employees are no longer office-bound. Customers are continually on the go. People need access to information and services *wherever* they are, using *whatever* devices they have. This can include someone using a laptop at an Internet café, someone using a PDA in an airport, and someone using a mobile handset while riding in a taxi. The possibilities are almost limitless.

The challenge for any service provider, wireless carrier, or telecommunication company that wants to succeed has always been to deliver the newest, most appealing network-based services, applications, and content, and do it all ahead of its competitors.

Now, in this age of mobility, the challenge has become more complicated. To succeed means also providing these services, applications, and content to a growing range of devices while maintaining a consistently good user experience. Even without workforce size and budget constraints, this is a formidable task that can quickly get out of hand. The matrix in Figure 1 shows how complicated it can be to meet this growing market of pervasive devices.

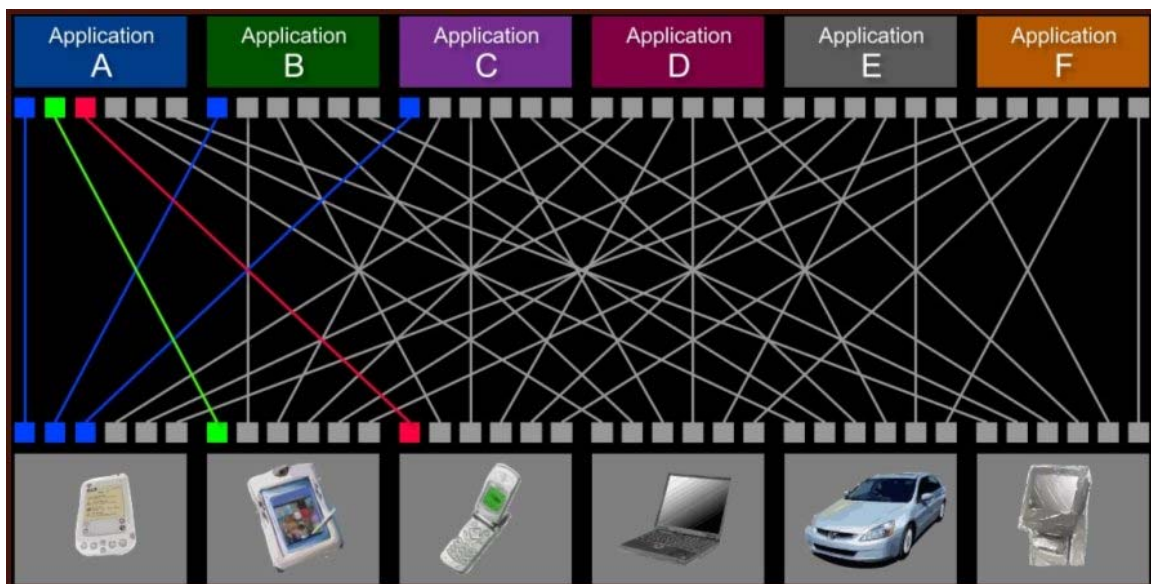


Figure 1. Matrix of Web applications and pervasive devices

IBM WebSphere Everyplace Mobile Portal

IBM® WebSphere® Everyplace® Mobile Portal (hereafter called Mobile Portal) can help. This first offering in the WebSphere Everyplace Service Delivery family extends the capabilities of IBM WebSphere Portal to an almost limitless range of mobile devices. You can provide your Web content, services, and applications to these devices while maintaining the benefits and advantages of a portal Web site, such as on-screen application integration, single sign-on, and separation of logic/layout/and branding...all administered though the same powerful and easy-to-use graphical user interface.

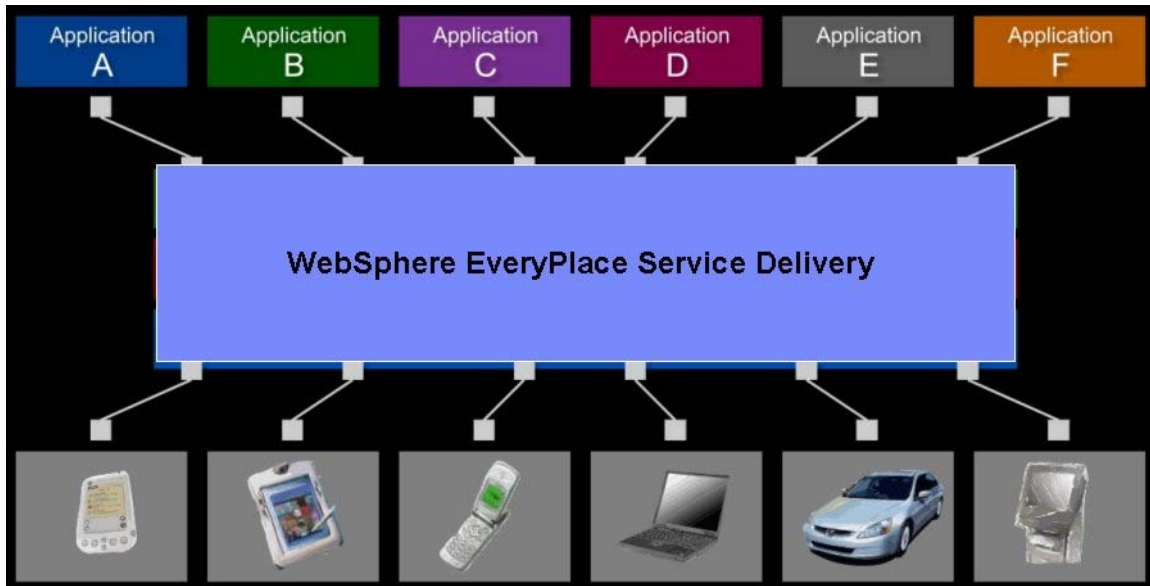


Figure 2. Managing the matrix with WebSphere Everyplace Service Delivery

Mobile Portal includes a multi-channel server (MCS) component that recognizes the capabilities of a device and sends it the correct content. It also includes state-of-the-art tools that enable you to create, adapt, manage, and scale cutting-edge Web applications that are not only widely accessible today, but will also stay accessible despite the proliferation of mobile devices. The applications stay accessible because Mobile Portal uses a flexible device-independent markup language and a repository of over 900 mobile device profiles that can be easily updated by a subscription service whenever new devices come to market.

