



BEA WebLogic Portal™

Strategies for Developing E-Business Web Sites

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.

Getting Started with Developing an E-Business Web Site

Document Edition	Date	Software Version
4.0	October 2001	WebLogic Portal 4.0

Contents

About This Document

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

1. Developing an E-Business Web Site

Milestone 1: Analyze Business Needs and Design a Prototype	1-3
Develop a Storyboard	1-5
Review WebLogic Portal Services and Constraints	1-5
Determine Extension Requirements	1-7
Design a Prototype	1-8
Milestone 2: Create a Base Configuration	1-8
Create a Rudimentary Web Application	1-9
Set Up Infrastructure and Tools to Support Your Web Application	1-10
Set Up the Database Repository and Load Sample Data	1-12
Set Up Developer Security	1-12
Verify the Configuration	1-13
Milestone 3: Place Files Under Source Control	1-14
Files for the Domain and Server	1-15
Files for the Enterprise Application and Web Application	1-15
Data from the E-Business Control Center	1-16
Other Files and Directories	1-17
For More Information	1-17

Milestone 4: Set Up a Development Site.....	1-18
Setting Up the Development Site (Example)	1-19
Concurrent Development (Example).....	1-21
Milestone 5: Set Up Basic Services.....	1-25
Design the Presentation for Your Site	1-25
Design the Data Flow for Your Site	1-26
Set Up Customer Profiles	1-28
Developing Profiles and Property Sets.....	1-29
Adding Customers and Customer Profile Data	1-30
Develop Customer Segments	1-31
Set Up the Product Catalog	1-33
Set Up a Content Management System.....	1-35
Milestone 6: Set Up Personalization, Commerce, and Campaign Services	1-36
Add JSP Tags for Retrieving Personalized Content.....	1-36
Set Up the Payment, Order, and Shipping Services	1-40
Set Up Campaigns	1-41
Milestone 7: Deploy in a Test Environment.....	1-42
Milestone 8: Tune Performance for a Production Environment.....	1-44
Set Up a Clustered Environment	1-44
Milestone 9: Deploy in a Production Environment	1-45

2. Documentation Roadmap for WebLogic Portal

Documents for Development Milestones	2-2
Development Roles.....	2-4
Web or User Interface Designer (WD)	2-4
Business Analyst (BA).....	2-5
Business Engineer (BE)	2-5
Developer	2-5
Visitor, Customer, and User Roles	2-7
Visitor.....	2-7
Customer	2-8
User	2-8

Index

About This Document

This document outlines the process of creating an e-business Web site that uses the services of WebLogic Portal. It includes the following topics:

- Chapter 1, “Developing an E-Business Web Site,” which recommends the order in which you start and complete your development tasks.
- Chapter 2, “Documentation Roadmap for WebLogic Portal,” which indicates which WebLogic Portal document supports each major milestone and describes role names that the documentation uses to refer to specific types of customers and developers.

What You Need to Know

This document is intended for all members of a Web-site development team. It 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.

The workflow that this document suggests does not assume or require that your organization follows any specific development methodology.

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/>.

Related Information

The following BEA WebLogic Portal documents contain information that is relevant to using the `idljtojava` compiler and understanding how to implement Java CORBA applications in the WLE system.

The following documents provide background and additional information:

- *Java™ 2 Platform Enterprise Edition Specification, v1.3*
- WebLogic Server documentation, which is available from <http://edocs.bea.com/>.

For information about methodologies that are appropriate for e-commerce Web development, refer to the following, commonly used methodologies:

- BEA SteelThreadSM, which is a methodology that BEA Architecture Validation ServiceTM has adopted to create a site architecture. For more information, refer to http://www.bea.com/service/architecture_validation.shtml.
- The Rational Unified Process®. For more information, refer to <http://www.rational.com/>.

-
- Extreme Programming (XProgramming). The following Web site contains information about XProgramming: <http://www.xprogramming.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> <pre>#include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
monospace boldface text	Identifies significant words in code. <i>Example:</i> <pre>void commit ()</pre>
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> <pre>String <i>expr</i></pre>
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> <pre>LPT1 SIGNON OR</pre>

Convention	Item
{ }	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 Developing an E-Business Web Site

