



BEA WebLogic Portal™

Architectural Overview

Version 4.0
Document Date: October 2001

Copyright

Copyright © 2001 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, WebLogic, Tuxedo, and Jolt are registered trademarks of BEA Systems, Inc. How Business Becomes E-Business, BEA WebLogic E-Business Platform, BEA Builder, BEA Manager, BEA eLink, BEA WebLogic Commerce Server, BEA WebLogic Personalization Server, BEA Campaign Manager for WebLogic, BEA WebLogic Portal, E-Business Control Center, BEA WebLogic Process Integrator, BEA WebLogic Collaborate, BEA WebLogic Enterprise, and BEA WebLogic Server are trademarks of BEA Systems, Inc.

All other product names may be trademarks of the respective companies with which they are associated.

WebLogic Portal Architectural Overview

Document Edition	Date	Software Version
4.0	October 2001	WebLogic Portal 4.0

Contents

About This Document

What You Need to Know	vii
e-docs Web Site	vii
How to Print the Document	viii
Contact Us!	viii
Documentation Conventions	ix

1. Architectural Overview

WebLogic Portal in the BEA E-Business Platform.....	1-3
Web Applications, Enterprise Applications, and WebLogic Server Domains..	1-4
Your Web Application in Context	1-4
Types and Scope of Resources.....	1-6
URLs for Accessing Web Applications	1-8
The BEA_HOME and PORTAL_HOME Directories	1-9
The Reference Domain and Enterprise Applications	1-9
petflowDomain.....	1-10
p13nDomain.....	1-10
wlcsDomain	1-11
portalDomain	1-11
Non-Archived Enterprise Applications.....	1-12
Portals and Portlets	1-13
Relationships Among Applications and Portals.....	1-14
Portal Concepts	1-14
What Is a Group Portal?.....	1-15
What is a Portal Page?	1-15
What Is a Layout?	1-15
What Is a Portlet?.....	1-15

Designing the Appearance of Portlets	1-16
Personalization versus Customization	1-17
Visitor Entitlements	1-17
Delegated Administration	1-18
Flow Management with Webflow	1-18
Webflow as a State Machine	1-19
Namespaces	1-20
The Scope of Webflow Components	1-20
Creating, Modifying, and Synchronizing Webflow Components	1-21
For More Information	1-22
User and Group Management	1-22
Services for Registering Customers and Managing Customers	1-25
Customer and Group Profiles	1-26
Unified User Profile	1-27
Security	1-29
Security Realms	1-29
Security Roles and Declarative Security	1-30
Property Sets	1-31
Property Sets Serve as Namespaces for Properties	1-31
Where Property Sets Are Used	1-33
Rules	1-35
Customer Segments	1-35
Content Management	1-37
Content Management Systems	1-38
Content Selectors	1-39
Ad Placeholders	1-41
Comparison of Methods for Retrieving and Displaying Documents	1-42
The Advisor	1-45
Events and Behavior Tracking Service	1-46
Campaign Services	1-48
Campaigns and the Mail Service	1-50
Commerce Services	1-51
Product Catalog	1-51
Payment and Order Services	1-54
Discounts	1-56

Item Discounts and Order Discounts	1-56
Global Discounts and User Discounts	1-56
Internationalization Framework	1-58
Development Tools	1-59
WebLogic Server Administration Console	1-59
Accessing the WebLogic Server Administration Console.....	1-60
E-Business Control Center	1-62
WebLogic Portal Administration Tools	1-63
Accessing the WebLogic Portal Administration Tools.....	1-63
Logging In to the Administrator Account.....	1-65
Changing the Administrator Password	1-66

Index



About This Document

This document provides an overview of WebLogic Portal services and data structures.

What You Need to Know

Each developer on your team can use this document to understand how an individual component contributes to the overall application. The document assumes that you are familiar with features that are common to all Web sites and that you are aware of basic Java™ 2 Platform Enterprise Edition (J2EE) concepts.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the “e-docs” Product Documentation page at <http://e-docs.bea.com>.

How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser.

A PDF version of this document is available on the WebLogic Portal documentation Home page on the e-docs Web site. A PDF version of this document is also available in the documentation kit on the product CD. Or you can download the documentation kit from the WebLogic Portal portion of the BEA Download site. You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the WebLogic Portal documentation Home page, click the PDF files button and select the document you want to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at <http://www.adobe.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 the BEA WebLogic Portal 4.0 release.

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 through BEA WebSupport at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card, 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
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
<i>italics</i>	Indicates emphasis or book titles.
monospace text	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard. <i>Examples:</i> #include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float
monospace boldface text	Identifies significant words in code. <i>Example:</i> void commit ()
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> String <i>expr</i>

Convention	Item
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> LPT1 SIGNON OR
{ }	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. <i>Example:</i> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
...	Indicates one of the following in a command line: <ul style="list-style-type: none"> ■ 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. <i>Example:</i> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
.	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

1 Architectural Overview

While most WebLogic Portal documents focus on a specific development role and task, this document provides an overview of WebLogic Portal services and data structures. Each member of your team can use this document to understand how an individual component contributes to the overall application.

BEA WebLogic Portal comprises a set of services that enable you to efficiently build, launch, and maintain high-performance e-business sites. The following list highlights this functionality:

- Portal services provide a single solution for creating, deploying, and managing multiple enterprise portals. You can use these services to create a single integrated, personalized point-of-access for employees, customers, and business partners.
- Campaign services provide the ability to target advertising, e-mail, and product discounts to attain business goals. For example, a campaign might use e-mail to entice customers to your site to purchase specific items.
- Commerce services support such e-commerce functions as shopping carts, product catalogs, transaction management, and order fulfillment.
- Personalization services deliver dynamic and personalized Web site content.

All of these services include a distinct set of properties that you can tune to provide the best performance for your particular environment. In addition, you can use APIs to extend the functionality of WebLogic Portal services.

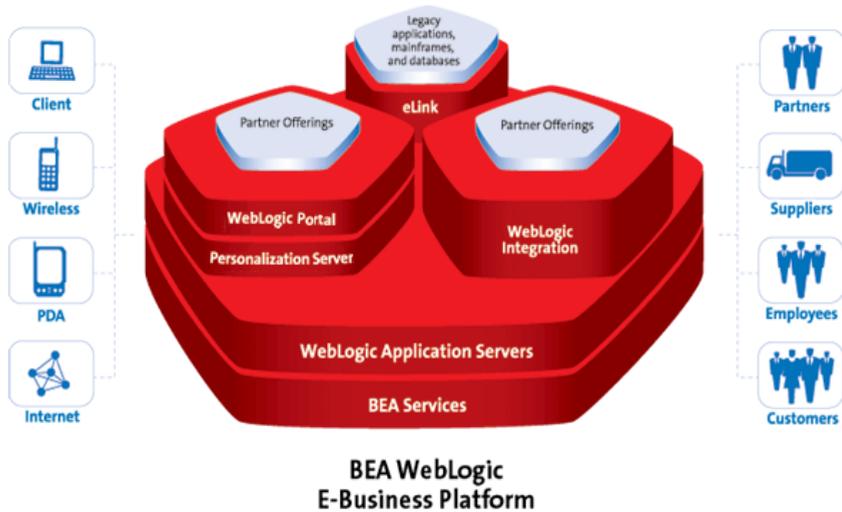
This topic focuses on the following architectural components of WebLogic Portal:

- WebLogic Portal in the BEA E-Business Platform
- Web Applications, Enterprise Applications, and WebLogic Server Domains
- Portals and Portlets
- Flow Management with Webflow
- User and Group Management
- Security
- Property Sets
- Rules
- Content Management
- Events and Behavior Tracking Service
- Campaign Services
- Commerce Services
- Internationalization Framework
- Development Tools

WebLogic Portal in the BEA E-Business Platform

WebLogic Portal is fully integrated with BEA WebLogic Server and is a part of the BEA E-Business Platform. (See Figure 1-1.)

Figure 1-1 BEA WebLogic E-Business Platform



Web Applications, Enterprise Applications, and WebLogic Server Domains

This section introduces the basic units for deploying, accessing, and managing the WebLogic Portal resources that you develop for your e-business Web site. It contains the following subsections:

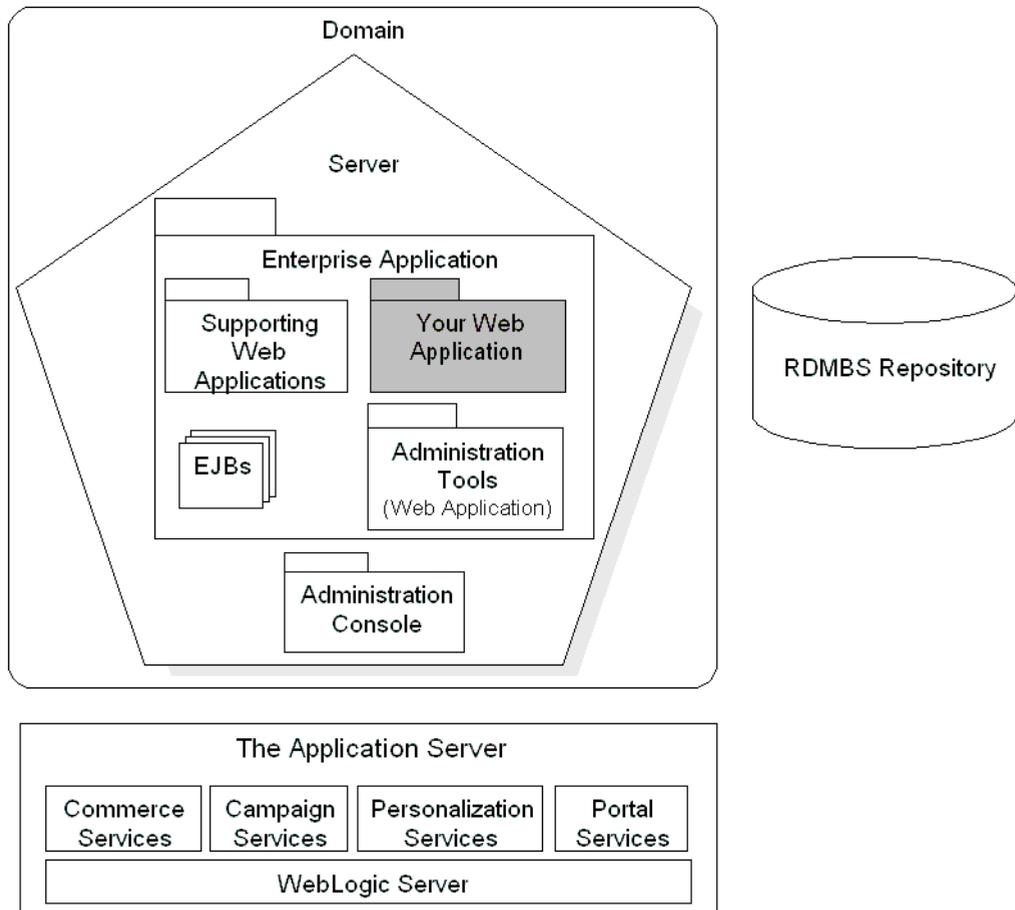
- Your Web Application in Context
- Types and Scope of Resources
- URLs for Accessing Web Applications
- The BEA_HOME and PORTAL_HOME Directories
- The Reference Domain and Enterprise Applications

Your Web Application in Context

You use a Web application to implement your e-business model. To share resources and properties across a set of Web applications and EJBs, you deploy your e-business Web application within an enterprise application. (See Figure 1-2.)

For example, you can use an enterprise application to share security settings between EJBs and Web applications (though individual modules can also contain a private set of resources and properties).

Figure 1-2 Your Web Application in Context



A WebLogic Server *domain* is the administrative unit that you use to manage enterprise applications. For example, a domain contains properties that define server instances, clusters, message logging, and connection pools. A *server configuration* is a set of properties that determines operating parameters for an active instance of your application server. For any given development project, you might create several different domain and server combinations, each configured to support various environments.

Types and Scope of Resources

All of the resources in a WebLogic Portal application are of one of the following types:

- EJBs and JSPs.

JSPs provide the presentation layer with which your customers interact. EJBs provide the data storage and processing layer and are scoped for an enterprise application.

- Content.

Content is the information that you present to your customers. You can add content, such as images and HTML fragments, directly to your JSPs or retrieve documents from a content-management system. WebLogic Portal includes a simple content management system and supports integrations with third-party systems.

- Data as described in Table 1-1.

