BEA WebLogic Portal™

Portal Overview

Version 8.1 with Service Pack 4
Document Revised: December 2004

Copyright

Copyright © 2004-2005 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

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

BEA, BEA WebLogic Server, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic JRockit, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server Process Edition, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Contents

About This Document

What You Need to Know v

Product Documentation on the dev2dev Web Site v

Related Information vi

Contact Us! vi

Documentation Conventions vi

Portal Overview

What is a Portal? 1

WebLogic Portal 2

Anatomy of a Portal 3

Multiple Portals and Desktops in a Single Web Application 7

Security – Delegated Administration and Entitlements 7

WebLogic Portal Tools and Services 7

Portal Development Lifecycle 9

About This Document

This document describes portals and how the WebLogic Portal framework lets you create powerful portals quickly and easily.

What You Need to Know

This document is written for application developers who are familiar with Web technologies, object-oriented programming techniques, and the Java programming language.

Non-developers will also benefit from reading this document, which describes the place of WebLogic Portal in enterprise software systems.

Product Documentation on the dev2dev Web Site

BEA product documentation, along with other information about BEA software, is available from the BEA dev2dev Web site:

`http://dev2dev.bea.com`

To view the documentation for a particular product, select that product from the list on the dev2dev page; the home page for the specified product is displayed. From the menu on the left side of the screen, select Documentation for the appropriate release. The home page for the complete documentation set for the product and release you have selected is displayed.

Related Information

For general information about Java applications, go to the Sun Microsystems, Inc. Java Web site at `http://java.sun.com`.

For general information about XML, go to the O'Reilly & Associates, Inc. `XML.com` Web site at `http://www.xml.com`.

Contact Us!

Your feedback on the BEA WebLogic Portal documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the WebLogic Portal documentation.

In your e-mail message, please indicate that you are using the documentation for BEA WebLogic Portal Version 8.1.

If you have any questions about this version of BEA WebLogic Portal, or if you have problems installing and running BEA WebLogic Portal, contact BEA Customer Support at http://support.bea.com. You can also contact Customer Support by using the contact information provided on the quick reference sheet titled "BEA Customer Support," which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.
`monospace text`	Indicates *user input*, as shown in the following examples: • Filenames: `config.xml` • Pathnames: *`BEAHOME`*`/config/examples` • Commands: `java -Dbea.home=BEA_HOME` • Code: `public TextMsg createTextMsg(`
	Indicates *computer output*, such as error messages, as shown in the following example: `Exception occurred during event dispatching:java.lang.ArrayIndexOutOfBoundsException: No such child: 0`
`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
[ ]	Indicates optional items in a syntax line. The brackets themselves should never be typed. *Example*: `java utils.MulticastTest -n` *`name`* `[-p` *`portnumber`*`]`
|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed. *Example*: `java weblogic.deploy [list|deploy|update]`

Convention	Item
. . .	Indicates one of the following in a command line: • That an argument can be repeated several times in a command line • That the statement omits additional optional arguments • That you can enter additional parameters, values, or other information The ellipsis itself should never be typed. *Example*: `buildobjclient [-v] [-o name] [-f "file1.cpp file2.cpp file3.cpp . . ."`
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

Portal Overview

Welcome to WebLogic Portal. This section describes portals and how the WebLogic Portal framework lets you create powerful portals quickly and easily.

What is a Portal?

