BEA WebLogic Workshop™ Help

Version 8.1 SP4
December 2004

Copyright

Restricted Rights Legend

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

Table of Contents

Assembling Portal Applications

Once you have developed the resources for portal user interfaces, you can assemble and configure portals using those resources. The topics in this section provide instructions for organizing a desktop, determining the navigation used by a desktop, and changing a desktop's user interface behavior and appearance.

Portal Key Concepts and Architecture

Provides a graphic that illustrates the flexibility and extensibility of the WebLogic Portal architecture.

Creating a Portal File

Shows you how to create a portal file so you can begin creating portal desktops in WebLogic Workshop and the WebLogic Administration Portal.

Adding Books and Pages to a Portal

Shows you how to organize your portal content by adding books and pages to a portal desktop.

Changing a Page Layout

Shows you how to change a page layout to use a different number and configuration of table cells for portlet and book placement on the page.

Setting up Navigation

Shows you how to change book menus to affect the way page and book navigation tabs are displayed.

Changing Look & Feel

Shows you how to change the skins and skeletons used to render a portal desktop by changing the look & feel used for the desktop.

Changing the Header and Footer

Shows you how to change the content of the desktop header and footer by changing the shell used for the desktop.

Properties for Portal Components

Provides reference information on the properties you can set in the Property Editor window for portal components.