Table 1-1 Type and Scope of Data That Supports a WebLogic Portal Application

Data Type	Created, Edited In	Stored In	Scope
<p>Configuration Data Determines how a server, enterprise application, or Web application interacts with the underlying platform. Examples: HTTP listen ports, cache sizes, connection-pool names.</p>	<p>WebLogic Server Administration Console, text editors</p>	<p>Properties files and deployment descriptors</p>	<p>Component and subcomponents. For example, server properties apply to the server and all applications that are deployed on the server. However, each subcomponent can override some of the properties from its parent container.</p>
<p>Application Data Specifies how services within an application behave to support your business goals. Examples: Rules, Property Sets, Webflows, Campaigns, portal and portlet metadata.</p>	<p>E-Business Control Center</p>	<p>XML files on a Developer's workstation. On the server, EJBs provide runtime representations and the RDBMS repository provides persistent storage.</p>	<p>The enterprise application, except for Webflows, which apply only to a single Web application. For more information, refer to "Scope of Application Data" under "Synchronizing Application Data" in the <i>Deployment Guide</i>.</p>
<p>Business Data Represents customers, products, and orders. Examples: customer profiles, product catalog, orders and order histories.</p>	<p>WebLogic Portal Administration Tools, scripts, third-party administration tools.</p>	<p>RDBMS repository. User and group data can also be stored in other locations such as an LDAP server.</p>	<p>The enterprise application.</p>

URLs for Accessing Web Applications

The URL of a WebLogic Portal Web application contains two components:

- The name of the server, which uses the following syntax:

```
http:// { localhost | IP-address | hostname }:listen-port
```

where *listen-port* is the port number that the server reports when it successfully starts. For example, when a server starts successfully the shell that runs the server process displays the following messages:

```
<Oct 15, 2001 9:21:49 AM MDT> <Notice> <WebLogicServer>  
<ListenThread listening on port 7501>
```

```
<Oct 15, 2001 9:21:49 AM MDT> <Notice> <WebLogicServer>  
<SSLListenThread listening on port 7502>
```

- The Web application's context root, which you declare in the parent enterprise application's `application.xml` file (deployment descriptor). For example, the sample `wlcs` Web application is deployed within the `wlcsApp` enterprise application. The deployment descriptor for `wlcsApp`, `PORTAL_HOME\applications\wlcsApp\META-INF\application.xml`, declares the following context root for `wlcs`:

```
<module>  
  <web>  
    <web-uri>wlcs</web-uri>  
    <context-root>wlcs</context-root>  
  </web>  
</module>
```

To access the `wlcs` Web application that is deployed on a server that uses `127.0.0.1` as an IP address and port `7501` as the listen port, use the following URL:

```
http://127.0.0.1:7501/wlcs
```

For more information about URLs and Web applications, refer to “Web Application Basics” in the *WebLogic Server Assembling and Configuring Web Applications* guide.

The BEA_HOME and PORTAL_HOME Directories

By default, when you install WebLogic Portal, the installer places all files under the BEA home directory, which this document refers to as `BEA_HOME`. In addition to WebLogic Portal, the `BEA_HOME` directory contains license files, BEA WebLogic Server™, and the JDK. All installations must include or refer to a `BEA_HOME` directory.

However, during installation you can specify that WebLogic Portal reside outside the `BEA_HOME` directory tree. For this reason, this document refers to the directory in which you install WebLogic Portal as `PORTAL_HOME`. This directory contains the Java classes that extend WebLogic Server. It also contains the reference domain and enterprise applications.

For more information about directory structure, refer to the following topics in the *Deployment Guide*:

- “Deploying Domains and Server Configurations”
- “Deploying Enterprise Applications”

The Reference Domain and Enterprise Applications

WebLogic Portal provides the following reference domains, each of which includes one or more enterprise applications:

- `petflowDomain`
- `p13nDomain`
- `wlcsDomain`
- `portalDomain`

petflowDomain

petflowDomain contains the petflowApp application, which features a demonstration of the Webflow service. To activate the domain and server, do the following:

1. Change to the `PORTAL_HOME/config/petflowDomain` directory.
2. Run `startPetflow.bat` (`startPetflow.sh` on UNIX).

Then, to view petflowApp, in a Web browser enter the following URL:

`http://localhost:7501/petflow`

p13nDomain

p13nDomain contains the p13nApp application, which demonstrates personalization services. To activate the domain and server, do the following:

1. Change to the `PORTAL_HOME/config/p13nDomain` directory.
2. Run `startP13N.bat` (`startP13N.sh` on UNIX).

Then, to view p13nApp from the p13nDomain, in a Web browser enter the following URL:

`http://localhost:7501/`

Note that this URL only specifies the server name and port number because the p13n Web application within p13nApp has been configured as the default Web application of the p13nDomain.

Any domain can specify a default Web application, which responds to HTTP requests that cannot be resolved to another deployed Web application. For information on how to specify a default Web application, refer to “Configuring WebLogic Server Web Components” in the *WebLogic Server Administration Guide*.

wlcsDomain

wlcsDomain contains both the p13nApp and wlcsApp applications. The p13nApp is the same application that you can access from p13nDomain. WebLogic Portal includes it in wlcsDomain to demonstrate that you can deploy multiple applications in a domain, and as a convenience to you: if you know that you want to use personalization and commerce services on your site, you can start wlcsDomain to access both p13nApp and wlcsApp.

wlcsApp features a demonstration of personalization, commerce, and campaign services.

To activate the domain and server, do the following:

1. Change to the `PORTAL_HOME/config/wlcsDomain` directory.
2. Run `startWLCS.bat` (`startWLCS.sh` on UNIX).

To view wlcsApp, in a Web browser enter the following URL:

`http://localhost:7501/wlcs`

Then, to view p13nApp, in a Web browser enter the following URL:

`http://localhost:7501/p13n`

Note that the p13n Web application is not the default Web application for wlcsDomain and therefore the URL must specify the context root of the p13n Web application. To view the default application for wlcsDomain, enter the following URL:

`http://localhost:7501/`

portalDomain

portalDomain includes p13nApp, wlcsApp, and portal. The portal application demonstrates the portal service.

Because portalDomain includes three applications, it provides the best demonstration of the scope of WebLogic Portal features. This document assumes that in your development environment, you use portalDomain as your reference, though you can use any domain that best reflects your business needs.

Each application that you add to a domain increases the amount of time to start the server.

To activate the portalDomain and server, do any of the following:

- From `PORTAL_HOME/config/portalDomain`, run `startPortal.bat` (`startPortal.sh` on UNIX).
- From `PORTAL_HOME`, run `StartPortal.bat` (`StartPortal.sh` on UNIX).
`StartPortal.bat` calls `config/portalDomain/startPortal.bat`. We provide this script as a convenience to you.
- On Windows, click Start → Programs → BEA WebLogic E-Business Platform → BEA WebLogic Portal 4.0 → Start BEA WebLogic Portal

Then, to view `p13nApp`, in a Web browser enter the following URL:

`http://localhost:7501/p13n`

To view `wlcsApp`, in a Web browser enter the following URL:

`http://localhost:7501/wlcs`

To view `portalApp`, in a Web browser enter the following URL:

`http://localhost:7501/stockportal`

Non-Archived Enterprise Applications

The J2EE specifies that enterprise applications can be archived into a single file (EAR file), which simplifies the process of deployment. However, to facilitate your development process, we do not archive the reference enterprise applications.

We recommend that you do not archive your applications until you are ready to deploy them in your production environment.

Portals and Portlets

The type of portal provided by the WebLogic Portal platform has the following characteristics:

- Provides a layout paradigm for dynamic, interactive, personalized content
- Blocks of content called portlets laid out in rectangular grid
- Portal resources can be reused among multiple group portals
- Portlets can be personalized at many levels
- Layout of portlets can be configured by Visitor role
- Multiple page views can be defined for the same group portal
- Color schemes called Skins can be designated and customized
- Delegated Administration enables multiple administrative roles

The following subsections introduce portals on the WebLogic Portal platform:

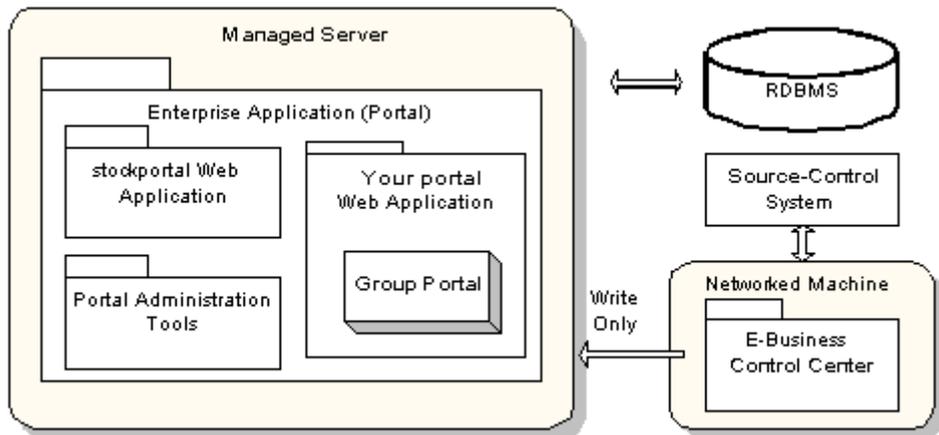
- Relationships Among Applications and Portals
- Portal Concepts
- Designing the Appearance of Portlets
- Personalization versus Customization
- Visitor Entitlements
- Delegated Administration

For more information, refer to the *Getting Started with Portals and Portlets* and the *Guide to Using the E-Business Control Center*.

Relationships Among Applications and Portals

The application hierarchy in WebLogic Portal can be described starting at the domain level on an application server, down to enterprise applications, which in turn contain Web applications, one of which is called portal. This application constitutes a single instance of portal, and it can contain multiple Group Portals. Each Group Portal can include multiple pages, also known as portal page views. Each of these pages consist of a configurable array of discrete content modules called portlets.

Figure 1-3 Portal Application Overview



Portal Concepts

This section introduces the following portal concepts:

- What Is a Group Portal?
- What is a Portal Page?
- What Is a Layout?
- What Is a Portlet?

What Is a Group Portal?

A group portal is a collection of resources that are intended to be used by a specific group of visitors. Any given portal Web application can support multiple group portals, each of which is an instantiation of a larger, single portal.

For example, you design a portal that provides information on the civic facilities in Los Angeles county. You want different groups of visitors to have access to different resources within the portal, so you design one group portal for the chamber of commerce, another group portal for the tourism industry, and another group portal for residents of the county. Within the portal, you maintain one collection of resources, but you provide different groups access to different sets of resources.

What is a Portal Page?

Within a group portal, multiple tabbed pages, also referred to as portal page views, can be created and customized, enabling you to make optimal use of precious layout space and allowing for quick and efficient navigation within a portal.

A portal page is a single view of an array of content modules which can be configured by users at many different levels. Changes made and saved during a session will be evident the next time the visitor logs on.

What Is a Layout?

Layouts are configurations of portlets within a portal page. A set of portlets can be arranged on a given layout in one page, and reused or reshuffled on the same or a different layout for a different page. The WebLogic Portal Administration Tools are used to select the default layout, which will be used if no other selection is made. Each visitor can create a custom layout.

What Is a Portlet?

Architecturally speaking, a portlet is a JSP with some XML-based metadata that fits into a Portal. Strategically, a portlet is a highly focused channel of information directed to a specific user or group using a portal as its container. Within a portlet, certain layout elements are available: the header, subject and footer sections of a portlet appear in a portal page. The contents of the title bar are determined by the framework, based on customization and evaluated at runtime.

Each of these objects can be designated as a separate JSP page, and the portlet can further be assigned the following characteristics:

- Editable - user can edit and save the content of the body page.
- Floatable - portlet can be displayed outside the portal container in a separate browser window.
- Minimizable - user can minimize the portlet.
- Helpable - user can open online help from within the portlet.

Individual portlets are arranged in portals using a construct called a Layout. Portlets can be manipulated using directional arrows.

Multiple portlets can be assigned to one or more portals, their inclusion based on a flexible set of determiners. Application functionality can be implemented within these highly focused content channels: For example, an employee portal could elicit immediate employee feedback to some general corporate communication, sub-groups could be given access to departmental applications which have been “portalized,” and each individual employee could be given secure access to personal benefits data within a custom portlet.

Note: For an explanation of how roles are defined within this document set, consult the “Documentation Roadmap for WebLogic Portal” topic in the *Strategies for Developing E-Business Web Sites* documentation.

Importantly, much of this content personalization can be accomplished with relatively little coding on the part of the Business Engineer (BE) user.

Designing the Appearance of Portlets

Skins define the look and feel of a portlet, including portlet control images (buttons used to invoke edit or minimize portlet views), default font face and size, and color attributes for parts of the portlet. Skins are made up of CSS (Cascading Style Sheet) files and directories of images.

Personalization versus Customization

In the WebLogic Portal platform, personalization means varying the content and functionality presented to an authenticated visitor, that is, to an end user successfully logged into a portal. Once logged in, a visitor may customize various aspects of the portal - provided these controls have been made available by the administrator who set up the portal.

The WebLogic Portal platform enables personalization at several different levels: First of all, when a visitor logs in, the visitor's membership in one or more groups is evaluated. After group membership, the next factor considered is called entitlements - meaning access to portal resources based on dynamic evaluation of attributes. Finally, if the visitor has performed any customization of the portal, these customizations are applied to the portal resources to which the visitor has access, and the page is rendered accordingly.

Visitor Entitlements

The visitor entitlement feature is a system for granting permissions to view or modify the pages or portlets that are controlled by a portal.

Visitor entitlements are managed within the WebLogic Portal Administration Tools, and make use of Entitlement Segments which are created in the E-Business Control Center. Basically, for each portal resource, you can define the access and the degree to which a user may change the behavior of a portlet.

For example, you could give members of one Entitlement Segment the ability to edit the layout of a certain portlet, while denying the same ability to members of another group.