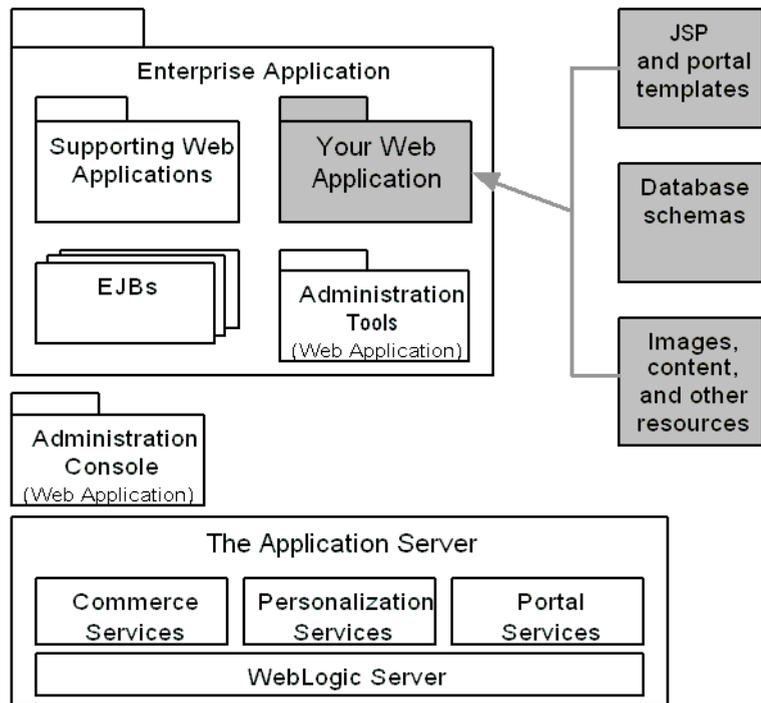
WebLogic Portal is designed to enable quick, efficient development cycles. Using its services, your development team can take advantage of the scalability and reliability of the J2EE platform without having to write large amounts of Java code.

To enable you to develop, launch, and maintain e-business Web sites, WebLogic Portal provides the following resources (see Figure 1-1):

- Building blocks for a Web application that implements your business model. The building blocks include JavaServer Pages (JSPs), portal and portlet templates, EJBs, database schemas, and configuration files. With these resources, you can design and develop a Web application that provides portals and portlets, registers and authenticates customers, manages and fulfills business transactions, presents personalized content, and delivers other value-added services to online customers.
- Java applications, such as the E-Business Control Center, WebLogic Portal Administration Tools, and WebLogic Server Administration Console, that you use to design, configure, deploy, and manage your Web application.
- A Web-application server (WebLogic Server) plus a set of e-business services that extend the functionality of WebLogic Server.

As Figure 1-1 illustrates, the Web application that you develop is one module in a larger enterprise application. With this modular approach, your development efforts focus on implementing your business model, and BEA provides the remaining infrastructure and tools for deploying and managing your Web application.

Figure 1-1 Your Web Application in Context



To develop your Web application efficiently, you can arrange iterative development cycles with multiple development roles. Some of these cycles can be completed concurrently.

This document recommends a general workflow for developing your Web application and suggests roles for your development team. The recommendations are based on dependencies (that is, some tasks must be completed before others can begin) and on the experience of WebLogic Portal developers.

The following milestones indicate major groupings of tasks and suggests checkpoints for testing your progress:

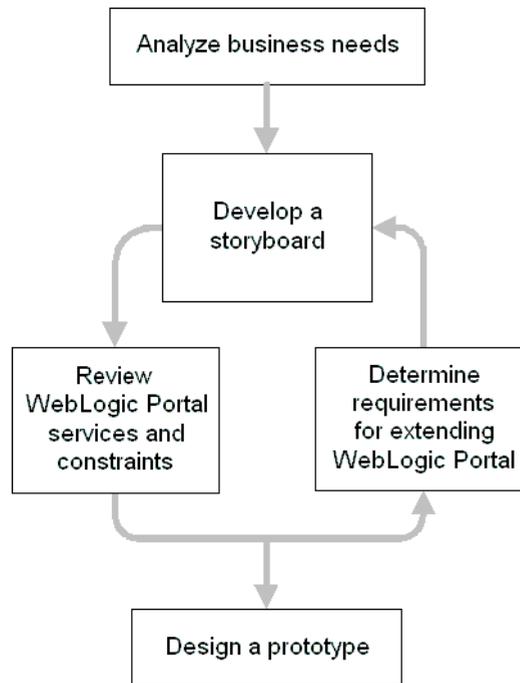
- Milestone 1: Analyze Business Needs and Design a Prototype
- Milestone 2: Create a Base Configuration
- Milestone 3: Place Files Under Source Control
- Milestone 4: Set Up a Development Site
- Milestone 5: Set Up Basic Services
- Milestone 6: Set Up Personalization, Commerce, and Campaign Services
- Milestone 7: Deploy in a Test Environment
- Milestone 8: Tune Performance for a Production Environment
- Milestone 9: Deploy in a Production Environment

Milestone 1: Analyze Business Needs and Design a Prototype

The key to a successful e-business is understanding your business needs and transferring them into a model that a Web site can support. Start your project by thoroughly analyzing your business needs, creating a storyboard or usability walkthrough that represents your business model, and comparing the model to the services that WebLogic Portal provides.