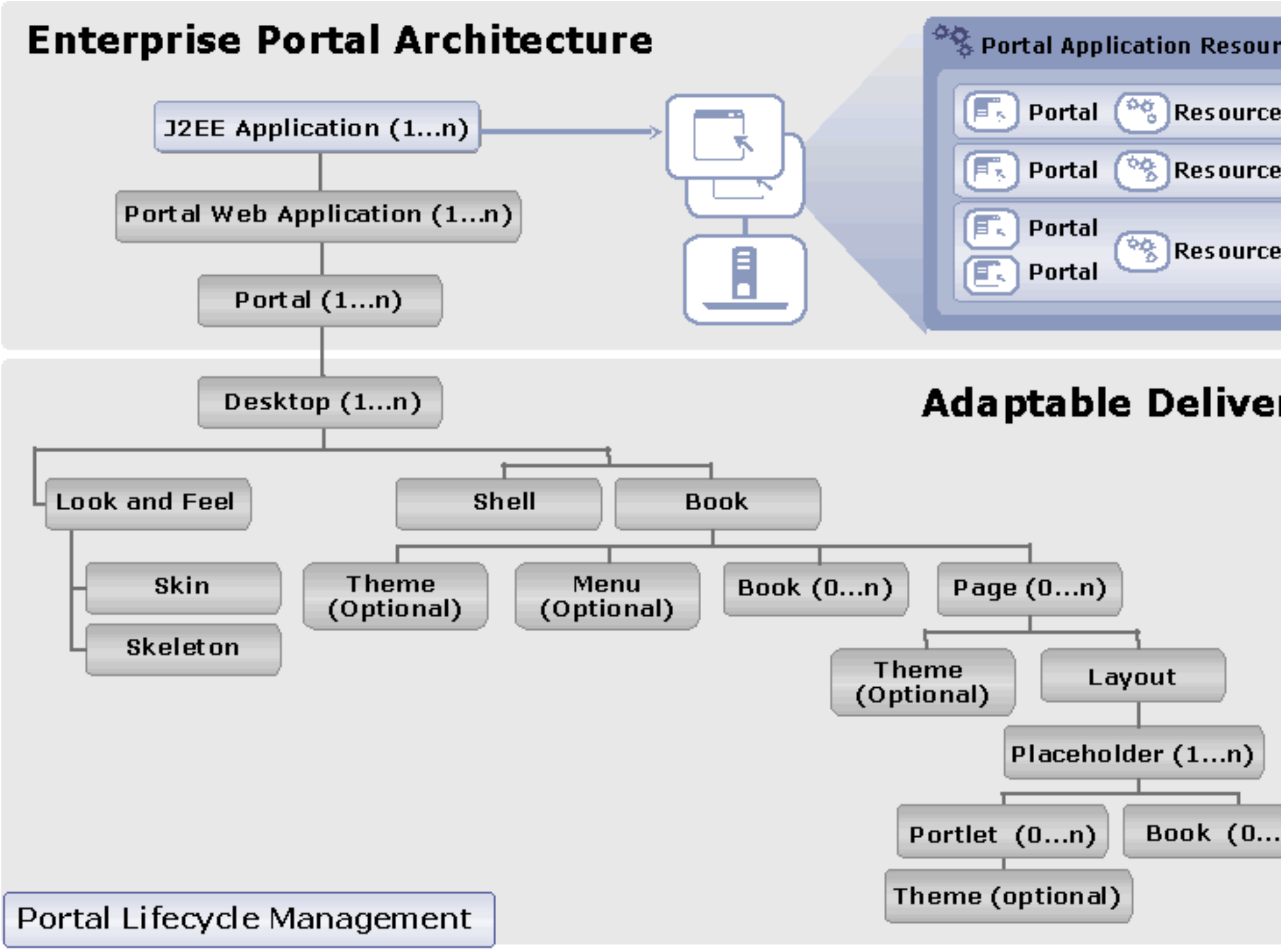
Related Topics

Developing Portal User Interfaces

Portal Key Concepts and Architecture

The following figure illustrates the flexibility and extensibility of the WebLogic Portal architecture. Whether you are building portal resources and templates in WebLogic Workshop or creating and administering portals with the WebLogic Administration Portal, you can work with individual components that are then unified by the portal framework.

In the figure, (1...n) means one or more, and (0...n) means zero or more. For example, a portal can contain one or more desktops. For resources that occur only once, like Look and Feel and Shell, you can still develop multiple versions even though only one at a time is allowed.



For descriptions of the portal components, see What is a Portal?.

What's Next

Getting Started

Creating a Portal File

The first step in creating a Portal is creating a Portal file in the Portal Designer. The file, which represents a portal desktop, appears in the Portal Designer as a graphic representation of your desktop, letting you drag and drop books, pages, and portlets and set properties to construct your portal. The .portal file that is created, along with any .portlet files for corresponding portlets added to the portal, supply all the configuration information the portal framework needs to perform the rendering process.

The .portal files you create serve as templates that portal administrators use to create new desktops in the WebLogic Administration Portal.

1. Create a portal application.
2. With your portal application open in WebLogic Workshop Platform Edition, right−click a portal Web project in the Application window and choose ***New −−> Portal***.
3. In the New File window, enter the name of the portal file in the ***File name*** field. Be sure to keep the .portal extension. Click ***Create***.

   The new portal desktop appears in the Portal Designer with a default header, body, and footer. The main book contains a page.
4. Add books and pages to the Portal Desktop.
5. Apply layouts to pages.
6. Set up navigation for the Portal Desktop.
7. Select the default header and footer you want the desktop to use.
8. Add portlets to your pages.
9. Change the Look & Feel of your portal desktop.
10. In the Document Structure window, select each component and set its properties in the Property Editor window. See Properties for Portal Components for guidance.
11. Save the portal file.
12. To view your Portal in a browser, choose ***Portal−−>Open current portal***.

Samples

See the Portal Samples for instructions on viewing a sample portal file.

Related Topics

The Portal User Interface Framework

Tutorial: Building Your First Portal

Adding Books and Pages to a Portal

Setting up Navigation

Adding Portlets to Pages

Changing Look & Feel

Deploying Portal Applications

Adding Books and Pages to a Portal

You can organize the information on your portal desktop by adding books and pages. Each book and page has its own link or tab to access it.

For example, you could add the following to your Portal:

- A Page called "Company Information"
- A Book called "Human Resources" that contains two pages: "Benefits" and "Forms." You could even make "Forms" a sub−book instead of a page and add sub−pages to it such as "New Hire Forms" and "Personal Information Change Forms."

You can also rearrange and delete books and pages and enable book editing for users.

This topic contains the following sections:

To add books and pages to a portal desktop

To rearrange books and pages

To delete books and pages

To enable book editing

To add books and pages to a portal desktop

1. Open a Portal file.
2. In the Palette window, drag and drop the ***Book*** control or the ***Page*** control into the area of the designer where you want it to appear.

   Note: If you want to use drop−down menu navigation (Multi Level Menu), drag the Book control into the tab area of a parent book rather than into a page.
3. In the Document Structure window, select the book or page and set its properties in the Property Editor window. Use the Book and Page Properties reference topic for guidance.

4. Save the Portal file.
5. Set up navigation for the portal desktop to control how the books and pages are accessed.
6. Add portlets to your pages.
7. Apply a Look & Feel to your Portal Desktop.

The books and pages you add to your portal in the Portal Designer become part of the .portal template that portal administrators can use to create portal desktops in the WebLogic Administration Portal.

To rearrange books and pages

You can change the order of books and pages. For example, if the main page book contains a page and a book in the following order:

Home Page | My Book

you can change the order to:

My Book* | *Home Page

To change the order of books and pages, right−click the book or page you want to move in the Document Structure window, and choose ***Move Up*** or ***Move Down***. The book or page moves up or down in the Document Structure window, and the horizontal reordering occurs in the Portal Designer.