Visitor entitlements in the WebLogic Portal platform go far beyond this simple example. The rules you can create in the E-Business Control Center can involve user attributes stored within the user properties set, or they can be based on system parameters, dates, and so on.

Delegated Administration

Maintaining large Web applications requires the ability to parse out administrative tasks into discrete tasks. For instance, if you needed to manage the content of your portal applications at a centralized corporate office while delegating localization and design to regional offices, an elegant solution would be to create some administrative user accounts and then assign different sets of permissions to each user based on the tasks you needed to hand out. WebLogic Portal now includes advanced Delegated Administration functionality to enable the creation of administrator roles with fine-grained administrative privileges.

Flow Management with Webflow

Webflow is a mechanism designed to maintain the much-desired separation between presentation logic and underlying business processes. It is a set of expressions, formatted in XML, that determine how a Web site responds to visitor actions.

Webflow can support a complex set of responses to visitor interactions. For example, when a customer click an Add To Cart button, Webflow can invoke a string (pipeline) of components that calculate cost, associate data with the customers session, and display a confirmation to the customer that the item was add to the shopping cart.

This section contains the following subsections:

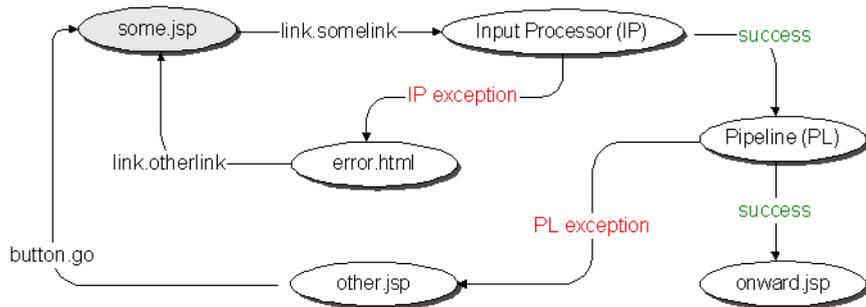
- Webflow as a State Machine
- Namespaces
- The Scope of Webflow Components
- Creating, Modifying, and Synchronizing Webflow Components
- For More Information

Webflow as a State Machine

As shown in Figure 1-4, Webflow can be thought of as a state machine. A Webflow begins at a node that handles presentation, such as a JavaServer Page (JSP). When a visitor interacts with a Web site that uses Webflow, their interaction causes information about the origin of the interaction and the event to be sent to the Webflow. Based on this information, the Webflow mechanism makes a decision about the next node it should transition to. If the Webflow decides to invoke a type of processor node (that is, an Input Processor or Pipeline) as a destination, the Webflow may temporarily store state information. Input Processors and Pipelines are Webflow components that process and store data. For example, Input Processors can validate HTML form data and provide conditional branching within a Web page and Pipelines can persist a visitor's information to a database. Pipelines are aggregations of Pipeline Components, which are discrete units of business logic.

Using Figure 1-4 as an example, a visitor may be viewing a page called `some.jsp` that exists in your Web application. The visitor then clicks a link called `someLink`, which is available to them on `some.jsp`. In this case, the origin of the interaction is the current page (`some.jsp`) and the event is clicking on the link named `someLink`. Based on this information, the Webflow may decide to transition to an Input Processor.

Figure 1-4 Webflow as a State Machine



Namespaces

You can use namespaces to divide a Webflow into a number of smaller, more manageable modules. This modularity may make your development team more productive by allowing individual developers to work simultaneously with various portions of a Web application, without having to worry about naming collisions. For example, within a given portal, each portlet can use a separate Webflow namespace.

You can have any number of namespaces within a Webflow, but namespaces can only be one level deep.

The Scope of Webflow Components

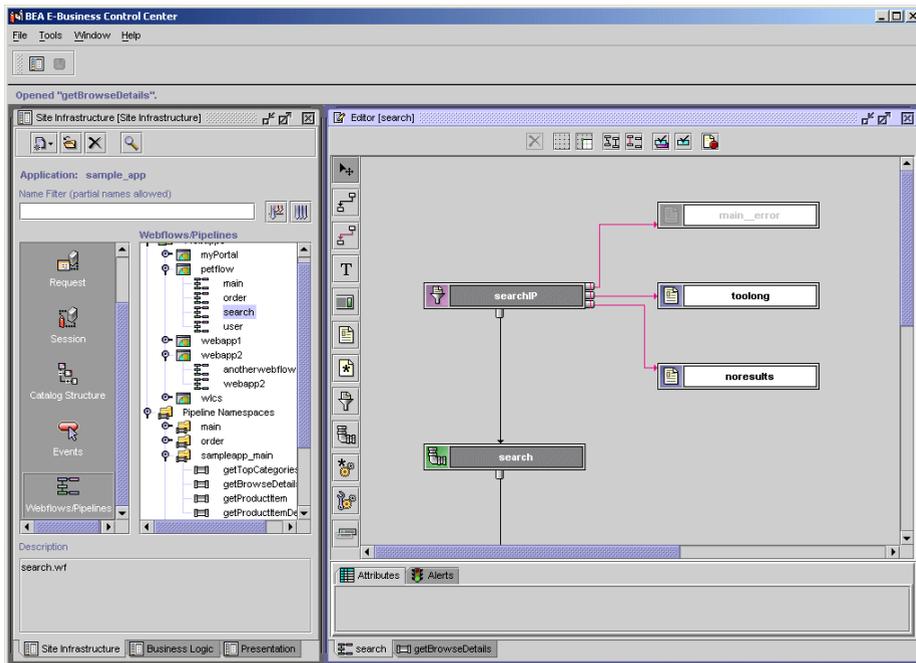
A Webflow is scoped to a Web application. In other words, you can only have one Webflow (with potentially many namespaces) per Web application.

Pipelines are scoped to an enterprise application, meaning that multiple Webflows (which are within Web applications) may all invoke the same Pipeline.

Creating, Modifying, and Synchronizing Webflow Components

The Webflow Editor in the E-Business Control Center is designed to help you create, modify, validate, and synchronize Webflow components. (See Figure 1-5.)

Figure 1-5 WebFlow Editor in the E-Business Control Center



The E-Business Control Center saves Webflow data as XML files on the local host. When you are ready to make a Webflow available to a Web application, you use the E-Business Control Center to synchronize the XML Webflow data with the Web application on an active server.

For more information on synchronizing data, refer to “Synchronizing Application Data” in the *Deployment Guide*.

For More Information

Refer to the following:

- For information on the E-Business Control Center, refer to “E-Business Control Center” on page 1-62.
- For more details on Webflow and Pipelines, refer to the *Guide to Managing Presentation and Business Logic: Using Webflow and Pipeline*.

User and Group Management

BEA WebLogic Portal applications may store personal information about customers or display Web site content based on a customer’s identity. Personal information about customers is stored the RDBMS repository and organized by a customer profile.

A customer profile is a schema that determines which data you collect and store about a customer. At a minimum, a customer profile must contain a user ID and password. To provide a personalized experience, the profile must store other information that is pertinent to your business needs. For example, if you store information about the types of mutual funds in which your customer invests (conservative, moderate, aggressive), you can present content, advertisements, and additional fund recommendations that reflect each customer’s preference.

Each customer profile must belong to a single group profile. You can use hierarchies of group profiles to inherit property values. To manage customer and group profiles, WebLogic Portal provides the following services:

- User Registration, which is a set of JSP, Webflow, and Pipeline templates that you use to collect and store information about new customers. User Registration stores information in the RDBMS repository. Figure 1-6 shows a User-Registration JSP template for customers to enter their registration information or begin the registration process.

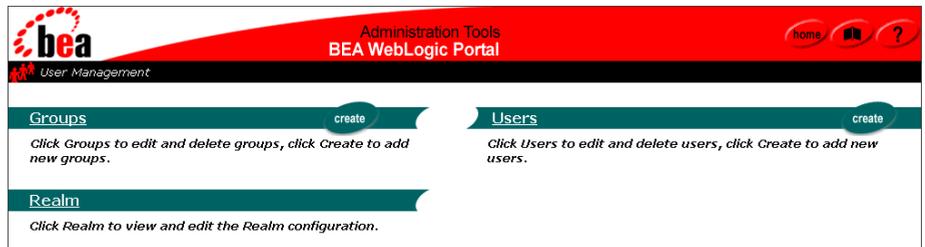
Figure 1-6 login.jsp



- User Management, which you use to manage existing customer data. The service includes support for the Unified User Profile (UUP), which creates a customer profile by combining properties from an existing data source with properties in the RDBMS repository. For example, User Management provides a default customer profile that can combine customer properties from the RDBMS repository with properties from an LDAP server into a single-user profile (customer profile). Developers do not have to worry about the different underlying data sources; they just need to go to customer profiles for user information.

You can manage user and groups using the WebLogic Portal Administration Tools and the E-Business Control Center. Figure 1-7 shows the User Management screen in the WebLogic Portal Administration Tools.

Figure 1-7 WebLogic Portal Administration Tools—User Management Screen



Services for Registering Customers and Managing Customers

The Registering Customers and Managing Customer services support creating a customer profile (registration), storing profile information, and determining the activities that customers can perform after they register. These services store their data in the RDBMS data repository.

The services consist of JSP templates and the business logic associated with them. You can modify the JSP templates, input processors, and Pipeline components, or create new ones to fit your specific needs. Figure 1-8 shows a JSP template for creating customer profiles.

Figure 1-8 newuser.jsp Template

The screenshot displays a web browser window with a navigation bar at the top. The main content area is titled "Create a New Account" and contains a registration form. The form includes the following fields and options:

- First name**: Text input field.
- Middle initial**: Text input field.
- Last name**: Text input field.
- Address**: Text input field.
- Address 2**: Text input field.
- City**: Text input field.
- State / Province**: Dropdown menu with "SELECT" and a plus sign. A note indicates "(Required for U.S. and Canadian address)".
- Zip/Postal Code**: Text input field.
- Country**: Dropdown menu with "SELECT" and a plus sign.
- Home phone**: Text input field.
- Business phone**: Text input field.
- Email address**: Text input field.
- Demographic Options**:
 - Gender**: Radio buttons for "Female" and "Male".
 - Date of Birth**: Text input field with a placeholder "(mm/dd/yyyy)".
 - Occupation**: Radio buttons for "Clerical", "Executive Management", "Professional", "Engineering", "Management", and "Sales".
- Opt-in**: A checked checkbox with the text "Yes, I want to be offered specials and notified of discounts. Please send me promotional e-mail."

On the left side of the form, there is a promotional banner: "Register Now and save \$10 on your order of at least \$50 placed today!!". Below this is a note: "Catalog data provided courtesy of The Catalog, where demand...". At the top right, a red banner reads "BEA/WebLogic Commerce Server Commerce Templates".

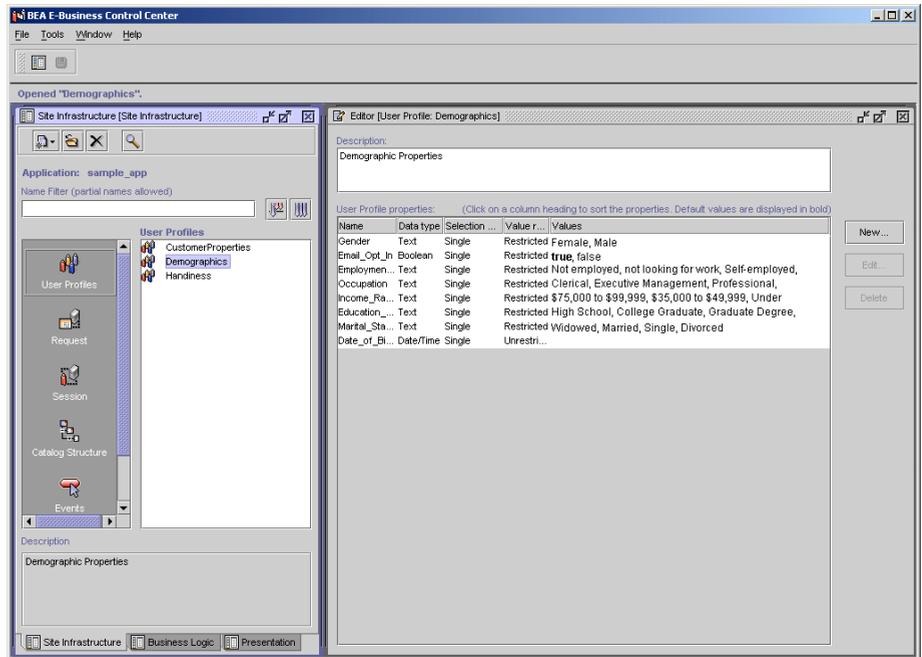
Note: For more information, refer to the *Guide to Registering Customers and Managing Customer Services*.

Customer and Group Profiles

Customer profiles can contain any information that is relevant to the context of your application. Things like age, gender, and favorite genres of music could all be stored in the profile. Things like department, position, and office location could also be in a profile. Each item in the profile, such as age, is called a customer property. The E-Business Control Center allows you to define and edit customer profiles. Figure 1-9 shows the User Profiles dialog box in the E-Business Control Center.

Note: Customer profiles are called User Profiles in E-Business Control Center.