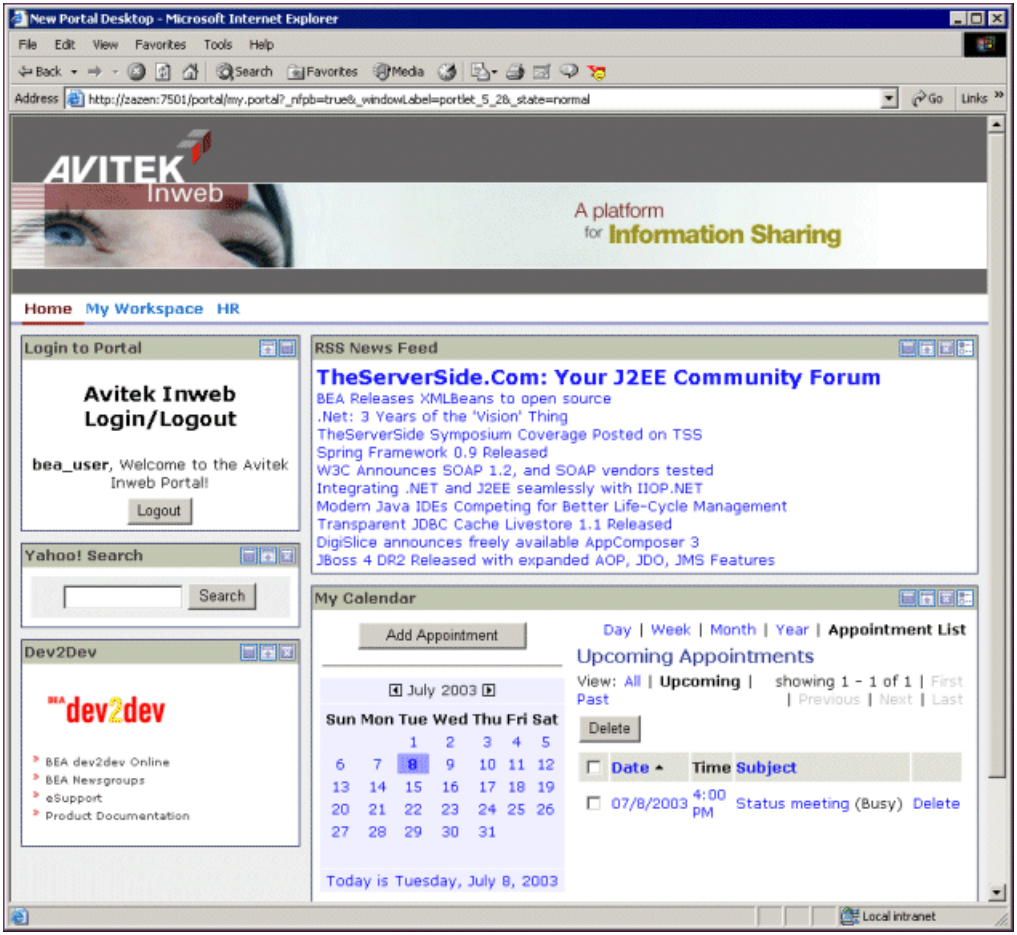
Since the dawn of the World Wide Web users have been accessing one Web page at a time. At first, excitement over the ability to provide platform-agnostic content to all users with network access or an Internet connection overshadowed any drawbacks or inadequacies with the new technology. Later, emerging technologies such as Java, JavaScript, and application servers provided application functionality, usability, stability, and performance improvements that have been the mainstay of Internet computing.

Now organizations need more. They want to not only surface their legacy applications, processes, and data in a Web interface, but they want to be able to do so more than one page at a time. They want portals.

A portal is a powerful Web site that gives users a single point of access to applications and information in a unified interface.

A portal lets users view each application or Web page in its own window, called a portlet, and a single browser window can contain multiple portlets. For example, a portal page can contain portlets for logging in, searching, displaying news feeds, and managing appointments with a calendar application, as shown in Figure 1.

Figure 1 Portal desktop



Portlets are arranged on portal pages, and users can easily navigate among pages with page tabs, drop-down menus, or other mechanisms to access the portlets they want. The portal in Figure 1 contains three pages: Home, My Workspace, and HR, whose links appear just above the Login portlet.

WebLogic Portal

WebLogic Portal's flexible, powerful framework lets you create portal interfaces independently of your application logic or Web pages, and through the portal development tools available in

WebLogic Workshop, you can surface the applications and Web services you develop seamlessly and easily in your portal interfaces.

In addition, WebLogic Portal's framework, lifecycle management tools, and business services let you quickly build and assemble portals that provide employees, partners, and customers with audience-specific, integrated views of applications, information, and business processes while enforcing business policies, processes, and security requirements.