Rearranging books and pages does *not* rearrange them in any portal desktops you have created with the WebLogic Administration Portal. See the WebLogic Administration Portal online help system for instructions on rearranging those books and pages.

To delete books and pages

To delete books and pages from a .portal file, right−click the book or page you want to delete in the Document Structure window and choose ***Remove Page*** for a page or ***Remove > This Book*** for a book. You can also delete a book's sub−books and pages by right−clicking the book and choosing ***Remove > <book/page name>***.

Deleting books and pages does *not* delete them from any portal desktops you have created with the WebLogic Administration Portal. See the WebLogic Administration Portal online help system for instructions on deleting those books and pages.

To enable book editing

You can create a JSP to let users modify book properties and let users access that edit JSP from a book.

1. Create the edit JSP and store it within the portal Web project.
2. Add a link to the edit page on the book. With the .portal file open in the Portal Designer, select the book you want users to be able to edit.
3. Set the editing link in one of the following ways in the Property Editor window:
    a. If your book uses a menu style (in the Navigation field), set the ***Editable*** field to ***Edit in menu***.
    b. If your book does not use a menu style, set the ***Editable*** field to ***Edit in titlebar***.
4. When you select an Editable value, the Property Editor window adds a new set of properties called Mode Properties. Enter values for these properties, especially ***Content URI***, which points to the edit JSP you created. Click the ellipsis icon [...] in this field to select your JSP.

   See Mode Properties in "Book and Page Properties" for more information.
5. Save the portal file.

When the desktop is rendered, the skeleton puts an edit icon in the table row of the menu or active book. When a user clicks the edit icon (activate), the edit page appears and the edit icon changes to a deactivate icon.

The names of the activate and deactivate icon graphics are registered in the project's WEB−INF/netuix−config.xml file under the <window−mode name="edit"> entry. The actual graphics are stored in the skin's /images subdirectory.

Samples

See Portal Samples for instructions on viewing a sample portal file.

Related Topics

The Portal User Interface Framework

Setting up Navigation

Adding a Portlet to a Portal

Changing Look & Feel

What is a Portal?

WebLogic Administration Portal Online Help

Changing a Page Layout

Layouts provide placeholders on a page for adding books and portlets. Different layouts display books and portlets on a page in different locations. For example, one layout can display portlets in two columns; another can display portlets in three columns and two rows.

Changing the layout used on a page

1. With the portal file open, select a page in the Document Structure window.
2. In the Property Editor window, select a different layout in the ***Layout Type*** field.

   You can also right−click the page in the Document Structure window and choose ***Layout > <layout name>***.
3. After the layout is re−rendered on the page in the portal designer, you may need to rearrange the portlets on the page by dragging and dropping them.
4. Save the portal file.

The vertical or horizontal placement of books and portlets in a placeholder is determined by properties set in the .layout file for the selected layout.

Samples

See Portal Samples for instructions on viewing a sample portal file.

Related Topics

Creating Layouts

Creating a Portal File

Adding Books and Pages to a Portal

Adding a Portlet to a Portal

Setting up Navigation

Changing Look & Feel

The Portal User Interface Framework

Changing the Header and Footer

Setting up Navigation

After you have created a portal and added books and pages to the portal desktop, you should determine the type of navigation to use between books and pages and set the default page that appear when users access the Desktop.

The main body of a desktop is contained within a main page book that is created automatically when you create a portal. You set navigation properties on this main page book.

To set up navigation on books

1. With the Portal file open, use the Document Structure window to select the book you want.
2. In the Property Editor window, select one of the following from the ***Navigation*** field:
    - ♦ Single Level Menu − Provides a single row of tabs for navigation among books and pages.
    - ♦ Multi Level Menu − Provides multiple levels of nested tabs for navigating among books and pages. Sub−books and pages are accessed through a cascading drop−down menu. Drop−down functionality occurs when books are added directly to books rather than to placeholders on pages.
    - ♦ No Navigation − Suppresses the sub−book and pages tabs in the book. This option is useful, for example, if you use the Targeted Menu Portlet or the Left Navigation Shell for book navigation.

   Note: If you created your own navigation menus by copying and modifying the default menus, they are also available for selection.
3. Save the portal file.

Samples

You can use the Portal Samples for to change and view navigation modifications. The Tutorial Portal also contains different Look & Feel implementations.

Related Topics

How Do I: Use Drop−Down Navigation in Portals?

Tutorial: Changing a Portal's Look & Feel and Navigation

Modifying Navigation Menus

The Portal User Interface Framework

Changing Look & Feel

What is a Portal?