Figure 1-9 User Profiles in the E-Business Control Center



Note: For more information about using the E-Business Control Center to define and edit customer profiles, see the online help in the E-Business Control Center or “Creating and Managing Property Sets” in the *Guide to Building Personalized Applications*.

Group hierarchies permit property inheritance. For example, portals use properties to determine such characteristics as the color schemes for portlets. You can make these color properties part of a group profile. Then, all customers within a specific group see the same color scheme for their portlets.

Groups can belong to parent groups, from which they can inherit properties. Groups may have only one or no parent group

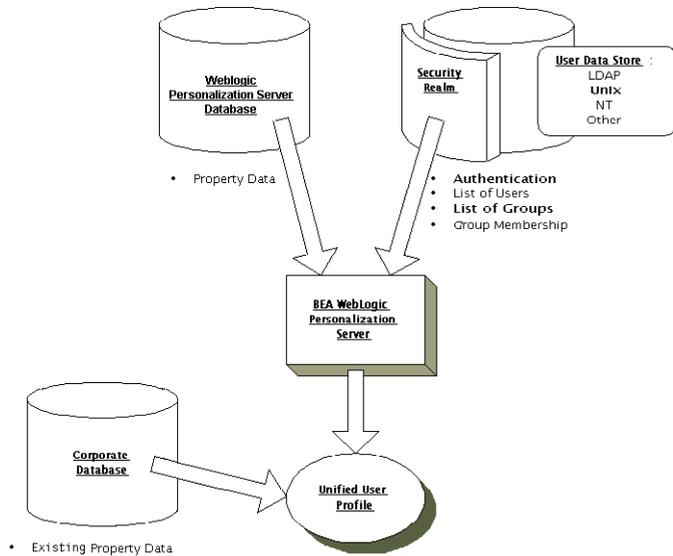
Unified User Profile

The Unified User Profile (UUP) combines properties from an existing data source with properties that you store in the WebLogic Portal RDBMS repository. For example, the customer profile that ships with WebLogic Portal can combine user properties from the RDBMS data repository with user properties from an LDAP server into a single profile for use within an application. Developers and system users need not worry about the different underlying data sources. To them there is just one place to go for user information – the customer profile.

1 Architectural Overview

Figure 1-10 provides an example of the using Unified User Profile in an environment in which users and groups exist in some type of data store already, such as an LDAP directory. Existing user property data must be incorporated into the Unified User Profile.

Figure 1-10 UUP Example



Security

The following mechanisms protect the confidentiality and integrity of customer and group profile data, customer preferences, and the overall integrity of customer transactions:

- Security Realms
- Security Roles and Declarative Security

Security Realms

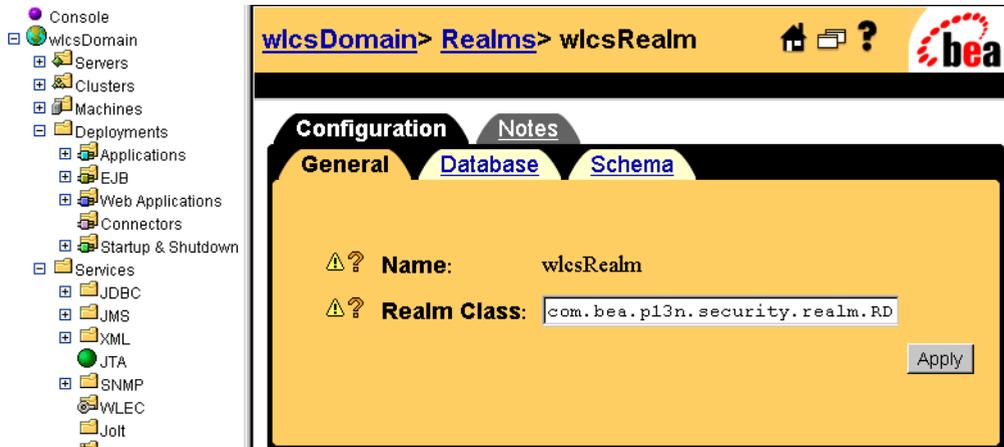
A *security realm* determines how a user is authenticated and retrieves access control lists for given names. WebLogic Server supports several types of security realms that support different environments and security needs. For a list of the types of security realms that WebLogic Server supports, refer to “Security Fundamentals” in the *WebLogic Server Programming WebLogic Security* guide.

By default, WebLogic Portal uses a realm called `wlcsRealm` to authenticate users and verify group identity. This realm, which is an instance of `com.bea.p13n.security.realm.RDBMSRealm`, stores user IDs and passwords in the WebLogic Portal RDMBS repository. (This is the same RDMBS repository that stores customer profile data.)

If you use a different security realm, the Unified User Profile (UUP) keeps the customer profile data in the RDBMS repository synchronized with the data that is in your alternate security realm.

Figure 1-11 shows the `wlcsRealm` in the WebLogic Server Administration Console.

Figure 1-11 wlsRealm in the WebLogic Server Administration Console



Security Roles and Declarative Security

While the security realm authenticates that a user belongs to a group, security roles offer one way of managing access to resources within Web applications and EJBs. The deployment descriptor for each Web application and EJB does the following:

- Maps groups from the security realm (principals) to a J2EE security role.
- Declares a set of protected resources and indicates which security role can access the protected resources.

With declarative security, roles and the resources that they can access are defined in deployment descriptors. Access to protected resources is granted based on whether the requesting user's security role matches the role declared in the deployment descriptor for that resource.

In addition to declarative security, WebLogic Portal provides the entitlements feature, which is a system for granting permissions to view or modify the pages or portlets that are controlled by a portal. For more information about entitlements, refer to "Visitor Entitlements" on page 1-17.

For more information, refer to the following:

- For information about declaring roles for an enterprise application, refer to "Assembling and Deploying Enterprise Applications" in the *Deployment Guide*.

- For information about mapping groups in the security realm to roles in an application, refer to “Assembling Your Web Application” in the *Deployment Guide*.
- For information about declaring secured resources in a Web application, refer to “Declarations of Secure JSPs” under “Assembling Your Web Application” in the *Deployment Guide*.

Property Sets

A property set is a convenient way to give a name to a group of properties for a specific purpose. For example, in the sample application you will find a User Profile property set named Customer Properties. This property set defines around thirty properties for an e-commerce customer, such as First Name, Last Name, Home Phone, Email, and Customer Type.

Property sets and property definitions are created in the E-Business Control Center, on the Site Infrastructure tab. A Property Editor allows you to give a new a property a name and a description, assign a data type, a selection mode, and a value range, and create a list of possible values for the property.

Although properties are designed in the E-Business Control Center, the value assigned to a property is created in the application. Properties are generally represented in an application as fields on a page. To use the Customer Properties property set to collect information about an e-customer, the application will typically present the thirty properties in a list or a table, with text boxes provided to fill in specific values for the customer.

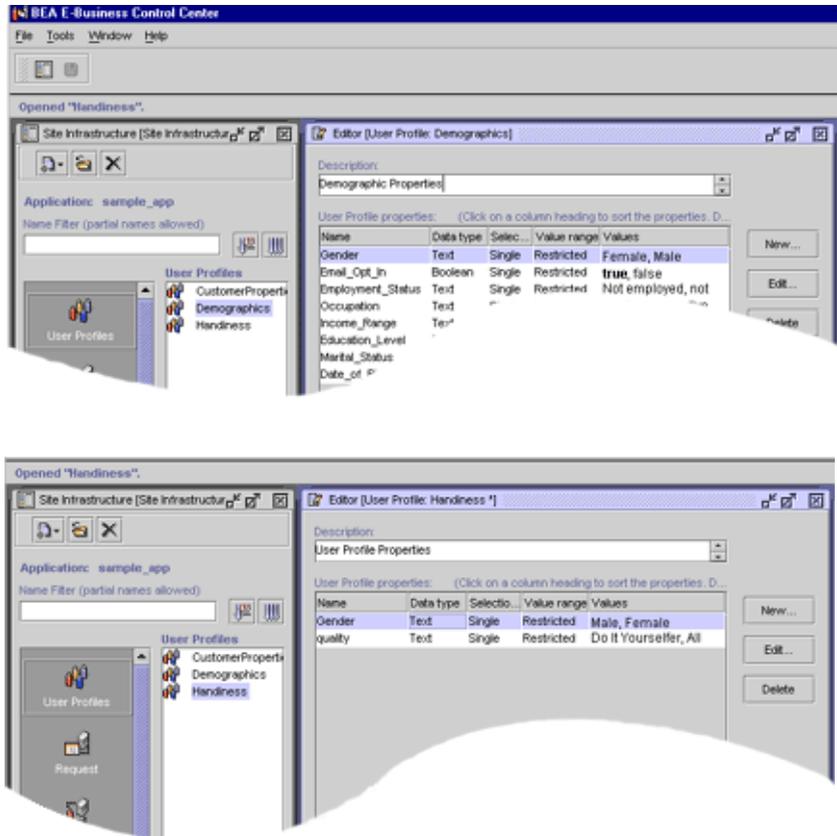
Property Sets Serve as Namespaces for Properties

In the most general sense, a property can be considered a name/value pair. Property sets serve as namespaces for properties so that properties can be conveniently grouped and so that multiple properties with the same name can be defined.

1 Architectural Overview

For example, you might create a property set called Demographics to describe user profile properties. The Demographics property set contains properties called Age, Gender, Income, and so forth. Because property sets create unique namespaces for properties, another property set can also have a property called Gender, and the two values will be kept separate.

Figure 1-12 Property Sets Serving as Namespaces



Where Property Sets Are Used

For Portal and Personalization Server purposes, property sets are applied to six major areas. Of these, five are configured in the E-Business Control Center.

Table 1-2

1. User Profiles



The User Profiles property set type is used for defining the property sets and properties that apply to user and group profiles. For example, a property set of this type might be created called CustomerProperties. Subsequent property retrieval for a particular user or group can then be scoped with this property set name to retrieve the user's email address.

2. HTTP Requests



The Request property set type is used for defining the property sets and properties that apply to HTTP requests. A Request property set type might be called "portalA." Properties available through this property set can then be accessed via the Advisor.

We provide this property set so you can write rules that use the data and methods of the HTTP `Request` object.

3. HTTP Sessions



The Session property set type is used for defining the property sets and properties that apply to HTTP sessions. Like the User Profiles property set type, a "Session" property set type might be called "portalA." Properties available through this property set can then be accessed via the Advisor.

We provide this property set so you can write rules that use the data and methods of the HTTP `Session` object.

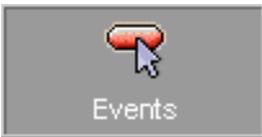
Table 1-2

4. Catalog Structure



The Catalog Structure property set type is used to define custom attributes for a product item in the Commerce services catalog. These custom attribute can be used in addition to the default attributes provided by Commerce services in the catalog database tables.

5. Events



The Events property set type is used to register a custom behavior-tracking event. For example, add properties of this type to track a user behavior that is in addition to the standard behavior tracking events.

For the purpose of registering an event, you can consider an event property as a name-value pair. During the registration of a custom event, you specify the event's name, description, and one or more properties. Each property has a range, type of permissible value, and default value. The information you need to register an event should be available from your Business Engineer (BE) or Java developer.

Do not modify properties for any of the standard behavior-tracking events.

6. Content Management

Use the tools associated with your document management system.

The Content Management property set type is used for defining the configuration and run-time use of the content management system. Content Management property sets cannot be created or manipulated with the Personalization Server Administration Tools or the E-Business Control Center. Instead, use the tools associated with your document management system.

For more information about property sets, refer to “Creating and Managing Property Sets” in the *Guide to Building Personalized Applications*.

Rules

At the core of many WebLogic Portal features is the Rules service, which compares logical statements with run-time data to provide things such as personalized content.

You interact with the Rules service by creating customer segments, which are described in this section, and content selectors, ad placeholders, and campaign scenarios, which are described in other sections of this topic.

Customer Segments

A customer segment is a set of criteria for dynamically categorizing customers for online marketing purposes. It can group customers based on common characteristics such as product preferences or demographics. It can gather data from the customer's current session, profile, or other data available to the server (such as the current date).

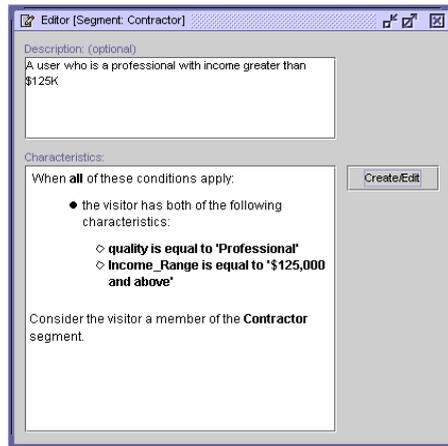
For example, if your site sells or advertises investment opportunities, you can define customer segments for conservative or aggressive investment portfolios. When a customer logs in to your site or portal, JSP tags trigger the Rules service to determine whether the customer's profile contains a preference for investment types and whether the customer has bought or sold investments during the current session. Based on this data, the Rules service determines whether the customer belongs to a conservative-investment segment or an aggressive-investment segment.

You can use customer segments in conjunction with features in the WebLogic Portal to personalize a customer's experience on your Web site. For example, campaign scenarios, campaign discounts, and personalized content selection use customer segments to determine the appropriate actions for each customer.