WebLogic Portal's framework and tools are so powerful and easy to use that you can create a sophisticated, fully functional portal in minutes.

All pieces of a portal, from pages and portlets to a portal's Look & Feel elements, are individual components that can be developed quickly and independently, combined dynamically, and reused, giving portal developers, portal administrators, and end users the power to create and aggregate custom, audience-specific portals.

The following sections provide more details about WebLogic Portal's framework, tools, and services.

Anatomy of a Portal

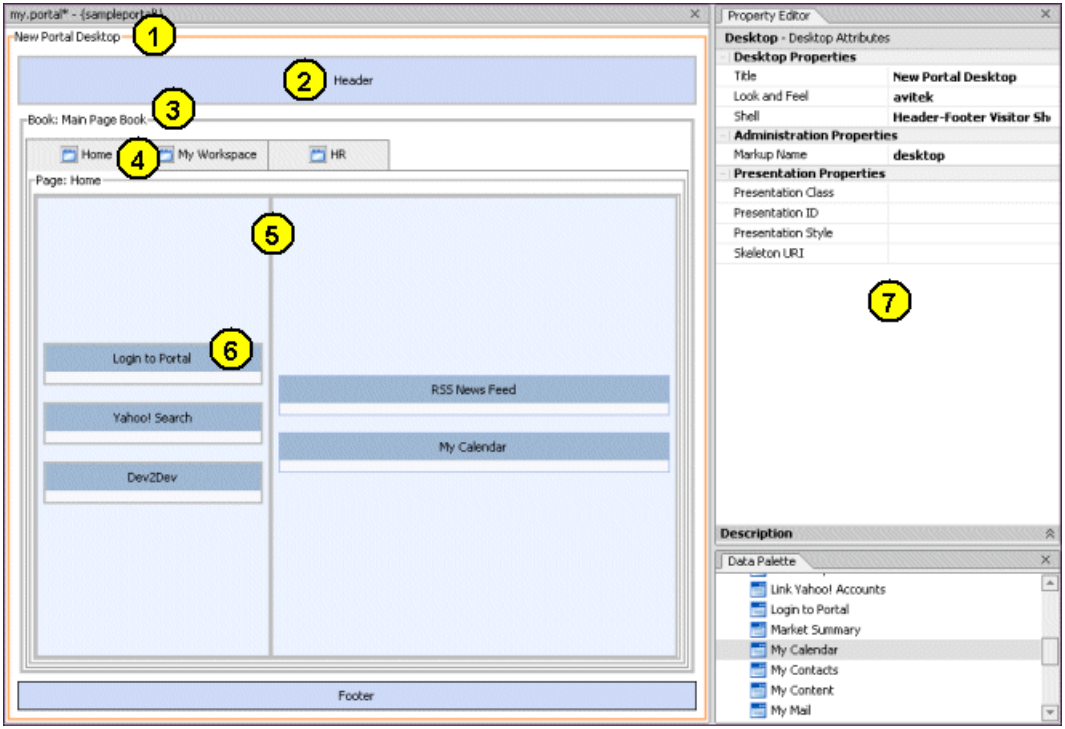
In WebLogic Portal, a portal is more than what appears in a browser. It is a collection of resources that can be assembled in different views called "desktops." A desktop is a user's view of a portal, and a single portal can contain multiple desktops. Technically speaking, Figure 1 is a desktop rather than a portal. The portal can contain many resources such as portlets that are not included in a desktop.

For example, the HR page in an employee desktop can display non-sensitive human resources portlets such as forms and holiday schedules. The HR page in a manager desktop can contain sensitive personnel portlets such as employee salary information and performance reviews. All portlets, sensitive and non-sensitive, are part of a single portal.

Figure 2 is the same portal desktop shown in Figure 1 open in the WebLogic Workshop Portal Designer where it was built. The portal desktop is constructed using resources available in the portal.

Figure 2 and table that follows it highlight key components of a portal desktop.