Changing Look & Feel

The physical appearance of a portal is determined by the Look & Feel selected for the portal desktop. Look & Feels are a combination of skins, themes, and skeletons that control the structure, portlet titlebar graphics, JavaScript behavior, and HTML styles in your portal desktops.

You can change a the Look & Feel of an entire desktop or of individual components in the desktop in the following ways:

- Select a Look & Feel for the desktop
- Apply Themes to Books, Pages, and Portlets
- Override Style Sheet Classes and Styles on individual components

To select a Look & Feel for the desktop

Selecting a Look & Feel for a desktop provides a uniform Look & Feel to all components in the desktop.

1. With the portal file open, select the ***Desktop*** icon in the Document Structure window.
2. In the Property Editor window, select a different Look & Feel in the ***Look and Feel*** field.
3. Save the portal file.

To apply themes to books, pages, and portlets

Themes are subsets of skins and skeletons. They can contain their own graphics, cascading style sheets, and JavaScript behavior, and they can provide skeleton JSPs that render components differently than the parent skeleton JSPs. Themes can be applied to books, pages, and portlets.

While you can select an available theme for any book, page, or portlet, the theme will only render if the skin and/or skeleton for the selected look & feel contains the necessary theme subdirectory and files to make the theme work.

For more information, see Creating Skins and Skin Themes and Creating Skeletons and Skeleton Themes.

1. With the portal file open, select a book, page, or portlet.
2. In the Property Editor, select a theme in the ***Theme*** field.
3. Save the portal file.

To override style sheet classes and styles on individual components

You can manually override style sheet classes and styles on individual portal components using the Property Editor window.

1. With the portal file open, use the Document Structure window to select the component you want.
2. In the Property Editor window, enter the class and style values you want in the ***Presentation Class*** and ***Presentation Style*** fields. See the Properties for Portal Components reference topic for more information.
3. Save the portal file.

Samples

You can use the the Portal Samples for to change and view Look & Feel modifications. The Tutorial Portal also contains different Look & Feel implementations.

Related Topics

How Look & Feel Determines Rendering

Tutorial: Changing a Portal's Look & Feel and Navigation

Creating Look & Feel Resources

Creating Skins and Skin Themes

Creating Skeletons and Skeleton Themes

Creating a Portal File

Adding Books and Pages to a Portal

Changing a Page Layout

Setting up Navigation

Changing the Header and Footer

Changing the Header and Footer

You can determine the content that appears in the header or footer of a desktop (typically above and below the body containing the books, pages, and portlets). For example, you can set up a campaign to display a piece of Web content in the desktop header when a user logs in.

Header and footer content is determined by the shell selected for the desktop.

To change the shell used for a desktop

1. Open your portal file in the WebLogic Workshop Platform Edition Portal Designer.
2. In the Document Structure window, select the ***Desktop*** icon.
3. In the Property Editor window, use the ***Shell*** field to select the shell containing the header and footer you want to use.
4. Save the portal file.

You add content to a header or footer by creating or modifying a shell. See Creating Shells.

Samples

See Portal Samples for instructions on viewing a sample portal file.

Related Topics

How the Shell Determines Header and Footer Content

How Do I: Personalize a Desktop Header or Footer

Creating a Portal File

Adding Books and Pages to a Portal

Changing a Page Layout

Setting up Navigation

Building Portlets

Changing Look & Feel

The Portal User Interface Framework

Creating Shells

Properties for Portal Components

As you modify a portal file in WebLogic Workshop Platform Edition, you can use the Property Editor window to set component properties. The following topics describe the properties you can view and set for the following components:

Properties for All Portal Components

Desktop Properties

Header and Footer Properties

Book and Page Properties

Placeholder Properties

Portlet Properties

Samples

The WebLogic Workshop Portal Extensions contain sample portals you can open in WebLogic Workshop Platform Edition to view portal properties in the Property Editor window. See Portal Samples.

Properties for All Portal Components

As you modify a .portal or .portlet file in WebLogic Workshop Platform Edition, you can use the Property Editor window to set portal and portlet properties. The following table describes the common properties you can set for all portal components.