1 Architectural Overview

You use the E-Business Control Center to define customer segments. (See Figure 1-13.)

Figure 1-13 Customer Segment Screen



Note these important distinctions between customer and group profiles and customer segments:

- A **customer profile** is a collection of data about a customer. They are unique to each customer. The server authenticates customers based on the information in their profile.
- A **group profile** is a static, hierarchical organization of customer profiles. Each customer profile must belong to a group. For example, portals use properties to determine such characteristics as the color schemes for portlets. You can make these color properties part of a group profile. Then, all customers within a specific group see the same color scheme for their portlets. Groups can belong to parent groups, from which they can inherit properties.

The server grants access privileges to customers based on their association with a group.

You create customer and group profiles in the WebLogic Portal Administration Tools. These profiles can retrieve data that you store in the RDBMS or in other data systems, such as an LDAP server.

- A **customer segment** is a set of criteria for dynamically categorizing customers. It can categorize customers based on information from the customer profile, but it can also use other sources, such as the customer's HTTP session or request objects, to determine whether a customer currently fits into a segment. Customers can belong to multiple segments, and their membership can change during a single session. Individual JSP tags and services invoke the Rules service to determine whether a customer fits into a segment.

Content Management

Some of the information on your site, such as small static pieces of text, can easily be maintained within the JSPs that you use for your presentation layer. However, we recommend that you use a content management system to store and maintain the bulk of the information on your site, especially dynamic content such as banner ads and informational documents. In addition to storing content (HTML or text fragments, images, and other file types), content management systems enable you to describe the

documents in your system by attaching attributes to each document. You can then use queries to search and retrieve documents based either on a set of criteria or by a unique ID.

This section discusses the following:

- Content Management Systems
- Content Selectors
- Ad Placeholders
- Comparison of Methods for Retrieving and Displaying Documents
- The Advisor

Content Management Systems

To use content selectors, ad placeholders, and JSP tags that retrieve and display content, you must store documents in a content management system.

WebLogic Portal provides a content management system, the Content Manager, and a program for loading documents into the system, `BulkLoader`. Content Manager and `BulkLoader` are appropriate for sites with limited content personalization needs. The `BulkLoader` can parse a directory of HTML files and store their URL address and metadata attributes in the server's data repository. The `BulkLoader` automatically creates the schema for these attributes.

If your site contains larger amounts of content and you want more control over the publishing and tagging of content, WebLogic Portal provides integrations with third-party content management systems.

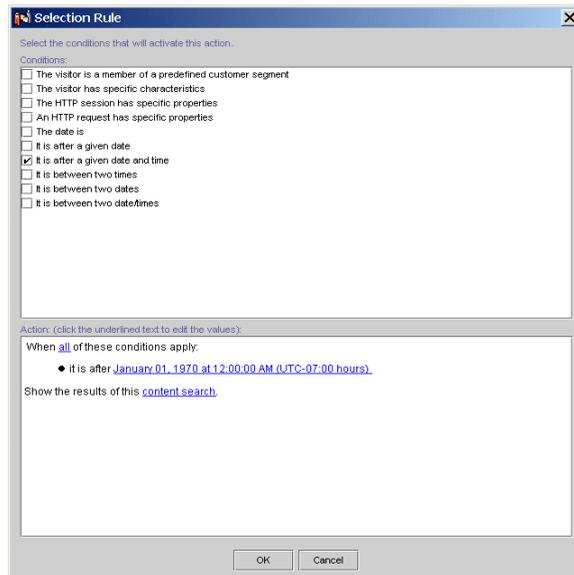
Content Selectors

Content selectors query the content management system for personalized content. They consist of the following elements:

- A set of conditions that determine when the content selector queries the content management system. BAs create and modify the set of conditions in the E-Business Control Center.
- A query that searches the content management system for one or more documents. BAs create and modify the query in the E-Business Control Center.
- A JSP tag that triggers the content selector to evaluate its conditions. The content selector JSP tag includes attributes that BEs can use to tune the performance of the content selection process. BEs create the JSP tags.
- A data object that WebLogic Portal creates to contain the results of the query. To display the documents that are in the object, a JSP developer must use additional JSP tags.

Figure 1-14 shows an E-Business Control Center window that is used for defining content selectors.

Figure 1-14 Window for Defining Content Selectors



The following set of tags uses a rule named `SpringSailing` (which a BA created in the E-Business Control Center) to retrieve and display documents:

```
<pz:contentSelector rule="SpringSailing"
contentHome="<%=ContentHelper.DEF_DOCUMENT_MANAGER_HOME %>"
id="textDocs" />
<es:forEachInArray array="<%=textDocs%>" id="aTextDoc"
type="Content">

  <P><cm:printDoc id="aTextDoc" /></P>

</es:forEachInArray>
```

Ad Placeholders

Ad placeholders use a set of rules to select one query from a pool of queries. Then they run the query, retrieve a document from the content management system, and generate the HTML to display the query results. Ad placeholders consist of the following elements:

- A set of default queries and a pool of additional queries that campaign scenarios generate. BAs create and modify ad placeholders and queries in the E-Business Control Center.
- A JSP tag that runs the query and displays the results. BEs create the JSP tags.

Ad placeholders recognize and generate HTML for specific types of content (HTML, images, multimedia files), based on MIME types. You can create Java classes that generate HTML for additional MIME types.

For example, you want to display ads in the top region of your web site's home page. You define an ad placeholder and create ad queries for the placeholder. Then you add an ad placeholder JSP tag near the top of the home page. When a customer requests the home page, the placeholder selects a query, runs the query, and displays the results.

The following JSP tag runs any queries associated with the `top_banner` ad placeholder, and then selects and displays a single ad:

```
<ph:placeholder name="top_banner" />
```

Comparison of Methods for Retrieving and Displaying Documents

WebLogic Personalization Server provides several methods for retrieving documents from a content management system and displaying them on your Web site.

The following table compares the methods of content retrieval that WebLogic Personalization Server provides.

Table 1-3 Methods for Retrieving and Displaying Documents

Use This Method...	When You Want To...
Content selectors and <code><pz:contentSelector></code> tags	<ul style="list-style-type: none">■ Use a centrally maintained infrastructure for matching Web site content with customer profiles, or customer segments. BEs develop the infrastructure, then BAs use the E-Business Control Center to define and modify conditions under which content selectors query the content management system for documents.■ Retrieve any type of content that your content management system contains (and that a browser supports).■ Display each document that a content-management query returns. Content selectors store the results of a query in an array. You can use other JSP tags to display some or all of the documents that are in the array.■ Place the results of the query in a cache. <p>Content selectors require you to determine the MIME-type of the documents and to supply the appropriate HTML that the browser requires to display them. For more information about content selectors, refer to “Content Selectors” on page 1-39.</p>
<code><pz:contentQuery></code> tag	<ul style="list-style-type: none">■ Run a static, narrowly-defined query to display a document only in a specific JSP. <p>You must modify each occurrence of this tag if you want to modify its query. If you want this tag to display contents for specific customers or in response to an event, you must surround it with additional tags that evaluate the display condition.</p>

Table 1-3 Methods for Retrieving and Displaying Documents (Continued)

Use This Method...	When You Want To...
<p>Ad placeholders and <code><ph:placeholder></code> tags</p>	<ul style="list-style-type: none"> ■ Use a centrally maintained infrastructure for matching advertising documents with events, customer profiles, or customer segments. BEs develop the infrastructure, then BAs use the E-Business Control Center to define and modify the queries that each placeholder can run. ■ Run queries as part of a scenario action in a campaign (available only with Campaign services). ■ Use a single infrastructure to support multiple, concurrent advertising agenda. Ad placeholders use an Ad Conflict Resolver to select a single query if multiple agenda request to run multiple queries in the same location at the same time. ■ Automatically generate the HTML that the browser requires to display the query results. <p>Without customization, ad placeholders support only HTML, image, and Shockwave documents. For more information about ad placeholders, refer to “Ad Placeholders” on page 1-41.</p>
<p><code><ad:adTarget></code> tag</p>	<ul style="list-style-type: none"> ■ Make sure that a specific ad query runs in a specific location. ■ Automatically generate the HTML that the browser requires to display the query results. <p>The <code><ad:adTarget></code> tag is not part of the infrastructure for supporting multiple advertising agenda. It cannot run a query as part of a scenario action. You must modify each occurrence of this tag if you want to modify its query. If you want this tag to display contents for specific customers or in response to an event, you must surround it with additional tags that evaluate the display condition.</p> <p>Without customization, the <code><ad:adTarget></code> tag supports only HTML, image, and Shockwave documents.</p>
<p><code><cm:printDoc></code> tag</p>	<ul style="list-style-type: none"> ■ Use the content management system’s document ID to include non-personalized content in a HTML-based page. <p>The tag does not generate HTML to support the content it retrieves; it inserts the document into the JSP page exactly as it is stored in the content management system. BEs must modify each occurrence of this tag if you want to change the document that it retrieves.</p>

Table 1-3 Methods for Retrieving and Displaying Documents (Continued)

Use This Method...	When You Want To...
<code><cm:getProperty></code> tag	<ul style="list-style-type: none">Retrieves the value of the specified content metadata property into a variable specified by <code>resultId</code>. If <code>resultId</code> is not specified, the value will be inlined into the page, similar to the <code><cm:printProperty></code> tag. This tag operates on any <code>ConfigurableEntity</code>, not just the Content object. However, it does not support <code>ConfigurableEntity</code> successors.
<code><cm:printProperty></code> tag	<ul style="list-style-type: none">Display the value of a document attribute as a string. You can use this tag to display the value of any content object's attribute, not just document-type objects in a content management system.
<code><cm:select></code> tag	<ul style="list-style-type: none">Use a query to include non-personalized content in a HTML-based page.Place the results of the query in a cache. <p>The tag does not generate HTML to support the content it retrieves; it inserts the document into the JSP page exactly as it is stored in the content management system. BEs must modify each occurrence of this tag if you want to change the document that it retrieves.</p>
<code><cm:selectById></code> tag	<ul style="list-style-type: none">Use the content management system's document ID to include non-personalized content in a HTML-based page.Place the document in a cache. <p>The tag does not generate HTML to support the content it retrieves; it inserts the document into the JSP page exactly as it is stored in the content management system. BEs must modify each occurrence of this tag if you want to change the document that it retrieves.</p>

For more about using content management in WebLogic Portal, see “Creating and Managing Content” in the *Guide to Building Personalized Applications*.

The Advisor

The Advisor service provides advice. It is used by many of the personalization JSP tags to handle complex personalization requests such as classifying a user or to match users with content.

The Advisor service uses advislets to do the actual work of providing advice. Advislets are registered in the `advislet-registry.xml` file. This registry includes information that the Advisor uses to match advice requests to advislets.

Compound advislets create chains of advislets to perform more complex operations. In addition, the output of one advislet in a chain can be transformed into the correct input to the next advislet in the chain.

For more information, refer to the following:

- “Creating Personalized Applications with the Advisor” in the *Guide to Building Personalized Applications*
- WebLogic Portal *Javadoc*

Events and Behavior Tracking Service

Events and Behavior Tracking records how online customers interact with your e-business Web site. Specifically, you can use Behavior Tracking to record event information to a database where e-analytics and e-business systems can then analyze this information offline.

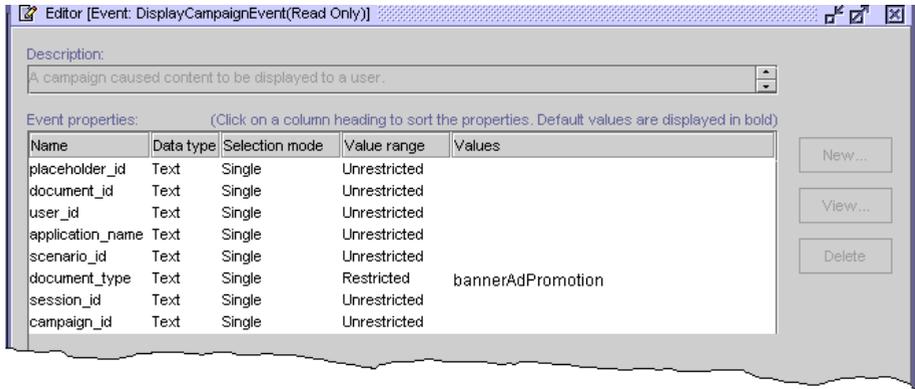
With advanced customer analytics that provide trend and predictive analyses, you can identify your most valuable customers and marketing opportunities. Each product provides a set of predefined events.

By default, WebLogic Portal tracks the following types of events:

- **Session:** The start time, end time, and if executed, the login time of the customer's session.
- **Registration:** The customer registers on the e-commerce site.
- **Product:** The customer is presented with a product or clicks (selects) the presented product.
- **Content:** The customer is presented some content, such as an ad, or clicks (selects) the presented content.
- **Cart:** An item is added, removed, or updated to the customer's shopping cart. Also when an entire order is purchased.
- **Buy:** The customer completes the purchase of one or more items.
- **Rules:** The rules that are fired as a customer navigates a web site.
- **Campaign:** The events generated within the context of a campaign.