Figure 2 Portal desktop in the WebLogic Workshop Portal Designer



1	**Desktop (and Look & Feel)**	The desktop is the top-level container for the portal components included in that specific view of the portal. An important aspect of the flexible portal framework, the Look & Feel, comes into play at the desktop level. A Look & Feel is made up of two parts referenced by a single XML file: skins and skeletons. Skins contain the graphics, styles, and JavaScript code that determine the look of a desktop. Skeletons control the physical boundary rendering of all portal components. In the Property Editor window (item 7), you can select different Look & Feels for a desktop. Portal administrators and end users can also change a desktop's Look & Feel. Portal administrators can create new desktops beyond what portal developers create in WebLogic Workshop.
2	**Header and Footer (Shell)**	The desktop header and footer display content outside the desktop's books, pages, and portlets (typically above and below). A header/footer combination is defined by a shell, which is an XML file that points to JSP or HTML files containing the content to display (colors, graphics, personalized content, and so on). In the Property Editor window you can select different shells for the desktop. Portal administrators and end users can also change a desktop's shell.
3	**Top-level book (and Menus, Themes)**	The top-level book contains all sub-books, pages, and portlets. The top-level book defines the initial menu navigation style used for the desktop. For each sub-book you add to a desktop you can select a different navigation style. In the Property Editor window you can select different navigation menu styles for books. Portal administrators and end users can change the navigation style for books. You can also apply themes to books. Themes are Look & Feel subsets that can make a book look physically different than the rest of the desktop. Portal administrators and end users can also change themes.

4	**Pages and books (and Menus, Themes)**	Pages and sub-books are the navigable containers used for organizing portlets. You can apply different navigation menu styles to books, and you can apply themes to pages and books. Portal administrators and end users can change navigation menus and themes. Portal administrators can also create new pages and books beyond what portal developers create in the WebLogic Workshop Portal Designer.
5	**Layouts (and Placeholders)**	Layouts determine book and portlet positioning on pages. Layouts, defined by an XML file, are divided into cells, or placeholders, in which portlets and books are placed. In the Property Editor window you can select different layouts for a page. You can also determine whether portlets are placed horizontally or vertically relative to each other in a placeholder. Portal administrators and end users can change page layouts.
6	**Portlets (and Themes)**	Portlets are the containers that surface Web content and applications in your desktops. Each portlet, which you create in WebLogic Workshop with the Portlet Wizard or with the Portlet Designer, is a single XML file with a `.portlet` extension that references the content or application it will surface. Using the Portlet Designer and the Property Editor window, you can add portlet preferences and configure portlet modes (such as edit and help) that add powerful functionality to your portlets—all of which is included in the `.portlet` XML file. WebLogic Portal's flexible framework lets you reuse a portlet multiple times (create new instances of the portlet). The content of each portlet instance is automatically updated if the source `.portlet` file changes, but each instance of a portlet can be configured in unique ways (such as changing the titlebar label). Portal developers, administrators, and end users can apply themes to portlets.
7	**Property Editor window**	When you select a portal component in the Portal Designer, you can set properties for it in the Property Editor window. Most portal configuration in the development environment occurs in the Property Editor window, and changes are automatically written to the portal or portlet XML file.

Figure 3 shows more accurately how the portal components are related hierarchically. The Document Structure window in WebLogic Workshop shows the parent/child relationships

among portal components as they appear in the underlying `.portal` XML file, also shown in the figure. The XML is built automatically as you work in the Portal Designer.

Figure 3 Hierarchical structure of a portal desktop

Notice that the XML contains configuration attributes for the portal elements.

Multiple Portals and Desktops in a Single Web Application

The WebLogic Portal architecture lets you create multiple portals in a single Web application (portal Web project). In addition, each portal can contain multiple desktops, letting you flexibly provide customized content and applications to multiple audiences with reusable portal resources.

Security – Delegated Administration and Entitlements

WebLogic Portal extends the underlying WebLogic Server role and security policy architecture. With WebLogic Portal's reusable component framework, portal administrators can use dynamic, configurable policies to set up delegated administration on individual administration areas, tasks, and portal resources. Portal administrators also set entitlements on portal resources that control end-user access to portal desktops and content. For example, a portal administrator can create a desktop for regular employees and another for managers. With proper entitlements set on each desktop, employees who log in can view only the employee desktop, and managers can view either desktop. Another approach is to entitle portlets in a single desktop, where employees are allowed to view only the portlets to which they are entitled.