By managing the delivery of your Web services and applications, Mobile Portal untangles the matrix you saw in Figure 1. It allows you to write your application once and deliver it to multiple devices.

Mobile Portal and XDIME

At the core of Mobile Portal is the idea that content and service authors should use device-independent markup to create mobile applications. XDIME is not intended as a markup language in its own right, but as a set of extensions upon industry-standard markup languages recommended by the W3C.

Although mobile handsets do support a small number of standards (such as XHTML Basic, XHTML Mobile Profile, I-Mode, WML, and HTML), there are detailed differences between the standards and the way they are implemented, and many do not support stylesheets. Because of this, they would require profoundly different markup to get consistent results. Also, because of other differences in device shapes and capabilities, the *same* markup might render quite differently on various devices, and do so in an unpredictable manner. XDIME eliminates these problems.

XDIME implements markup languages such as XHTML and XForms to provide true device-independent authoring facilities. It gives authors a way to build powerful and robust distributed applications, using their existing authoring skills. And, because the XDIME does not need to contain embedded device capability,

the application is protected against underlying changes in device knowledge and will not require revisions. Mobile Portal lets authors write application markup once and render it in the same many on many devices.

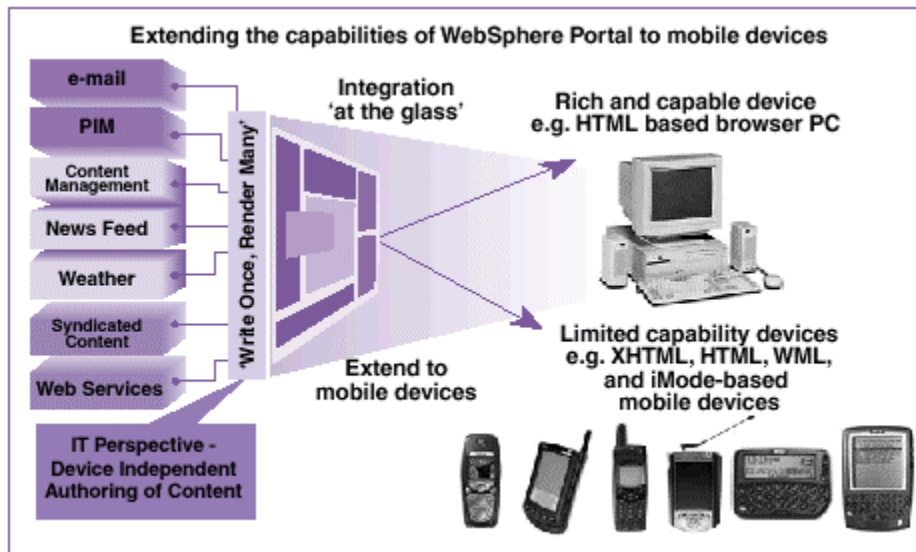


Figure 3. Extending the capabilities and benefits WebSphere Portal

Creating mobile applications with Mobile Portal

Mobile Portal comes with IBM award-winning tools: WebSphere Studio Application Developer and WebSphere Everyplace Toolkit with extensions for Mobile Portal development. The tools includes a combined WebSphere Application Server and WebSphere Portal runtime, making it easy to test and debug your applications.

Getting started with the Studio wizards

With the Mobile Portal tool extensions, you can use the WebSphere Studio wizards to create templates for all the files you will need. The New Project wizard enables you to create a basic portlet project and specify XDIME as one of your outputs.