Figure 1-15 shows Display Campaign event properties in the E-Business Control Center Events Editor.

Figure 1-15 Events Editor in the E-Business Control Center



In addition to the standard events, you can create a custom event for anything you want to track.

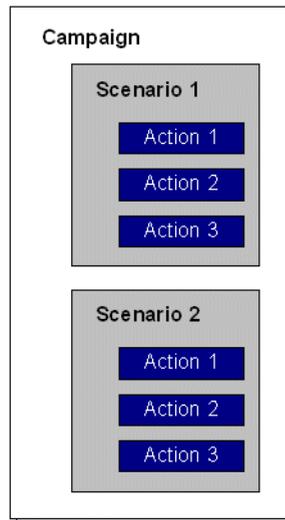
For more information about Events and Behavior Tracking, see the *Guide to Events and Behavior Tracking*.

Campaign Services

To attain specific business goals such as enticing latent customers to return to your site or employing cross-sell or up-sell tactics, you can use the E-Business Control Center to create and run a campaign.

A campaign is a named collection of scenarios that are focused on accomplishing a business goal. Scenarios, which are collections of scenario actions, can use ad placeholders to target advertisements, the Mail service to send mail to specific customers, and discounts to encourage or reward customers for purchasing items from your site. (See Figure 1-16.)

Figure 1-16 Campaigns, Scenarios, and Scenario Actions



The E-Business Control Center provides templates as well as a variety of conditions that you can use to trigger these actions. For example, Figure 1-17 shows a scenario that is used in conjunction with a discount.

Figure 1-17 Scenario Properties Screen

Scenario name:
Discount and Ad Actions

Scenario description:
This scenario displays a discount ad in conjunction with an associated discount action.

Customer segments: (optional)

Select Segments...

Tip: You can further subdivide the scenario's base Customer Segments (listed above) by specifying additional segments in the scenario's Actions.

OK Cancel

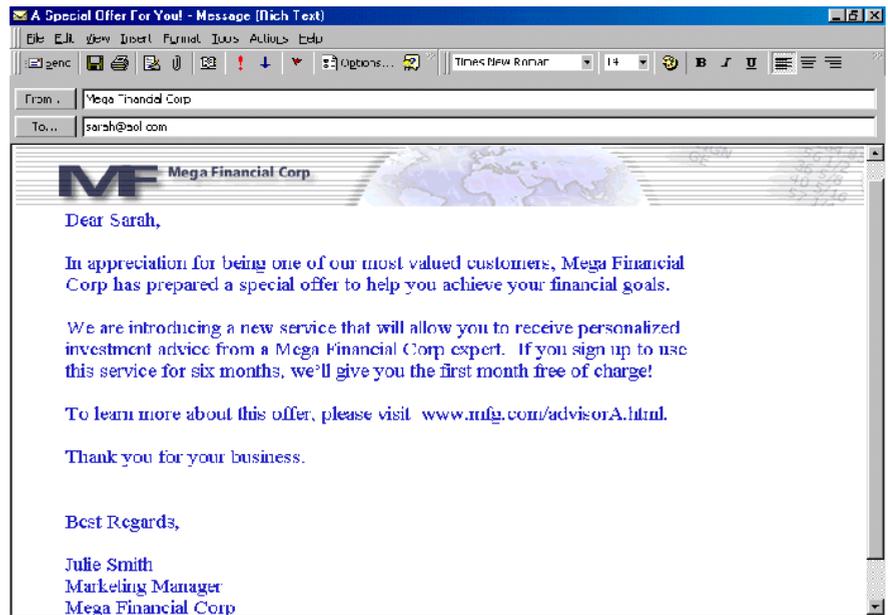
The E-Business Control Center saves all campaign data as XML files. When BAs are ready to make a campaign available to an application, they use the E-Business Control Center to synchronize the data with the application.

For information about how campaigns display ads, refer to “Ad Placeholders” on page 1-41. For information about how campaigns offer discounts, refer to “Discounts” on page 1-56. For information about how campaigns use the Email Service, refer to the following section.

Campaigns and the Mail Service

WebLogic Portal provides a basic mail service based on the JavaMail API. Campaigns can use this service to generate e-mail as part of a scenario action. For example, when a customer buys a television, a scenario can trigger an action to generate an e-mail that offers an extended warranty. An example of a promotional e-mail is shown in Figure 1-18.

Figure 1-18 Promotional E-Mail Example



WebLogic Portal organizes the generated email messages into batches, and a BE uses a send mail command to email the batches. You can use a third-party mail service to send and manage mail instead of the basic email service that WebLogic Portal provides. For more information about setting up and using the Mail Service for campaigns, refer to “Setting Up and Sending E-Mail for Campaigns” in the *Guide to Developing Campaign Infrastructure*.

Commerce Services

The following services enable your Web site for e-commerce:

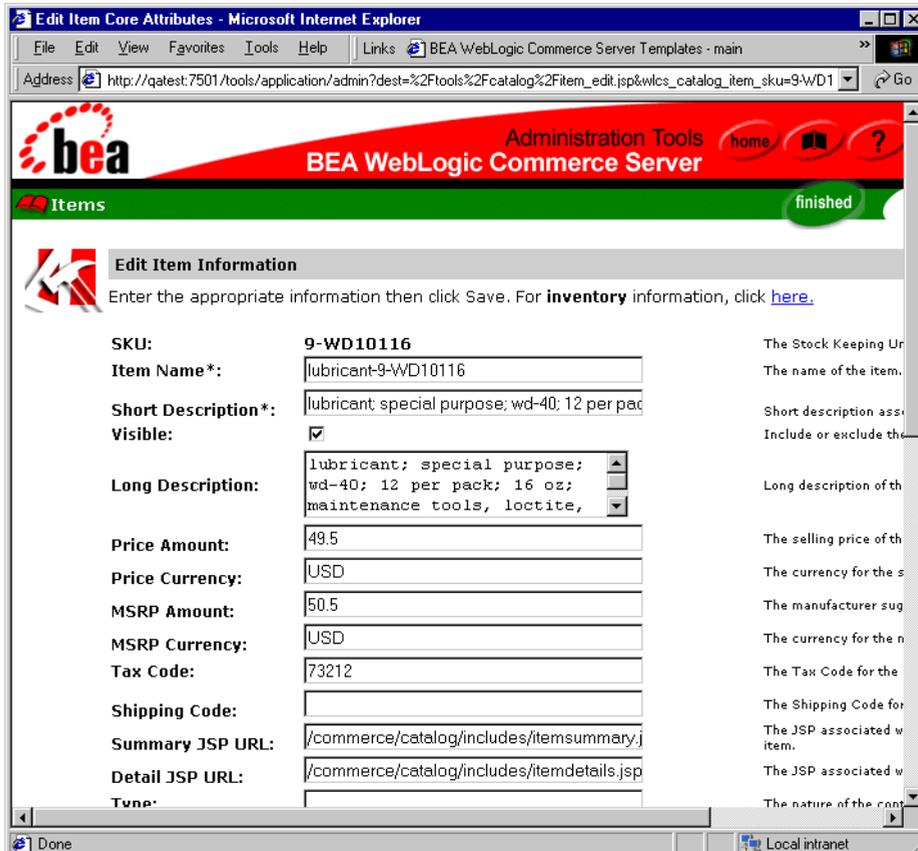
- Product Catalog
- Payment and Order Services
- Discounts

Product Catalog

A product catalog is essential to an e-commerce web site. It organizes the products that you offer and provides descriptions and pricing information. The WebLogic Portal Product Catalog provides the following features:

- **A well-designed database schema** and build scripts that define the commonly used product items and attributes found on Web-based catalog sites.
- **A bulk loader program called DBLoader** that takes a data input file and populates the Product Catalog database. With DBLoader, you can add large volumes of product item and category records in a single command.
- **Browser-based administration screens** that you can use to manage the Product Catalog's content and behavior. The screens allow you to find, add, edit, or remove product categories or items. Figure 1-19 shows a portion of a sample administration screen, where the administrator is editing the values for an existing item.

Figure 1-19 Sample Administration Screen

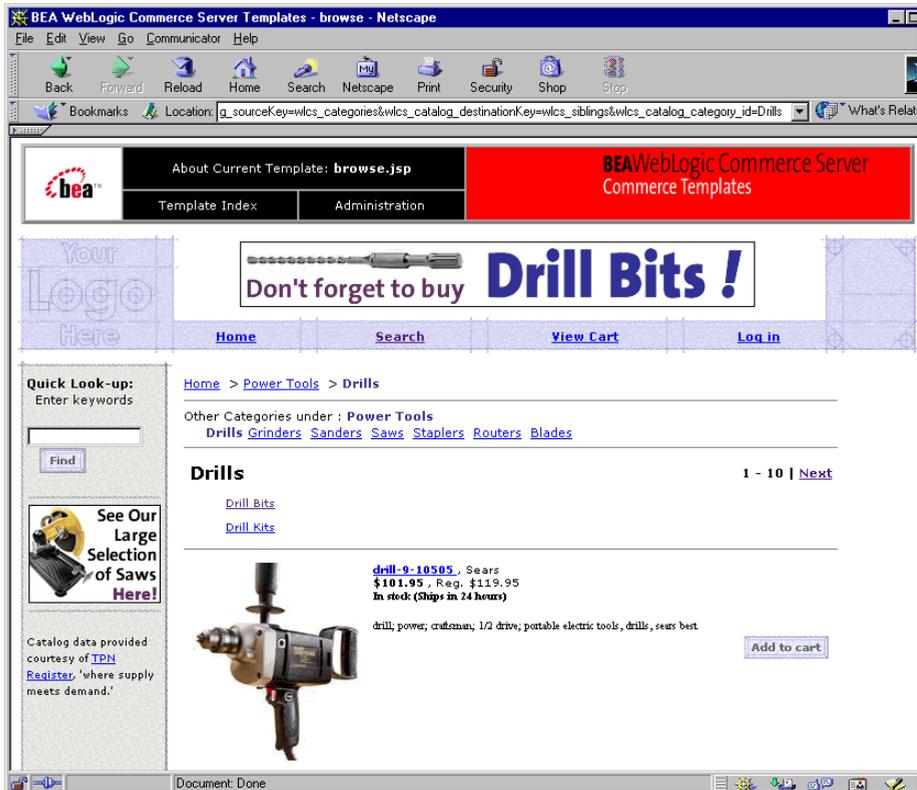


In addition to finding, adding, editing, and deleting product items and categories, you can also use the administration screens to tune the performance of your product catalog. You can adjust the in-memory cache of items and categories, a feature that can significantly improve the satisfaction level of customers shopping on your Web site.

- JavaServer Page (JSP) templates** that you can use as a starting point for your e-commerce Web development. You can change the presentation of each page to match your corporate branding requirements, design preferences, navigation options, and the content of your catalog. The JSP templates use a combination of HTML code, Java scriptlets, and JSP tags to provide the presentation layer of your Web-based product catalog.

For example, Figure 1-20 shows the running output of a sample `browse.jsp` file provided by Commerce services.

Figure 1-20 Running Output of a Sample JSP Template



- Behind the scenes, an **Application Programming Interface (API)** of pre-built Enterprise Java Beans (EJBs) and other features that provide the computing infrastructure and scalability that enables your site to support many concurrent users.
- **Flexible internationalization support** that allows you to choose among many architectural options when developing multilingual product catalogs. These features will help you internationalize your system and render a localized version of each category or item on a Web page, including text descriptions, images, item cost, type of currency, and so on.

For more information about the Product Catalog, refer to the *Guide to Building a Product Catalog*.

Payment and Order Services

The process customers go through when making a purchase from your Web site is one of the most common but complex aspects of an e-business. To help you get to market faster than your competitors, WebLogic Portal provides out-of-the-box Managing Purchases and Processing Orders services.

These services, which consist of JSP templates and Webflow components, provide default implementations for the following order-related functions:

- A shopping cart, which stores items that a customer purchases from your e-business. Shopping cart data includes a unique identifier for the product item, a quantity, a price, discounts, taxes, and so on.
- Shipping Services that record the shipping information related to a customer's order and calculate shipping costs.
- A Tax API that allows you to integrate with the tax service of your choice.
- A Payment API that allows you to integrate with the Payment Service of your choice.
- Pages that display an order summary and confirmation.

Figure 1-21 shows the JSP for customer checkouts.

Figure 1-21 checkout.jsp Template

The screenshot displays the 'Final Checkout Review' page. At the top, there is a navigation bar with 'About Current Template: checkout.jsp', 'Template Index', 'Administration', and 'BEA WebLogic Commerce Server Commerce Templates'. A banner for 'Safety Equipment' is visible. The sidebar on the left contains links for 'Welcome Demo Customer', 'View Profile', 'Logout', 'View History', 'Orders', and 'Payments'. The main content area includes sections for 'Shipping Destination' (Demo Customer, One Main Street, DENVER, CO-80212, United States), 'Shipment Splitting Preferences' (Ship all at once), 'Special Instructions', 'Ship Via' (Second Day Air), and 'Method of Payment' (Credit Card: xxxxxxxxxxxx8484). Below this is a table summarizing the order details.