Administration Properties	
Markup Name	Read−only. The unique name of a component markup type. For example, three shell files (markup type "shell") must each use a unique Markup Name inside their .shell files. (Because desktops, books, and pages do not have associated .desktop, .book, and .page files, the exact same Markup Name is used for each.)
Presentation Properties	
Presentation Class	Optional. Typically a style sheet style, or class. Overrides the class attribute that would otherwise be used by the component's skeleton. For proper rendering, the class must exist in a cascading style sheet (CSS) in the desktop's selected skin, and the skin's skin.properties file must reference the CSS file. Sample − If you enter my−custom−class, the rendered HTML from the default skeletons will look like this: <div class="my−custom−class"> The properties you enter are added to the component's parent <div> tag. On books, pages, and portlets, use the Content Presentation Class property to set properties on the component's content/child <div> tag, especially for setting a style class that enables content scrolling and height−setting.
Presentation ID	Optional. A unique ID inserted in the rendered HTML tag for the component. The value you enter (which must be unique among all presentation IDs in the portal) overrides the ID that might otherwise be inserted by the component's skeleton. An example use would be inserting a unique ID that JavaScript could operate on. Sample − If you enter 12345, the rendered HTML from the default skeletons will look like this: <div id="12345">

Presentation Style	Optional. HTML style attribute to insert for the portal component. This attribute is equivalent to a style sheet class attribute and overrides any attributes in the style sheet class. Separate multiple entries with a semicolon. Sample − If you enter {background−color: #fff} for a portlet titlebar, the rendered HTML from the default skeletons will look like this: <div class="bea−portal−window−titlebar" style="{background−color: #fff}"> and the portlet titlebar will have a white background. The background−color attribute you entered overrides the background−color attribute in the bea−portal−window−titlebar class. The properties you enter are added to the component's parent <div> tag. On books, pages, and portlets, use the Content Presentation Style property to set properties on the component's content/child <div> tag, especially for setting content scrolling and height.
Skeleton URI	Optional. The path (relative to the project) to a skeleton JSP that will be used to render the portal component. This JSP overrides the skeleton JSP that would otherwise be used by the selected Look & Feel for the desktop. For example, enter /frameworks/myskeletons/mytitlebar.jsp.

Related Topics

Desktop Properties

Header and Footer Properties

Book and Page Properties

Placeholder Properties

Portlet Properties

Desktop Properties

When you select the desktop in the Portal Designer, the following properties appear in the Property Editor window:

Title	Required. Enter a title for the desktop. The title is used only for labeling in the Portal Designer.
Look and Feel	Required. Select the Look & Feel to determine the default desktop appearance (combination of skins and skeletons).
Shell	Required. Select the default shell for the area outside of the books, pages, and portlets. Shells determine the content for the desktop header and footer.

Related Topics

Properties for All Portal Components

Header and Footer Properties

Book and Page Properties

Placeholder Properties

Portlet Properties

Header and Footer Properties

When you select the "Content" node under the Header or Footer node in the Portal Designer's Document Structure window, the following properties appear in the Property Editor window:

Content URI	Read−only. The JSP file referenced in a shell that is used to display content in the desktop header or footer. This value is read from the <netuix:jspContent> "contentUri" attribute in the .shell file <header> or <footer> tag. If you select a different shell for the desktop, you may see a different value here.
Error Page URI	Read−only. The file referenced in a shell that is used to display an error message in the desktop header or footer if the contentUri JSP encounters errors. This value is read from the <netuix:jspContent> "errorUri" attribute in the .shell file <header> or <footer> tag. If you select a different shell for the desktop, you may see a different value here.
Backing File	Read−only. The class referenced in a shell that is used for preprocessing prior to rendering a shell's header or footer JSP. This value is read from the <netuix:jspContent> "backingFile" attribute in the .shell file <header> or <footer> tag. If you select a different shell for the desktop, you may see a different value here.

Related Topics

Properties for All Portal Components

Desktop Properties

Book and Page Properties

Placeholder Properties

Portlet Properties

Book and Page Properties

When you select a book or page in the Portal Designer, the following properties appear in the Property Editor window.

The term "hint" in the descriptions means available capabilities that are not supported in the default skeletons provided with the WebLogic Workshop Portal Extensions.

Title	Required. Enter a title for the book or page. Page titles are used for the page tabs/navigation menus.
Theme	Optional. Select a theme to give the book or page a different look and feel than the rest of the desktop.
Definition Label	Required. Unique identifier for each book or page. A default value is entered automatically, but you can change the value. Each book or page must have a unique Definition Label. Definition Labels can be used to navigate to books or pages. Also, components must have Definition Labels for entitlements and delegated administration.
Backing File	Optional. If you want to use a class for preprocessing (for example, authentication) prior to rendering the book or page, enter the fully qualified name of that class. That class should implement the interface com.bea.netuix.servlets.controls.content.backing.JspBacking or extend com.bea.netuix.servlets.controls.content.backing.AbstractJspBacking. See the Tutorial Portal in the Portal Samples for examples of backing files.
Unselected Image	Optional. Select an image to override the icon that appears next to the book or page title. This image appears on the tab of unselected pages.
Selected Image	Optional. Select an image to override the icon that appears next to the book or page title. This image appears on the tab of selected pages.
Rollover Image	Optional. Select an rollover image for the icon that appears next to the book or page title. This image appears when the mouse rolls over the tabs of unselected pages.
Orientation	Optional. Hint to the skeleton to position the navigation menu on the top, bottom, left, or right side of the book. You must build your own skeleton to support this property. Following are the numbers used in the .portal file for each orientation value: top=0, left=1, right=2, bottom=3.
Packed	Optional. Rendering hint that can be used by the skeleton to render the book or page in either expanded or packed mode. You must build your own skeleton to support the property. When packed="false" (the default), the book or page takes up as much horizontal space as it can. When packed="true," the book or page takes up as little horizontal space as possible.