Go through these steps iteratively and increase the amount of detail in each iteration. When you have achieved a level of detail that includes specific pages, or portals and portlets, design a prototype of your Web site. (See Figure 1-2.)

Figure 1-2 Process for Designing a Prototype



This milestone includes the following tasks:

- Develop a Storyboard
- Review WebLogic Portal Services and Constraints
- Determine Extension Requirements
- Design a Prototype

Develop a Storyboard

To visualize how you and your customer conduct business on your site, develop storyboards and use cases.

Your storyboard should provide the following information:

- Whether your Web Application will use portals and portlets. Establish a list of one or more portals and a list of portlets. For example, if you are offering a Web application that features email, a stock ticker, and a news feed, you would probably want to create a portal that aggregates these features, offering within that portal an email portlet, a stock portlet and a news portlet.
- The navigation flow for customers who visit your site. Decide how customers enter your site, collect information, and start and end business transactions.
- A usability design that accounts for a positive user experience.

Review WebLogic Portal Services and Constraints

WebLogic Portal provides several sample enterprise applications, each of which features a specific set of WebLogic Portal services. In addition to demonstrating key product features, the sample applications provide a starting point (reference) for your own development. Depending on your business needs, you can adopt the basic structure of one of the sample applications. Then, as your needs change, you can make more significant modifications and use APIs to extend the functionality.

To view the reference applications, install WebLogic Portal on a computer that all team members can access via HTTP. Then do the following:

1. Start WebLogic Portal by doing one of the following:

- On Windows, click Start → Programs → BEA WebLogic E-Business Platform → BEA WebLogic Portal 4.0 → Start BEA WebLogic Portal
- On UNIX, run the following script: `PORTAL_HOME/startPortal.sh` where `PORTAL_HOME` is the directory in which you installed WebLogic Portal.

When WebLogic Portal starts successfully, it displays messages that are similar to the following examples:

```
<Oct 15, 2001 9:20:13 AM MDT> <Notice> <Management>
<Application Poller not started for production server.>

<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>

<Oct 15, 2001 9:21:51 AM MDT> <Notice> <WebLogicServer>
<Started WebLogic Admin Server "portalServer" for domain
"portalDomain" running in Production Mode>
```

Open a Web browser and enter one of the following URLs, where *IP-address* is the IP address for the computer that is running the WebLogic Portal instance:

To view...	Enter...
portal, which demonstrates the use of portals	<code>http://localhost:7501/stockportal</code> or, from a remote machine, <code>http://IP-address:7501/stockportal</code>
p13nApp, which demonstrates personalization services	<code>http://localhost:7501/p13n</code> or, from a remote machine, <code>http://IP-address:7501/p13n</code>
wlcsApp, which demonstrates commerce, campaign, and personalization services	<code>http://localhost:7501/wlcs</code> or, from a remote machine, <code>http://IP-address:7501/wlcs</code>

For more information about the reference applications, refer to the following:

- *Portal Tour*
- *JSP Commerce and Campaign Tour*

Determine Extension Requirements

Because each e-business Web site presents a unique set of requirements and constraints, you can extend or replace most of the services that WebLogic Portal provides.

For information on a few of the features that are commonly extended, refer to the following:

- “Extending Webflow and Pipelines” in the *Guide to Managing Presentation and Business Logic: Using Webflow and Pipeline*
- “Extending the Data Model” in the *Guide to Managing Purchases and Processing Orders*
- “Using the API to Extend the Product Catalog” in the *Guide to Building a Product Catalog*

Design a Prototype

After you review the sample enterprise applications, design an outline (prototype) of your e-business Web site. Your design should include the following information:

- A list of pages, or portals and portlets, that your customers will visit.
- The layout of the pages and the navigational options.
- A list of include files. For example, rather than rewrite the top banner on each page of your site, you can write it in a single file and then use an include statement to import the file into each JSP.
- The sample Web application that you plan to use as a template. Determine whether it is easier for you to start with an existing application and modify it to fit your business model or for you to create all or most of the JSPs without using templates.
- A collection of images, navigation bars, logos, and buttons to be used when the site is assembled.