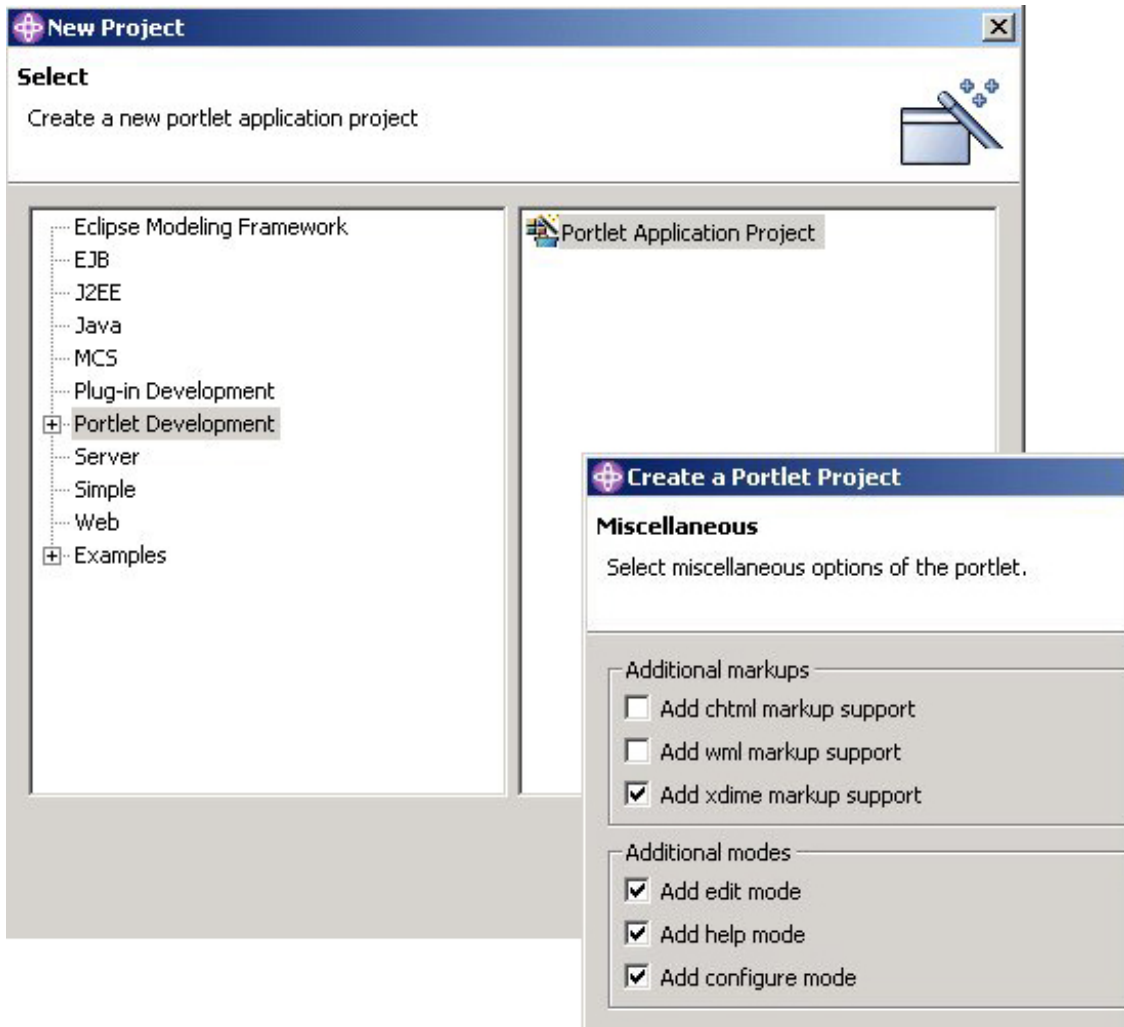


Figure 4. Creating a new portlet project for XDIME markup

The New Project wizard creates a Portlet Application Project for you with the correct directory structure for publishing to WebSphere Application Server, WebSphere Portal, and MCS. It includes basic JSP and Java files that you can edit and modify for your own needs.

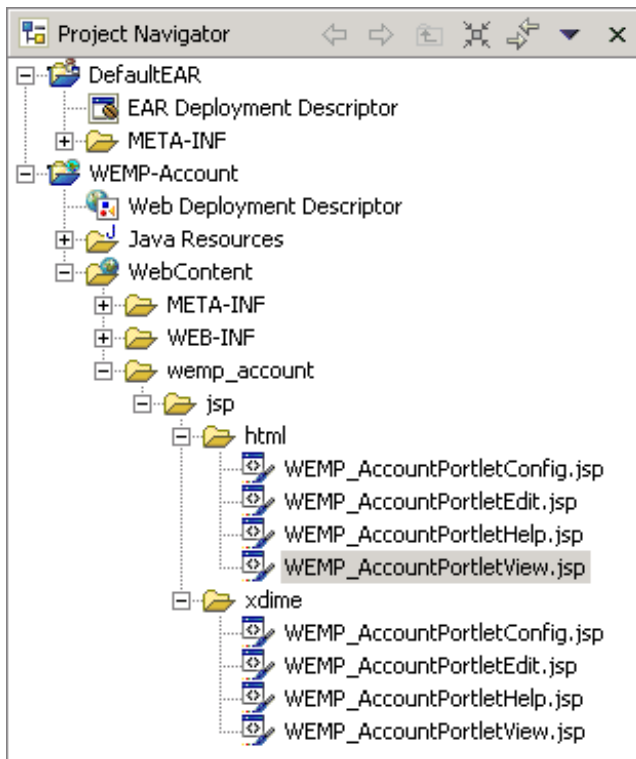


Figure 5. Project structure with JSP for XDIME markup

Figure 6 shows the HTML markup and JSP coding for the default page of a portlet's view mode. If you choose, the wizard can create similar pages for the other portlet modes to configure, edit and get help. As you develop your application you can add your specific content and application processing to these files, rename them, and also add other files to the project.

```

<%@ page session="false" contentType="text/html" import="java.util.*, wemp_account.*"%>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<portletAPI:init/>

<%
WEMP_AccountPortletViewBean viewBean =
(WEMP_AccountPortletViewBean)portletRequest.getAttribute(WEMP_AccountPortlet.VIEW_BEAN);
WEMP_AccountPortletSessionBean sessionBean =
(WEMP_AccountPortletSessionBean)portletRequest.getPortletSession().getAttribute(WEMP_Accou
untPortlet.SESSION_BEAN);
%>

<H3 style="margin-bottom: 3px">Welcome!</H3>
This is a sample <B>view mode</B> page. You have to edit this page to customize it for
your own use.<BR>
The source file for this page is "/Web
Content/wemp_account/jsp/html/WEMP_AccountPortletView.jsp"