WebLogic Portal Tools and Services

In addition to the features already mentioned, WebLogic Portal also provides the following tools and services to help you easily create powerful, dynamic, feature-rich portals:

- **Multichannel Support** - WebLogic Portal includes a multichannel framework for fast, flexible development of portals for mobile devices. You can develop portals that simultaneously serve multiple devices.
- **Content Management** - BEA's Virtual Content Repository lets you combine and manage multiple BEA-compatible content management systems in a single interface. Any content in the Virtual Content Repository can be used in your portals. WebLogic Portal also provides a content repository and a reusable My Content portlet for uploading, managing, viewing, and searching content stored in the Virtual Content Repository.
- **Interaction Management** - WebLogic Portal provides enhanced, integrated tools in WebLogic Workshop and the WebLogic Administration Portal for adding personalization and campaigns to your portal applications and tracking user behavior in your portals.
- **Search** - The reusable Portal Search portlet provides configurable enterprise-wide search functionality. With WebLogic Portal's built-in search functionality you can build search indexes and perform file system, ODBC, and HTTP searches.
- **Collaboration** - The reusable collaboration portlets (My Mail, My Task List, My Calendar, My Contacts, and Discussion Forums) let you provide collaboration tools in your portals.
- **My Yahoo! Enterprise Edition portlets** - My Yahoo! Enterprise Edition lets you seamlessly blend more than 100 familiar My Yahoo! modules into your portals. Dynamic modules draw on Yahoo!'s 2,700 content sources from 25 regions in 13 languages, and include content and tools such as Headline News, Company News, Stock Portfolios, Market Summary, Weather, Sports, Maps, Travel, Notepad, Calendar, Package Tracker and much more.
- **Java Controls** - Pre-built Java controls let portal developers quickly add Java code to portal applications for functionality such as user creation, authentication, and property set management.
- **WebLogic Portal Visitor Tools** - With the WebLogic Portal Visitor Tools, end users can customize their portal desktops by adding and rearranging books, pages, and portlets and by changing the Look & Feel, page layouts, and header and footer (shell) used by their desktop.
- **Commerce** - You can add commerce to your portal applications using WebLogic Portal's commerce API and JSP tags.
- **APIs** - WebLogic Portal provides a full Java API. The Javadoc for the API can be found at http://e-docs.bea.com/workshop/docs81/doc/en/portal/buildportals/navReference.html.

- **Open Standards Support** - WebLogic Portal fully supports Java Specification Request (JSR) 168 Portlet Specification standards. The portlet specification enables interoperability between portlets and portals and defines a set of APIs that address the areas of aggregation, personalization, presentation, and security.

Portal Development Lifecycle

WebLogic Portal's framework, tools, and services provide a seamless, integrated portal development lifecycle. There are two major parts in the portal development lifecycle that have a symbiotic and iterative relationship with one another: development and administration.

- **Development** - WebLogic Portal extends WebLogic Workshop to allow easy surfacing of Web services, Java Page Flows, business process applications, and personalization functionality in portals that can be flexibly assembled for different audiences. Development involves building applications, standards-based components, and portal resources. Whether you are a J2EE developer creating EJBs or Web services, an application developer using the tag libraries to build JSPs for portlets, a portal designer creating desktops, or a graphic designer creating the icons, images, and style sheets to be used in a portal application, you are developing applications and portal resources.

  Using the WebLogic Workshop Portal Extensions, developers provide administrators with the resources that they in turn make available to users, such as portal desktops, pages, portlets, campaigns, and so forth.