	From an HTML perspective, this property is most useful when the window is rendered using a table. When packed="false," the table's relative width would likely be set to "100%." When packed="true," the table width would likely remain unset.
Hidden	Optional. Hides the navigation tab for the book or page to prevent direct access. You can access the page or book with a link (to the definition label) or by using a backing file.
Default Page (Book only)	Required. Select the page that appears by default when the desktop is accessed. The list is populated with Definition Labels of all pages in the portal.
Return to Default Page (Book only)	Optional. Determines the page displayed when a book is selected. When Return to Default Page="false" (the default), the last page that was active in a book is displayed when the book is selected. When Return to Default Page="true," the page selected in the Default Page property is always displayed when a book is selected.
Content Presentation Class	Optional. The primary uses are to allow content scrolling and content height−setting. For scrolling, enter a stylesheet class that uses one of the following attributes: • overflow−y:auto − Enables vertical (y−axis) scrolling • overflow−x:auto − Enables horizontal (x−axis) scrolling • overflow:auto − Enables vertical and horizontal scrolling For setting height, enter a stylesheet class that uses the following attribute: height:200px where 200px is any valid HTML height setting. You can also set other style properties for the content as you would using the Presentation Class property. The properties are applied to the component's content/child <div> tag.
Content Presentation Style	Optional. The primary uses are to allow content scrolling and content height−setting. For scrolling, enter one of the following attributes: • overflow−y:auto − Enables vertical (y−axis) scrolling • overflow−x:auto − Enables horizontal (x−axis) scrolling • overflow:auto − Enables vertical and horizontal scrolling For setting height, enter the following attribute: height:200px

	where 200px is any valid HTML height setting. You can also set other style properties for the content as you would using the Presentation Style property. The properties are applied to the component's content/child <div> tag.
Layout Type (Pages only)	Required. Select the page layout style for positioning books and portlets in placeholders on a page.
Navigation (Book only)	Required. Select the default type of menu to use for navigation among books and pages.
Editable (Book only)	Optional. If you have visitor tools enabled so that users can modify book properties, setting Editable to "Edit in Titlebar" or "Edit in Menu" puts a visitor tool link in that location. "Edit in Menu" is only available if you select a menu type for the Navigation property. When you select "Edit in Titlebar" or "Edit in Menu," a group of Mode Properties appears in the Property Editor.
Mode Properties (edit mode properties available on books when you select "Edit in Titlebar" or "Edit in Menu" for the Editable property)	
Content URI	Required. The path (relative to the project) to the JSP or HTML file to be used for book's edit page. For example, if the content is stored in <project>/edit/editBook.jsp, the Content URI is /edit/editBook.jsp.
Backing File	Optional. If you want to use a class for preprocessing (for example, authentication) prior to rendering the book's edit page, enter the fully qualified name of that class. That class should implement the interface com.bea.netuix.servlets.controls.content.backing.JspBacking or extend com.bea.netuix.servlets.controls.content.backing.AbstractJspBacking. See the Tutorial Portal in the Portal Samples for examples of backing files.
Error URI	Optional. The path (relative to the project) to the JSP or HTML file to be used for the error message if the book's edit page cannot be rendered. For example, if the error page is stored in <project>/exception/errorBookEdit.jsp, the Content URI is /exception/errorBookEdit.jsp.
Name	Enter the name of the mode, such as "Edit".
Visible	Optional. Makes the edit icon in the titlebar or menu invisible (false) or visible (true). Set Visible to "false" when, for example, you want to provide an edit URL in a desktop header.

Related Topics

Properties for All Portal Components

Desktop Properties

Header and Footer Properties

Placeholder Properties

Portlet Properties

Placeholder Properties

When you select a placeholder in the Portal Designer, the following read−only properties appear in the Property Editor window. The property values are read in from the .layout file that corresponds to the selected Layout Type for the page.