```

Figure 6. Sample default HTML file for viewing a portlet

Default XDIME files, as shown in Figure 7, contain regular HTML tags and additional tags for the *canvas* and *pane* elements in the *layout*. In a similar way, you can add

your specific content and application processing to these files, rename them, and also add other files to the project.

```
<!-- keep JSP page compiler from generating code that accesses the session -->
<%@ page session="false" %>

<!-- load WPS portlet tag library and initialize objects -->
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<portletAPI:init />

<!-- Create a layout policy for your portlet. Specify the layout name as the
value for the layoutName attribute of the canvas element below. -->
<canvas layoutName="/REPLACE-WITH-PORTLET-LAYOUT-NAME.mlyt" type="portlet">

<!-- Specify the pane name from the portlet layout policy as the value for the
name attribute of the pane element below -->
<pane name="REPLACE-WITH-PANE-NAME-FROM-PORTLET-LAYOUT">
<p>Hello</p>
<p>
This is a sample xdime page
</p>
</pane>
</canvas>
```

Figure 7. Sample defaultXDIME file

About layouts

To display content in a web browser you need to specify a layout policy, and associate it with one or more devices. You give every layout a name that MCS uses to associate it with the related devices. When you choose these devices, you also define one of two categories of layout: a canvas or a montage.

You will find that you can use canvas layouts for most pages targeting multiple devices. You can divide a canvas layout into panes that will display your web content.

With smaller device displays, you also need to consider ways to partition the content. For example, a page authored for a PC screen will be much too large for a PDA. The solution is to divide the content into a set of page fragments that are small enough for display. Then you provide navigation links between the fragments.

Defining layouts for your devices

The layoutName in the <canvas> tag and the name in the <pane> tag both come from a MCS policy file that defines the layout template for a particular device or group of devices. In this completed WEMP-Account sample application shown in Figure 7, the account.mlyt file has 4 different device layouts that each refer to a category of devices in the Mobile Portal device repository. When a request comes to MCS, it determines which device is making the request, matches it to a canvas in the layout file, and sends back the content that you want served to that device.

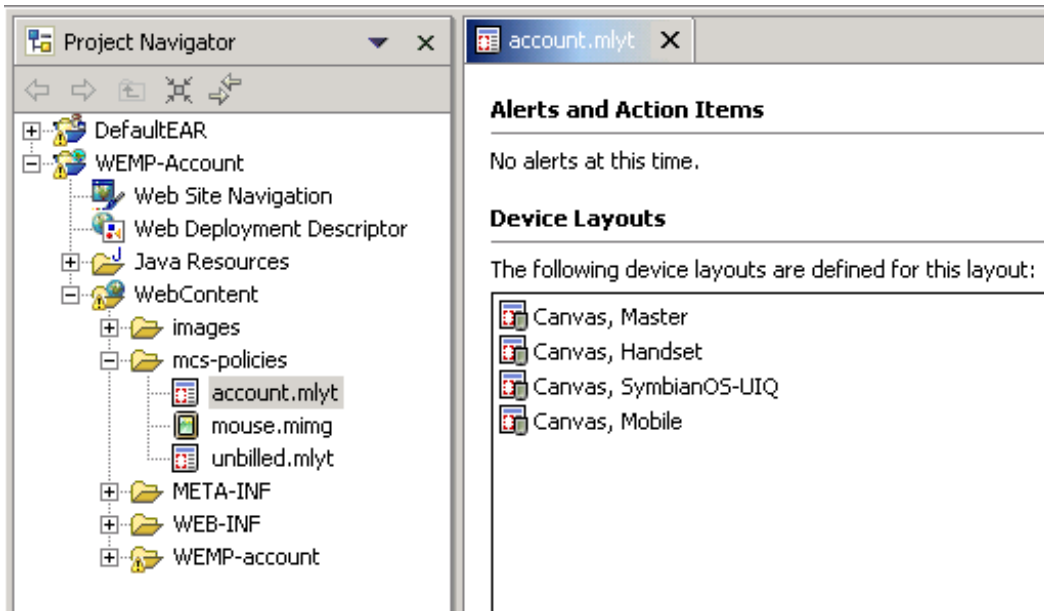


Figure 8. MCS layout file in the sample WEMP-Account application

You might have particular devices that you know your customers will be using but you will also want your Web content to look good and be appropriate and easy-to-use on any device. Mobile Portal makes it easy.

With the New Layout wizard, you can create a canvas layout at any level in the device repository. You can be as specific as a particular model of a particular device...or not. Because the device repository has all the known devices categorized by their display size and supported functions, you can use the general categories to handle a large number of similar devices.

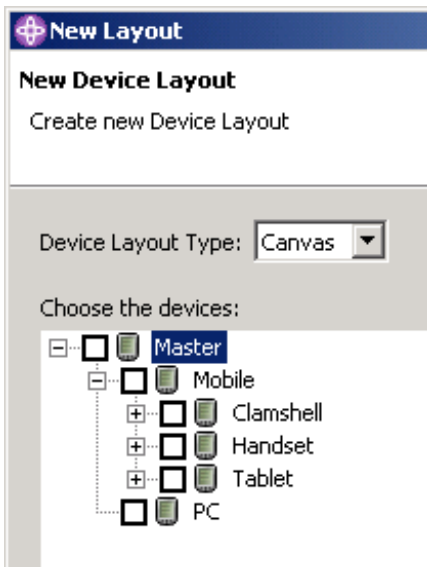


Figure 9. Hierarchy of devices in the New Device Layout wizard

Figure 10 shows how MCS uses the device repository to handle requests and determine which layout to return in our sample:

Any device that has a User-Agent of **SymbianOS-UIQ** gets the SymbianOS-UIQ canvas.