ID	Description	Quantity	Our Price	SubTotal
9-10144	drill-9-10144	1	\$ 62.95	\$ 62.95
				Discount (xxxxxx) \$ -10.00
				Shipping & Handling \$ 4.95
				Total tax \$ 4.96
				Total due \$ 62.86

At the bottom of the page, there is a copyright notice: 'Copyright © 1999-2001, BEA Systems Inc.' and a 'Built On' logo.

For more information about the payment and order services, see the *Guide to Managing Purchases and Processing Orders*.

Discounts

You can use discounts to reward or entice customers by adjusting the price of an order.

Item Discounts and Order Discounts

You can offer discounts for specific items or for the overall order (order subtotal).

Item discounts are based on an item's SKU and the number of each item a customer places in the cart. For example, when a customer purchases two items where SKU=T123, the shopping cart displays a 15% discount.

Order discounts can be applied to any order or based on the subtotal of the order. For example, you could apply a 10% discount to every order or only to orders with subtotals greater than \$50. Additionally, you can specify whether to apply order discounts to the order subtotal or to the shipping cost. For example, you could specify that an order with a subtotal greater than \$100 is discounted by \$10 or that the order will be shipped for free.

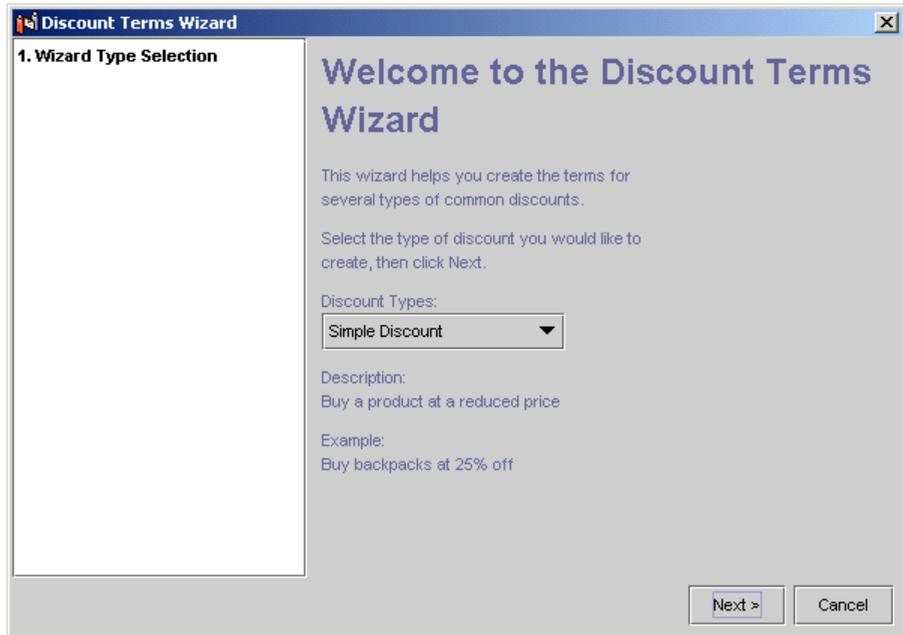
Global Discounts and User Discounts

There are two ways to offer item discounts and order discounts:

- Global discounts, which apply to all customers. For example, all customers receive free shipping. The E-Business Control Center refers to these discounts as stand-alone discounts.
- User discounts, which are designed specifically for promotional campaigns and are awarded to customers under certain conditions, such as inclusion in a customer segment. With the E-Business Control Center, which refers to these discounts as campaign discounts, you can define how long a user discount lasts; the conditions of the discount, such as buy two and get a third one for 50% off or get free shipping for orders over \$100; how many times a customer can receive a discount; and in cases where multiple discounts may apply, the priority of the discount. When you create a user discount, sure that the dates for which the discount is active fall within the dates for which the parent campaign is active.

Any easy way to define certain kinds of discount is with the Discount Terms Wizard in the E-Business Control Center, as shown in Figure 1-22.

Figure 1-22 Discount Terms Wizard Start Screen



For information on displaying discounts to your customers, refer to “Setting Up JSP Tags and Scriptlets for Campaigns” in the *Guide to Developing Campaign Infrastructure*.

Internationalization Framework

The WebLogic Portal internationalization (I18N) framework provides a set of JSP tags that you use to access text labels and messages. You can use the JSP tag library do to the following (see Figure 1-23):

1. Retrieve a static text label from a resource bundle.
2. Retrieve a message from a resource bundle.
3. Initialize a page context with a particular language, country, and variant for label and message retrieval throughout a page.
4. Set the content type (text/html) and character encoding for a page.

Figure 1-23 An Example of Internationalization Code

Before Internationalization

```
<html>
<body>
Hello!
</body>
</html>
```

} Hard coded text

After Internationalization

```
<%@ taglib uri="i18n.tld" prefix="i18n" %>
<%
// Array that defines two languages preferences -
// English and Spanish in that order of preference.
String[] languages = new String[] { "en", "es" };

// Definition of a single language preference
String language = "en";
%>

<i18n:localize language="<%=language%%"
bundleName="i18nExampleResourceBundle"/>
<html>
<body>
<i18n:getMessage messageName="greeting"/>
</body>
</html>
```

} Points to a tag library
 } Scriptlet defines language
 } This tag sets the language and encoding for the page.
 } Page body
 } This tag gets the text out of the resource bundle, instead of hard coding.

For more information, refer to “Creating Localized Applications with the Internationalization Tags” in the *Guide to Building Personalized Applications*.

Development Tools

WebLogic Portal provides the following tools to create and manage the data that your applications and servers use:

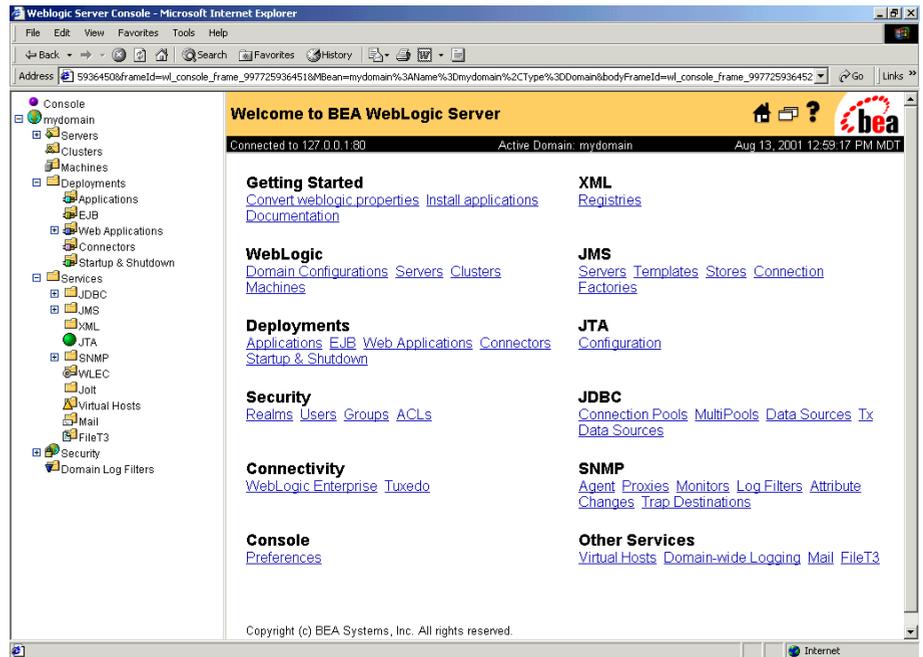
- WebLogic Server Administration Console
- E-Business Control Center
- WebLogic Portal Administration Tools

WebLogic Server Administration Console

Use the WebLogic Server Administration Console to configure your domain, enterprise application, and Web application. The WebLogic Server Administration Console runs in a Web browser and displays each configurable component in a graphical tree in the left pane. The right pane displays details and edit fields for the object selected in the left pane.

You can use the online help to find detailed information about the displayed attributes. (See Figure 1-24.)

Figure 1-24 Home Page of the WebLogic Server Administration Console



The usual process for configuring a service in the WebLogic Server Administration Console is to configure the service and then select the servers to which you want to deploy the component.

Accessing the WebLogic Server Administration Console

Each WebLogic Server domain must include a copy of the WebLogic Server Administration Console. Before you can access WebLogic Server Administration Console, you must start a domain and server.

To access the WebLogic Server Administration Console for the currently active domain, do one of the following:

- From the computer on which you installed and started the server, enter the following URL in a Web browser:

`http://localhost:7501/console`

- From a remote computer, enter the following URL in a Web browser:

`http://host-name or IP-address:7501/console`

Then enter a user ID and password with WebLogic Server system-level privileges. By default, you can use the `system` user ID and the password that you used when you installed WebLogic Portal.

Note: This user ID and password are stored in the WebLogic Server fileRealm.

To use the WebLogic Server Administration Console to modify a domain that is not currently active, do the following:

1. Access the WebLogic Server Administration Console as described above.
2. In the left pane of the WebLogic Server Administration Console, right click the current domain. From the shortcut menu, click Create or Edit Other Domains. (See Figure 1-25.)
3. In the All Domain Configurations window, click the name of the domain that you want to edit.

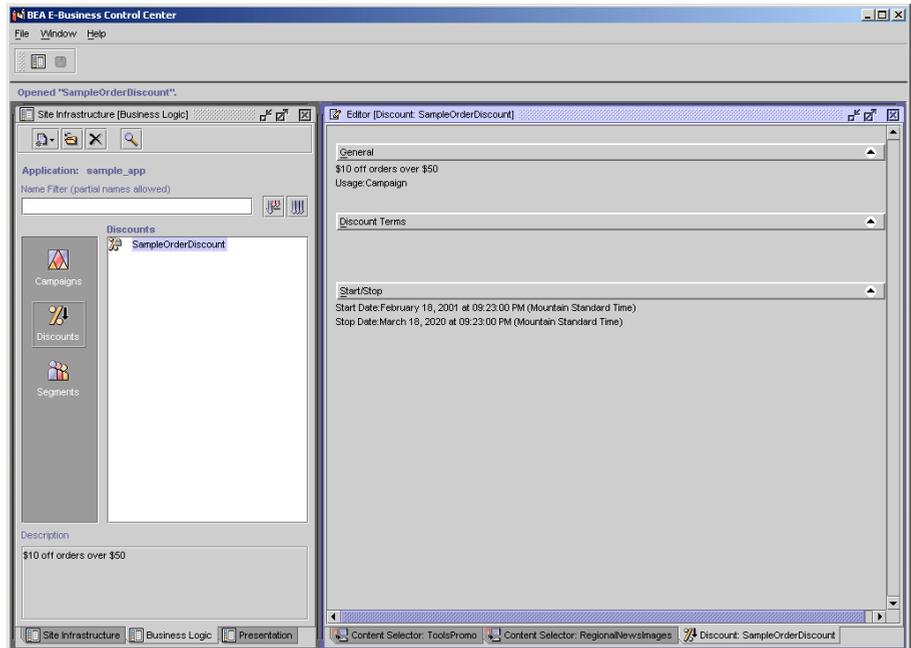
Figure 1-25 Edit a Domain



E-Business Control Center

Use the E-Business Control Center to maintain and synchronize application data. For example, you use it to create and maintain portals, campaigns, customer segments, webflows, and property sets and to synchronize your modifications with applications on active server instances. (See Figure 1-26.)

Figure 1-26 The E-Business Control Center



The E-Business Control Center runs as a Swing-based Java application. (It does not run in a browser.) You can install it on a computer that does not have a copy of the server software and you can use it to modify files without having access to a server. The data that the E-Business Control Center creates must be synchronized with an application that is deployed on an active server. You use the E-Business Control Center to synchronize data.

For information on installing the E-Business Control Center, refer to *Installation Guide*. For information on using the E-Business Control Center, refer to *Guide to Using the E-Business Control Center* or the E-Business Control Center online help.

WebLogic Portal Administration Tools

Use the WebLogic Portal Administration Tools to maintain your business data in the RDBMS data repository. For example, you use it to create categories in your product catalog or to add or remove properties from your customer profiles.

Modifications from the WebLogic Portal Administration Tools are available immediately to the applications that use the database instance.

Accessing the WebLogic Portal Administration Tools

The WebLogic Portal Administration Tools is a Web application that is installed with each reference enterprise application. Because some of the reference domains contain multiple enterprise applications, each enterprise application must provide a unique URL for accessing its copy of the WebLogic Portal Administration Tools. For information on Web applications and URLs, refer to “URLs for Accessing Web Applications” on page 1-8.

1 Architectural Overview

To access the WebLogic Portal Administration Tools, do the following:

1. Start a server on which you have deployed the WebLogic Portal Administration Tools Web application.
2. Then do one of the following:
 - From the computer on which you installed and started the server, enter the following URL in a Web browser:
`http://localhost:7501/admin-tool-name/index.jsp`,
where *admin-tool-name* is one of the following:
p13nAppTools
portalTools
wlcsAppTools
 - From a remote computer, enter the following URL in a Web browser:
`http://host-name or IP-address:7501/admin-tool-name/index.jsp`
 - On Windows systems, you can use the following Start menu path to access the WebLogic Portal Administration Tools for the portal enterprise application:
Start →BEA WebLogic E-Business Platform →BEA WebLogic Portal →Administration Tools
 - Click the Administration link that appears in the top banner of most JSP templates. Figure 1-27 shows the link in an extract of the `main.jsp` template.