Title	Read−only. The name of the placeholder. This value is read from the .layout file for the page's selected Layout Type.
Flow	Read−only. If the "Using Flow" property is set to true, this value can be "vertical" or "horizontal." Flow determines whether books or portlets put in the placeholder are positioned on top of each other (vertical) or beside each other (horizontal). This value is read from the .layout file for the page's selected Layout Type.
Using Flow	Read−only. If this value is set to "true," books and portlets put in the placeholder are positioned according to the value of the "Flow" property. If this value is set to "false," the default flow is used (vertical). This value is read from the .layout file for the page's selected Layout Type.
Placeholder Width	Read−only. Displays the width set for the placeholder. This value is read from the .layout file for the page's selected Layout Type.

Related Topics

Properties for All Portal Components

Desktop Properties

Header and Footer Properties

Book and Page Properties

Portlet Properties

Portlet Properties

This topic describes the properties you can set on portlets in the Portal and Portlet designers.

Portlet Properties in the Portal Designer

Portlet Properties in the Portlet Designer

The term "hint" in the property descriptions means available capabilities that are not supported in the default skeletons provided with the WebLogic Workshop Portal Extensions.

Portlet Properties in the Portal Designer

When you select a portlet instance in the Portal Designer (as opposed to opening the .portlet source file), the following properties appear in the Property Editor window.

Title	Required. Enter a title only if you want to override the default title provided by the .portlet file. The title is used in the portlet titlebar.
Instance Label	Required. A single portlet, represented by a .portlet file, can be used multiple times in a portal. Each use of that portlet is a portlet instance, and each portlet instance must have a unique ID, or Instance Label. A default value is entered automatically, but you can change the value. Instance Labels are necessary for inter−portlet communication between Java Page Flow portlets. Also, portlets must have Instance labels for entitlements and delegated administration.
Portlet URI	Required. The path (relative to the project) of the parent .portlet file. For example, if the file is stored in <project>\myportlets\my.portlet, the Portlet URI is /myportlets/my.portlet.
Theme	Optional. Select a theme to give the portlet a different look and feel than the rest of the desktop.
Orientation	Optional. Hint to the skeleton to position the portlet titlebar on the top, bottom, left, or right side of the portlet. You must build your own skeleton to support this property. Following are the numbers used in the .portal file for each orientation value: top=0, left=1, right=2, bottom=3. Change the value for this property only if you want to override the default orientation provided by the .portlet file.
Default Minimized	Optional. Select "true" for the portlet to be minimized when it is rendered. The default value is "false." Change the value for this property only if you want to override the default value provided by the .portlet file.

Portlet Properties in the Portlet Designer

When you open a portlet in the Portlet Designer, the following properties appear in the Property Editor window:

JSP Content	
Content URI	Required. The path (relative to the project) to the JSP or HTML file to be used for portlet's content. The Content URI was set when the portlet was created. You can use this property to point to a different file. For example, if the content is stored in <project>/myportlets/my.jsp, the Content URI is /myportlets/my.jsp.
Error Page URI	Optional. The path (relative to the project) to the JSP or HTML file to be used for the portlet's error message if the main content cannot be rendered. For example, if the error page is stored in <project>/myportlets/error.jsp, the Content URI is /myportlets/error.jsp.
Backing File	Optional. If you want to use a class for preprocessing (for example, authentication) prior to rendering the portlet, enter the fully qualified name of that class. That class should implement the interface com.bea.netuix.servlets.controls.content.backing.JspBacking or extend com.bea.netuix.servlets.controls.content.backing.AbstractJspBacking. See the Tutorial Portal in the Portal Samples for examples of backing files.
Thread Safe	Optional. Performance setting for books, pages, and portlets that use backing files. When Thead Safe is set to "true," an instance of a backing file is shared among all books, pages, or portlets that request the backing file. You must synchronize any instance variables that are not thread safe. When Thread Safe is set to "false," a new instance of a backing file is created each time the backing file is requested by a different book, page, or portlet.
Portlet Properties	
Title	Required. Enter the title that will appear in the portlet's titlebar. You can override this title in each instance of the portlet (in the Portal Designer).
Orientation	Optional. Hint to the skeleton to position the portlet titlebar on the top, bottom, left, or right side of the portlet. You must build your own skeleton to support this property in the .portal file. Following are the numbers used in the .portal file for each orientation value: top=0, left=1, right=2, bottom=3. You can override the orientation in each instance of the portlet (in the Portal Designer).