Others that fall under the **Smartphone** and more general **Handset** category (from HDML-Handset through XHTML-Handset) default to the Handset canvas.

All devices defined in the **Clamshell** and **Tablet** categories default to the Mobile canvas.

Every device that cannot be identified defaults to the Master canvas.

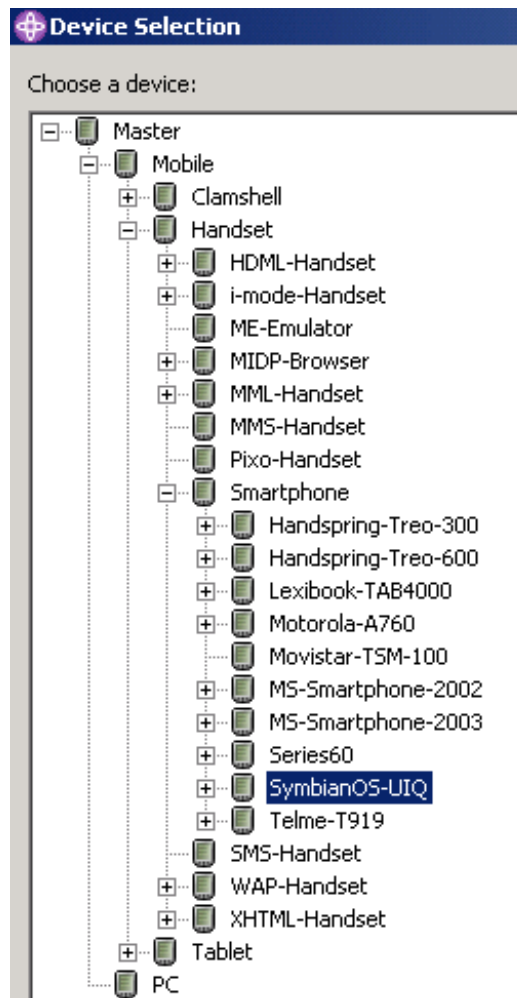


Figure 10. Hierarchy of devices in the Mobile Portal device repository

Connecting the canvas layout to XDIME

When you create a canvas for a device or category of devices, you decide what content to include and exactly how to position it. For instance, you can layout your text in a vertical rather than horizontal table, eliminate text that is not pertinent, and exclude images if they are inappropriate.

In our sample `account.mlyt` file shown in Figure 11, we have a pane named `combotitle` in our Mobile canvas and we have a pane named `title` in our Handset canvas. The name of the panes are links between the canvas layouts and the XDIME files.

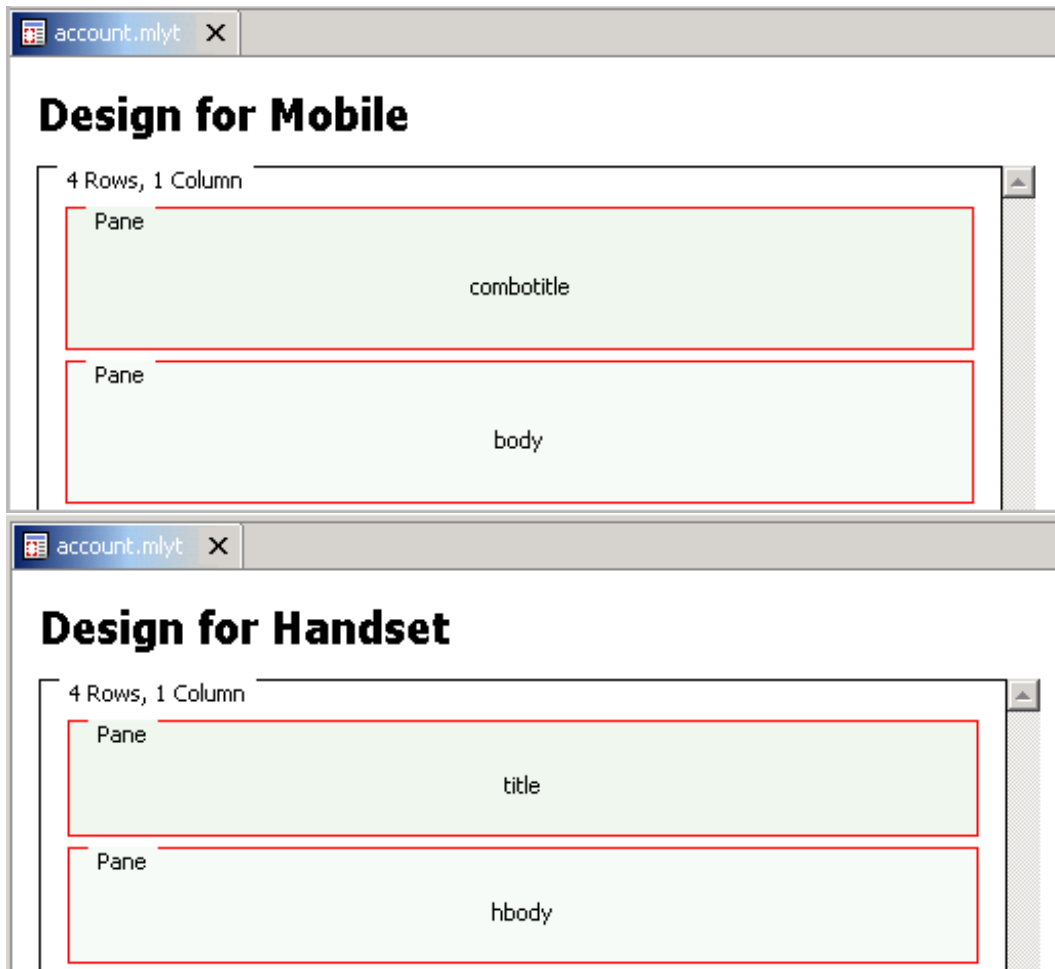


Figure 11. Sample canvas layouts

In this excerpt from our XDIME `welcome.jsp` file, we specified the `account.mlyt` file in our `<canvas>` tag. It contains the definition of all four of our canvases. We included coding with just text for the `title` pane, and text plus an image for the `combotitle` pane.

```

<canvas layoutName="/account.mlyt" type="portlet">
<pane name="title">
<b>Your account</b>
</pane>

<pane name="combotitle">
<b>Your account</b><br></br>
<br/>
</pane>