Figure 1-27 Link to WebLogic Portal Administration Tools from JSP Template



Logging In to the Administrator Account

Before the browser can open the main page of the WebLogic Portal Administration Tools, you must log into the Administrator account, as shown in Figure 1-28.

Figure 1-28 Administration Login Screen

WebLogic Portal defines an administrator account name for you. The default password is `password` (lowercase characters).

Following valid login credentials, the main administration screen is displayed, as shown in Figure 1-29.

Figure 1-29 WebLogic Portal Administration Tools Home Page

**Administration Tools
BEA WebLogic Portal**

Administration Tools Home Page

Click the icons () to launch the administration tools. To get more help as you work with the tools, click ? to view online help, or the book icon to view the developer's guide.

See the [Personalization Advisor](#) for tags and tips to help you create a sample application.

<p>User Management</p> <p>Administer users and groups and their associated properties more explanation...</p>	<p>Catalog Management</p> <p>Create and administer catalog content and structure more explanation...</p>
<p>Order Management</p> <p>View and manage product orders more explanation...</p>	<p>Payment Management</p> <p>View and administer order payments more explanation...</p>

Changing the Administrator Password

The initial administrator account information for the WebLogic Portal Administration Tools is:

Username: administrator
Password: password

To change the password of the administrator account, follow these steps:

1. Open the WebLogic Portal Administration Tools Home Page, as described in “Accessing the WebLogic Portal Administration Tools” on page 1-63. When you are prompted to log in, enter the current password.
2. On the WebLogic Portal Administration Tools Home Page, click the User Management icon, as shown in Figure 1-30.

Figure 1-30 User Management Icon on Main Administration Screen



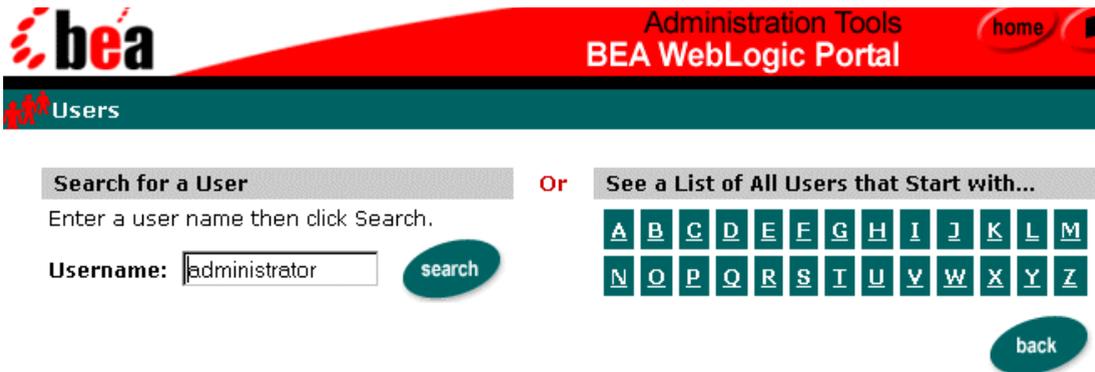
3. On the User Management screen, click the underlined Users link, as shown in Figure 1-31.

Figure 1-31 Users Link on the User Management Screen



- On the Users screen, search for the `administrator` account name, as shown in Figure 1-32.

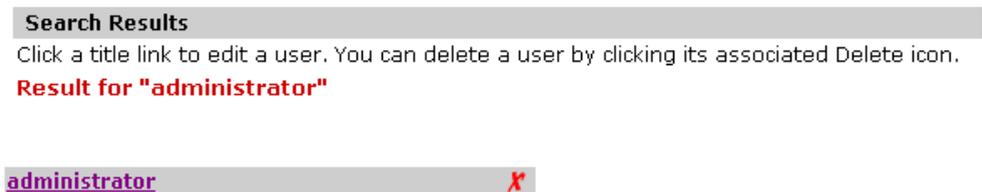
Figure 1-32 Searching for the Administrator Account



After you enter `administrator` in the Username input box, click the Search button.

- In the Search Results screen, click the underlined `administrator` link, as shown in Figure 1-33.

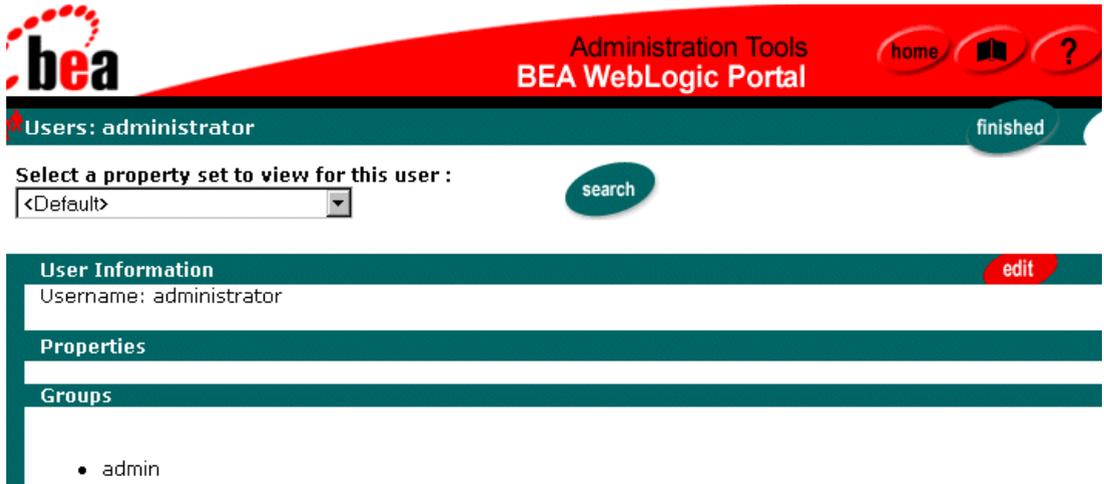
Figure 1-33 Administrator Link in the User Account Search Results



Warning: Do not click the red *X* to the right of the account name. Clicking the red *X* will delete the account.

6. On the Users: administrator screen, click the Edit button, as shown in Figure 1-34.

Figure 1-34 Edit Button on Users: administrator Screen



7. On the Users: Username screen, enter and verify the new password twice.

Warning: Make sure to remember the new password for the administrator account! That sounds obvious, but there is a potential “Catch-22” situation. To change the account password again, you must log into the WebLogic Portal Administration Tools, a step that requires the current administrator password. If you forget the account’s password, you will have to recreate the WebLogic Portal database.

After entering the new password, click the Save button. Then click the Home button at the top of the screen to return to the main administration screen.

Index

A

- Ad Conflict Resolver 1-43
- Administration Console. See WebLogic Server Administration Console
- Administration Tool. See WebLogic Server Administration Console
- ads 1-37
- Advisor 1-33, 1-45
- applications
 - p13nApp 1-10
 - petflowApp 1-10
 - portal application 1-11
 - wlcsApp 1-11
- archived and non-archived enterprise applications 1-12
- attributes of documents 1-38
- authentication 1-29

B

- BEA_HOME 1-9
- Behavior Tracking 1-34, 1-46
- BulkLoader 1-38

C

- caches 1-7, 1-44, 1-52
- campaigns 1-48
- Cascading Style Sheets (CSS) 1-16, 1-37
- categories, product catalog 1-51
- character sets 1-58

- checkout process 1-55
- connection pools 1-5
- Console. See WebLogic Server Administration Console
- content management system 1-6, 1-37
- content management system, property sets 1-34
- content management system, retrieving documents 1-42
- content selectors 1-39
- customer profiles
 - defined 1-26
 - distinguished from customer segments 1-37
 - property sets 1-33
 - security realms 1-29
 - where stored 1-7
- customer segments
 - defined 1-35
 - discounts 1-56
 - distinguished from group profiles 1-37
- customer support contact information viii

D

- DBLoader 1-51
- deployment descriptors 1-7
- deployment descriptors, definition 1-30
- discounts 1-35, 1-48, 1-56
- documentation, where to find it vii
- domains

- defined 1-5
- editing 1-61
- p13nDomain 1-10
- petflowDomain 1-10
- portalDomain 1-11
- wlcsDomain 1-11

E

- EAR (Enterprise application ARchive) files 1-12

- E-Business Control Center

- data types 1-7
 - defined 1-62
 - discount terminology 1-56
 - editing scenarios 1-49
 - format of data 1-21
 - managing property sets 1-33

- EJBs

- deployment descriptor 1-30
 - deployment location 1-4
 - representing synchronized data 1-7
 - security 1-30, 1-54

- e-mail, Mail Service 1-48, 1-50

- e-mail, property sets 1-33

- enterprise applications

- contents 1-4
 - relationship to a portal 1-14
 - security roles 1-30

- entitlements 1-17

- events

- defined 1-46
 - property sets 1-34
 - Webflow 1-19

G

- group portal 1-13, 1-15

- group profile property set 1-33

- groups, customer profile

- distinguished from customer segments

- 1-37

- hierarchies and inheritance 1-27

- security 1-29

- GUI

- E-Business Control Center. See

- E-Business Control Center

- WebLogic Portal Administration Tool.

- See WebLogic Portal

- Administration Tool

- WebLogic Server Administration

- Console. See WebLogic Server

- Administration Console

H

- HTTP

- request property set 1-33

- session property set 1-33

- unresolved requests 1-10

I

- image files 1-6, 1-16

- indexing metadata 1-38

- input processors 1-19, 1-25

- internationalization 1-58

- internationalization, product catalog 1-54

- IP addresses 1-8

J

- J2EE security role 1-30

- Java scriptlets 1-57

- JavaMail 1-50

- JSP tags

- content selectors 1-39

- internationalization 1-58

- personalization 1-45

- placeholders 1-41

- product catalog 1-53

- rules service 1-35

JSP templates, product catalog 1-53
JSP templates, registering and managing
customers 1-25

L

layouts 1-15
LDAP
 storing customer data 1-7
 Unified User Profile 1-27
 User Management 1-24
licensing 1-9
listen ports 1-7
loading metadata 1-38

M

messages 1-5, 1-8
messages, internationalization 1-58
metadata 1-15, 1-38
MIME types 1-41

N

namespaces 1-20

O

output, messages 1-5, 1-8

P

p13nApp 1-10
p13nDomain 1-10
passwords
 customer profile 1-22
 security realms 1-29
 WebLogic Portal Administration Tools
 1-65
persistence 1-19
petflowApp 1-10
petflowDomain 1-10

Pipeline components 1-19, 1-25
Pipelines 1-18
 in User Management 1-23
 scope 1-20
placeholders 1-41, 1-43, 1-48
port numbers 1-8
portal application 1-11
portal page 1-15
portal services 1-1
PORTAL_HOME 1-9
portalDomain 1-11
portals
 customer properties 1-27
 defined 1-13
 location of metadata 1-7
 rules 1-35
 security 1-30
 Webflow 1-20
portlets
 defined 1-13
 location of metadata 1-7
 personalizing 1-27
 security 1-30
 Webflow namespaces 1-20
printing product documentation viii
product catalog 1-7
product catalog, property set 1-34, 1-51
profile. See customer profile
property set
 content management 1-34
 HTTP request 1-33
 HTTP session 1-33
 location of data 1-7
 user and group profile 1-33
property sets 1-31–1-34

Q

queries 1-38
 ad placeholders 1-41
 content selectors 1-39

R

request property set 1-33

roles. See security roles

rules

- ad placeholders 1-41

- defined 1-35

- events 1-46

- location of data 1-7

- property sets 1-33

S

scenarios

- ad placeholders 1-41

- defined 1-50

schemas 1-51

scriptlets, Java 1-53, 1-57

security realms 1-29

security roles 1-13, 1-18, 1-30

sessions

- events 1-46

- portals 1-15

- property sets 1-33, 1-35

- Webflow 1-18

Shipping Services 1-54

Shockwave files 1-43

shopping cart 1-18, 1-46, 1-54

skins 1-16

support

- technical viii

T

transactions 1-29

transitions, Webflow 1-19

U

URIs 1-8

URLs 1-8, 1-38

- p13nApp 1-10

- petflowApp 1-10

- portal application 1-12

- WebLogic Portal Administration Tool
1-64

- WebLogic Server Administration
Console 1-61

- wlcsApp 1-11

User Management component 1-24

username

- for WebLogic Server Administration
Console 1-61

- WebLogic Portal Administration Tools
1-66

UUP (unified user profile) 1-24

- alternative security realms 1-29

- defined 1-27

V

visitor entitlements 1-17

W

Web applications

- context root 1-8

- default 1-10

- defined 1-4

- security 1-30

- Webflow 1-20

Webflow 1-18–1-23

- demonstration 1-10

- for Payment JSPs 1-54

- scope 1-20

Webflow Editor 1-21

WebLogic Portal Administration Tools 1-7

- accessing 1-63

- managing users and groups 1-24

WebLogic Server Administration Console
1-7

- accessing 1-60

- editing wlcsRealm 1-29

supported tasks 1-59
wicsApp 1-8, 1-11
wicsDomain 1-11

X

XML

E-Business Control Center data 1-49
portlet data 1-15
representing application data 1-7
Webflow data 1-18