Packed	Optional. Rendering hint that can be used by the skeleton to render the portlet in either expanded or packed mode. You must build your own skeleton to support this property. When packed="false" (the default), the portlet takes up as much horizontal space as it can. When packed="true," the portlet takes up as little horizontal space as possible. From an HTML perspective, this property is most useful when the window is rendered using a table. When packed="false," the table's relative width would likely be set to "100%." When packed="true," the table width would likely remain unset.
Definition Label	Required. Unique identifier for the portlet. A default value is entered automatically, but you can change the value. Each portlet must have a unique Definition Label. Definition Labels can be used to navigate to portlets. Also, components must have Definition Labels for entitlements and delegated administration.
Default Minimized	Required. Select "true" for the portlet to be minimized when it is rendered. The default value is "false."
Render Cacheable	Optional. To enhance performance, set to "true" to cache the portlet. For example, portlets that call Web services perform frequent, expensive processing. Caching Web service portlets greatly enhances performance. Do not set this to "true" if you are doing your own caching.
Cache Expires (seconds)	Optional. When the "Render Cacheable" property is set to "true," enter the number of seconds in which the portlet cache expires.
Fork Render	Optional. Intended for use by a portal administrator when configuring or tuning a portal. Setting to "true" tells the framework that it should attempt to multithread render the portlet. This property can be set to true only if the "Forkable" property is set to "true."
Forkable	Optional. Lets a portlet developer determine whether or not the portlet is allowed to be multithread rendered. When set to "true," a portal administrator can use the "Fork Render" property to make the portlet multithread rendered.
Client Classifications	Optional. Select the multichannel devices in which the portlet can be viewed. The dialog list of devices comes from <project>\WEB−INF\client−classifications.xml. In the Manage Portlet Classifications dialog: 1. Select whether you want to enable or disable classifications for the portlet. 2. Move the client classifications you want to enable or disable from the Available Classifications to the Selected Classifications, and click OK.

	When you disable classifications for a portlet, the portlet is automatically enabled for the classifications you did not select for disabling.
Portlet Titlebar	
Icon URI	Optional. The path (relative to the project) to the graphic to be used in the portlet titlebar. You must create a skeleton to support this property.
Help URI	Optional. The path (relative to the project) to the portlet's help file.
Edit URI	Optional. The path (relative to the project) to the portlet's edit page.
Can Maximize	Optional. If set to "true," the portlet can be maximized.
Can Minimize	Optional. If set to "true," the portlet can be minimized.
Can Delete	Optional. If set to "true," the portlet can be deleted from a page.
Mode Properties (Available when you add a mode to a portlet)	
Content URI	Required. The path (relative to the project) to the JSP or HTML file to be used for portlet's mode content, such as the edit page. For example, if the content is stored in <project>/myportlets/editPortlet.jsp, the Content URI is /myportlets/editPortlet.jsp.
Backing File	Optional. If you want to use a class for preprocessing (for example, authentication) prior to rendering the portlet's mode page (such as the edit page), enter the fully qualified name of that class. That class should implement the interface com.bea.netuix.servlets.controls.content.backing.JspBacking or extend com.bea.netuix.servlets.controls.content.backing.AbstractJspBacking. See the Tutorial Portal in the Portal Samples for examples of backing files.
Error URI	Optional. The path (relative to the project) to the JSP or HTML file to be used for the error message if the portlet's mode page cannot be rendered. For example, if the error page is stored in <project>/myportlets/errorPortletEdit.jsp, the Content URI is /myportlets/errorPortletEdit.jsp.
Name	Enter the name of the mode, such as "Edit".
Visible	Optional. Makes the mode icon (such as the edit icon) in the titlebar or menu invisible (false) or visible (true). Set Visible to "false" when, for example, you want to provide an edit URL in a desktop header.
Presentation Properties	
Presentation Class Presentation ID Presentation Style Skeleton URI	See Properties for All Portal Components.
Content Presentation	Optional. The primary uses are to allow content scrolling and content height−setting.

Style	For scrolling, enter one of the following attributes: • overflow−y:auto − Enables vertical (y−axis) scrolling • overflow−x:auto − Enables horizontal (x−axis) scrolling • overflow:auto − Enables vertical and horizontal scrolling For setting height, enter the following attribute: height:200px where 200px is any valid HTML height setting. You can also set other style properties for the content as you would using the Presentation Style property. The properties are applied to the component's content/child <div> tag.
Content Presentation Class	Optional. The primary uses are to allow content scrolling and content height−setting. For scrolling, enter a stylesheet class that uses one of the following attributes: • overflow−y:auto − Enables vertical (y−axis) scrolling • overflow−x:auto − Enables horizontal (x−axis) scrolling • overflow:auto − Enables vertical and horizontal scrolling For setting height, enter a stylesheet class that uses the following attribute: height:200px where 200px is any valid HTML height setting. You can also set other style properties for the content as you would using the Presentation Class property. The properties are applied to the component's content/child <div> tag.

Related Topics

Properties for All Portal Components

Desktop Properties

Header and Footer Properties

Book and Page Properties

Placeholder Properties