```

Figure 12. Sample XDIME markup

When a device in the Mobile category requests the WEMP-Account Welcome.jsp page, the MCS sends the layout with an image, but when a device in the Handset category requests that same page, it sends the layout without an image.



Figure 13. MCS delivers different Welcome.jsp pages to Mobile and Handset devices

Special features of XDIME

Sometimes you do not know how long the content on a served page will be, which is of even more concern when dealing with the small displays on mobile devices. For example, you might need to present the result set from a database query, or display the content from a dynamic data source such as a news feed or a Web search engine. XDIME markup has two features to assist with this: dissecting panes and iterators.

Dissecting panes

When you do not know the length of the content and want to avoid excessive scrolling, you can let MCS split your content for you into a series of pages with links to provide navigation between them. In this case, each individual page generated from a larger layout is called a fragment.

To do this, you create a root fragment for the whole canvas. This fragment will contain the links to the other fragments. The other fragments each contain a link back to the root fragment. Your users can select and traverse the links just as they would for any other Web link.

Iterators

Spatial iterators enable you to populate the rows and columns of tables, which can have both fixed and variable numbers of rows. Spatial iterators are especially useful for handling results from a database query or a Web search engine.

This excerpt of sample XDIME markup handles an undetermined number of news items, each with a headline, picture and text, from an outside source represented by the `nls.News` file.

```
<%
...
.....
NewsHeaderPane="NewsHeader."+i;
NewsImagePane="ImagePane."+i;
NewsDescriptPane="Description."+i;

NewsHeader="NewsHeadline."+i;
NewsImage="NewsImage."+i;
NewsDescript="NewsDescription."+i;
%>

<pane name="<%=NewsHeaderPane%>">
<a href="<portletAPI:createURI
  <portletAPI:URIAction name="viewOne"/>
  <portletAPI:URIParameter name="storyNum" value="<%=i_String%>" />
  </portletAPI:createURI>">
  <portletAPI:text key="<%=NewsHeader%>" bundle="nls.News" />
</a><br/><br/>
</pane>

<pane name="<%=NewsImagePane%>">
" bundle="nls.News"/>.mimg" />
</pane>

<pane name="<%=NewsDescriptPane%>">
<portletAPI:text key="<%=NewsDescript%>" bundle="nls.News" /><br/><br/>
</pane>
```

Figure 14. Sample XDIME markup for spatial iterators

With a spatial iterator in the canvas layouts, as shown in Figure 15, this block of ImagePane, NewsHeader, and Description panes can be repeated as many times as necessary.

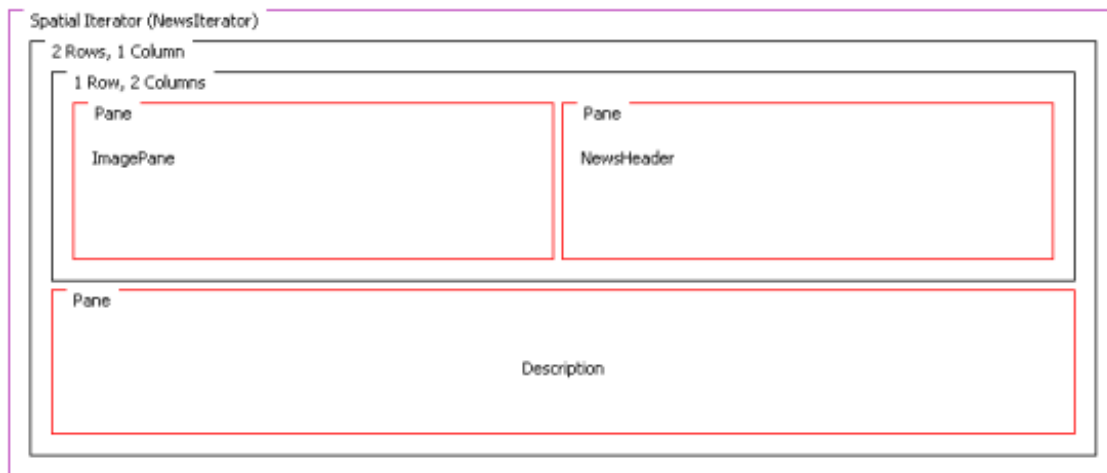


Figure 15. Sample Mobile canvas layout with spatial iterators

The layout for a handset device has a spatial iterator for the NewsHeader .



Figure 16. Sample Mobile canvas layout with spatial iterators

When the News page is rendered on a device in the Mobile device category, the headline, image and full story are repeated as many times as necessary. When it is rendered on a device in the Handset category, just the headline is repeated.



Figure 17. News page delivered to Mobile and Handset devices

In a similar way, a temporal iterator allows you to create a time-based, multimedia sequence with Synchronized Multimedia Integration Language (SMIL) markup, for use with the Message Preparation Sever (MPS).