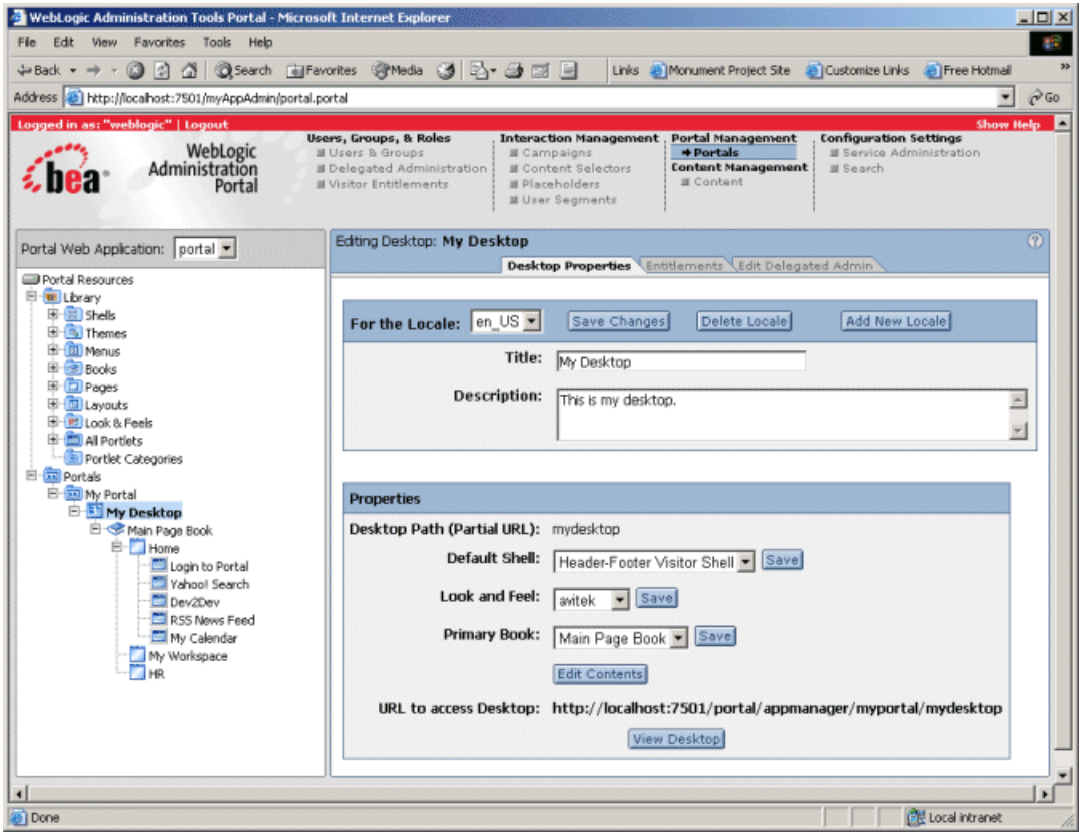
  Developers also create user profile properties (fields) that portal administrators can set for users, and they equip portals with the WebLogic Portal Visitor Tools to let end users customize their portal desktops.

- **Administration** - Portal administration involves portal aggregation and management using the WebLogic Administration Portal. Portal administrators use the resources created by developers to assemble and modify portals and prepare them for production. An administrator can, for example, use development-created resources such as desktops, pages, and portlets to construct a new portal; or modify the content query of a campaign. Administrators also create new portals and portal resources such as desktops, books, pages, and portlets. WebLogic Portal's component-based framework lets administrators set up delegated administration and define visitor entitlements that determine which users can administer and view specific portal components and content.

  Portal administration also involves other important tasks necessary to support the development environment. For example, an administrator must configure content repositories in BEA's Virtual Content Repository so that a developer can construct a query for a content selector, placeholder, or campaign. Administrators also add and manage the users that developers will target with their applications.

Figure 4 shows the WebLogic Administration portal. Notice that the desktop structure in the left resource tree matches that of the structure created in the WebLogic Portal Designer, as shown in WebLogic Workshop's Document Structure window in Figure 3.

Figure 4 WebLogic Administration Portal



Besides portal management, portal administration involves user and group management, setting up delegated administration and visitor entitlements, modifying interaction management (personalization and campaigns), managing content, and configuring server settings.

Where to Go Next

See the following documents for information on and instructions for creating and managing portals:

- Getting Started with Portal Development
- Getting Started with Portal Administration
- WebLogic Workshop Help System
- WebLogic Administration Portal Help System