Special features of Mobile Portal and MCS

There are many other ways that the combined functions of Mobile Portal and MCS can make it easy to serve appropriate content to a device without a lot of special coding on your part. This section highlights a few that are commonly used.

Automatic navigation selection

When a request is made, Mobile Portal and MCS determine the best way to present the overall navigation based on what it knows about the device capabilities. For instance, it decides whether to display icons, text, lists, or drop-down menus. You do not have to worry about navigational models, but if you want to you can customize the navigational model for any device.



Figure 18. Different navigational modes

Image selection

Another handy feature is automatic image selection. By using a MCS policy file for images, you can let MCS choose which image to return to a device. You can have several different images in your application - different sizes, formats, color density, whatever - and specify which devices get which images.

In our sample application shown in Figure 19, we have three color images of different sizes in our `images` folder. In the `mouse.mimg` policy file, we specify which devices get which image. You do not need to have a canvas layout defined for a particular device or device category to assign it a specific image.

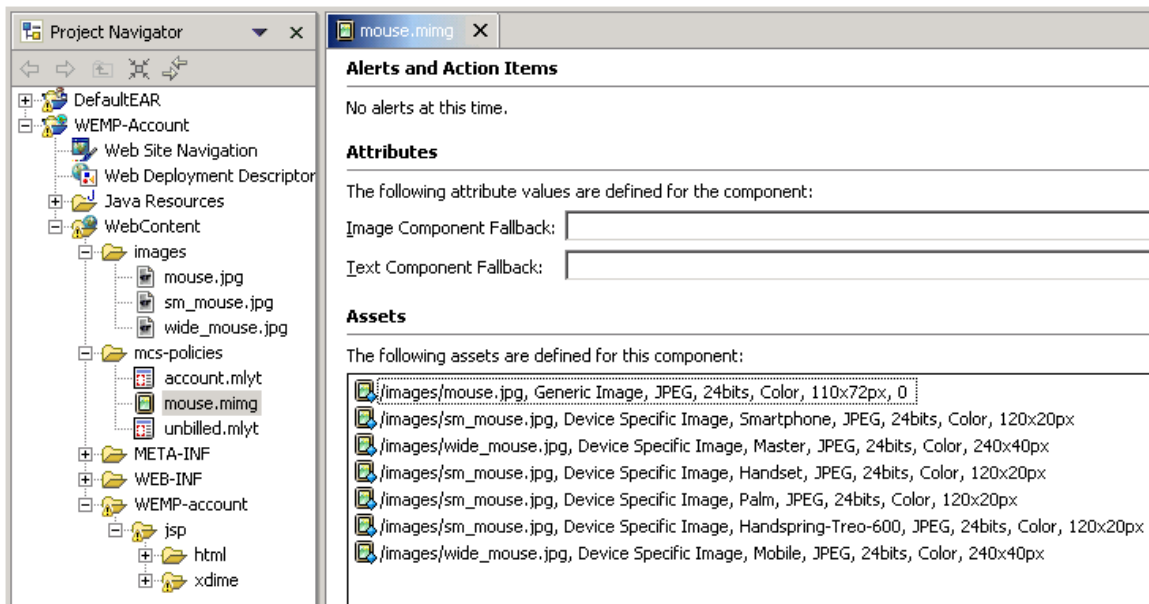


Figure 19. Images MCS policy file

Component selection and fallbacks

When you create the MCS policy file, you can also specify a *fallback* for that component. (See Figure 19.) A fallback is a resource that can be used if MCS fails to match the capability of device with a preferred component.

When MCS knows which asset, such as a layout or an image, is targeted for a particular device or group of devices, finding the asset to serve is simple. MCS searches up the hierarchy until either an appropriate asset or the root of the device tree is found.

But when no asset has been found and the component has a fallback specified, the search continues to try and find an asset that satisfies the fallback component. All assets can support an alternate text fallback. Some components support multiple fallbacks, and in this case, each fallback is searched in turn to attempt to find a suitable asset.

Type	Name	Audio	Chart	Dynamic	Image	Text	Link	Button	Rollover
	Audio								
	Chart								
Fallback	Dynamic								
	Image								
	Text								

Figure 20. Components and their fallbacks

In Figure 20, the shaded background shows the relationships between components and their possible fallbacks. For example Dynamic Visual components can have four fallback components: another dynamic visual plus audio, image, and text. An image can have another image and a text component as fallbacks.

Other helpful features

The Manage Mobile Pages interface extends the functions of WebSphere Portal administration. In addition to setting the properties for the Mobile Portal pages, in this interface you can also add preload notification to a page or to a portlet, or give them specific device requirements.



Figure 21. Mobile Portal administration extension

Preload notification

Preload notification is a feature that brings pop-up notices, commonly found on large browsers, down to mobile devices. You can use this feature to pop-up temporary pages on mobile devices, such as advertisements for special offers, warnings, license agreements, or notices of billable charges.

You create the pop-up pages apart from the rest of the project. They do not have to be built into the application, which keeps the programming separate from the presentation and gives you great flexibility. You link a pop-up notification to your mobile pages using WebSphere Portal administration functions and specify several properties, such as the time period for the notification and when and how often it pops-up. For example, you could specify that a notification pop-ups only the first time a customer accesses a certain page on every Wednesday in the month of January. You also have other choices, such as when the preload notice times out and whether or not the user can turn it off.

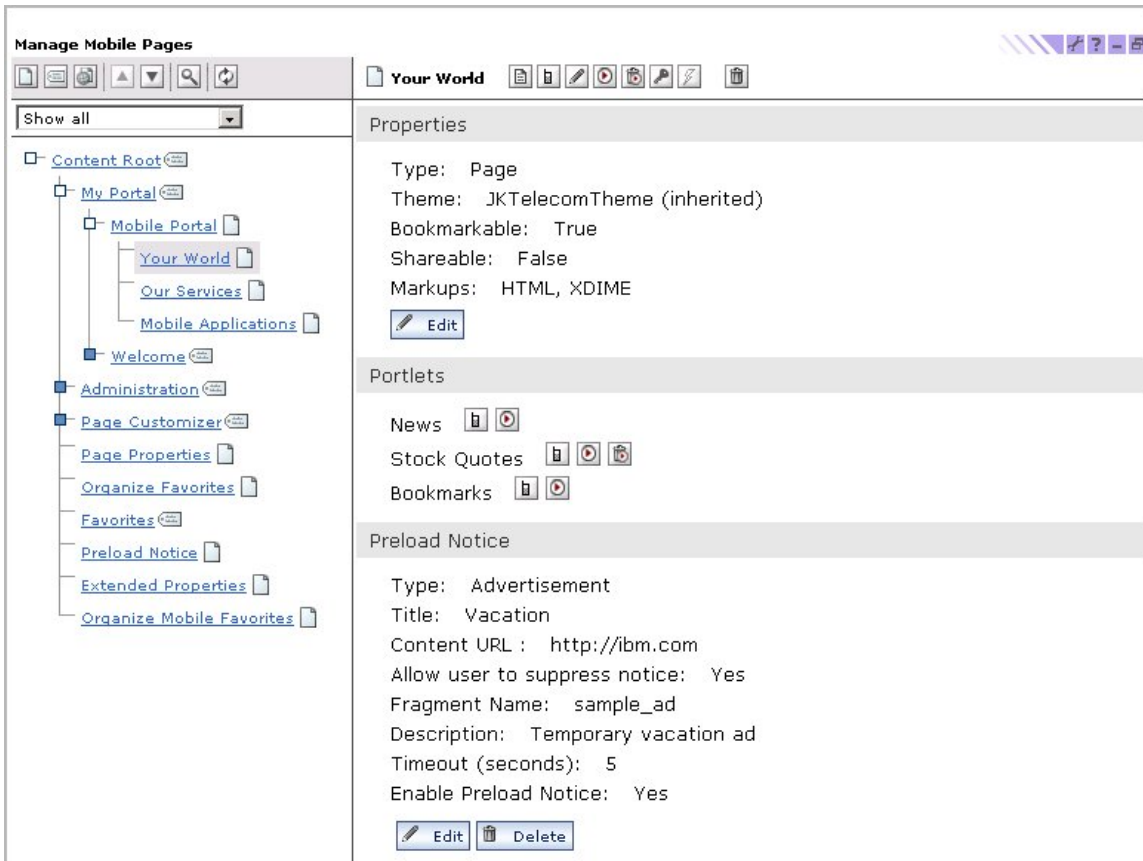


Figure 22. Preload notification added to the sample Your World page

Selective access by device or device category

In Manage Mobile Portal you can also change the Extended Properties to control which pages or portlets go to which devices. For example, if you had certain information that was appropriate for people who were using a particular model of a mobile device you could limit your pages only to those customers. Or, you could limit them to only those devices that have certain capabilities.

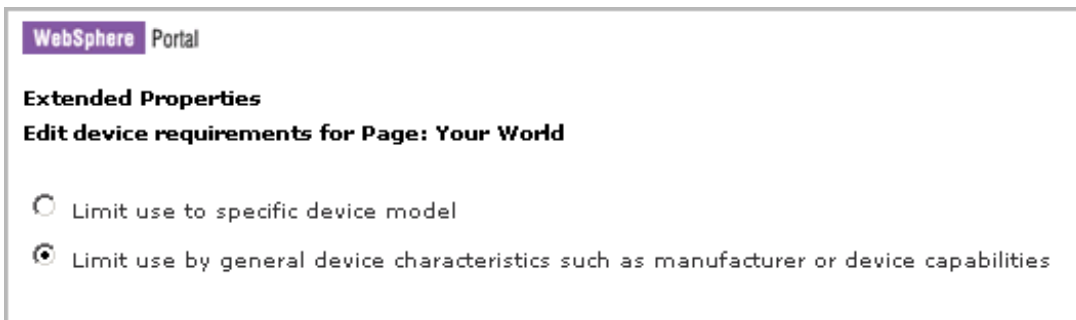


Figure 23. Limiting access to your portal pages by device

Conclusion

WebSphere Everyplace Mobile Portal can help you expand your business opportunities in a mobile on demand world. It has easy-to-use features that enable you to develop compelling Web applications, serve them to a wide range of devices, and know that they will continue to be effective in today's rapidly changing environment.

And, you can always stay ahead of the curve by subscribing to the update service for the device repository. This service automatically adds new devices to the repository for you, and puts them in their correct device category. MCS can then handle requests from these new devices with the templates you already assigned to their device category. You can be confident that your Web content reaches this ever-expanding market without having to spend valuable time tracking the specifications of each new device.

Mobile Portal is designed to take the future in stride. It is the fast path to Web mobility.

Resources

- For an overview of the product features and benefits, go to [WebSphere Everyplace Mobile Portal](#)
- To interact with a working Mobile Portal demo, see JK Telecom at the [Pervasive Computing Software Demo site](#)

© Copyright IBM Corporation 1994, 2004. All rights reserved.

IBM, WebSphere, and Everyplace are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.