Milestone 2: Create a Base Configuration

After designing a prototype, install WebLogic Portal on a separate machine. Then use the files in this installation to complete the following tasks:

- Create a Rudimentary Web Application
- Set Up Infrastructure and Tools to Support Your Web Application
- Set Up the Database Repository and Load Sample Data
- Set Up Developer Security
- Verify the Configuration

Create a Rudimentary Web Application

To start your development process, create a Web application that is based on your prototype design. The application should include the following components:

- Basic directory structure.
- Deployment descriptors. *Deployment descriptors* are XML files that declare the resources in an application and determine how the application server deploys and administers the application.
- JSPs. Add each of the JSPs that your design prototype requires. At this point, empty files are sufficient.
- Components from a sample Web application. If you plan to use one of the sample Web applications as a template for your own development, include the application in your base configuration.

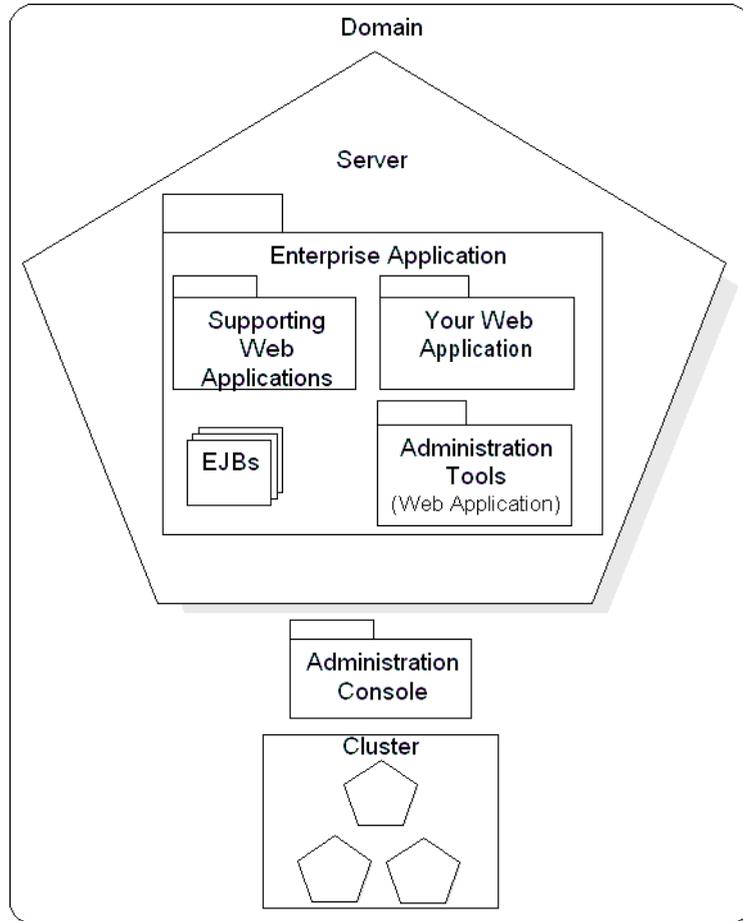
The Web application does not need to function at this point in the development cycle. For information on creating a Web application, refer to “Assembling Your Web Application” in the *Deployment Guide*.

Set Up Infrastructure and Tools to Support Your Web Application

Next, you must set up an environment that provides the infrastructure and tools for deploying and managing your Web application. The environment contains the following components (see Figure 1-3):

- A WebLogic Server domain, which contains definitions for servers and clusters.
- The WebLogic Server Administration Console, which you use to configure the domain, servers, and clusters.
- A server, which configures such services as HTTP listen ports, XML parsers, and JDBC connection pools for a single instance of the server.
- An enterprise application that contains the following modules:
 - The WebLogic Portal Administration Tools Web application. Business Engineers (BEs) use this Web application to manage the customer, catalog, and other data that WebLogic Portal stores in its RDBMS repository.
 - Helper Web applications, which the E-Business Control Center uses to communicate and exchange data with your application. Business Analysts (BAs) use the E-Business Control Center to set up portal, commerce, personalization, and campaign services. The E-Business Control Center can be installed on any machine that is able to communicate with the server where the application directory resides.
 - WebLogic Portal EJBs.
 - Your Web application.

Figure 1-3 Infrastructure for Your Web Application



For more information, refer to the following:

- For information on creating a domain and server, refer to “Deploying Domains and Server Configurations” in the *Deployment Guide*.
- For information on creating an enterprise application and Web application, refer to “Assembling Your Web Application” and “Assembling and Deploying Enterprise Applications” in the *Deployment Guide*.
- For information on creating a Web application that uses portals, see *Getting Started with Portals and Portlets*.

Set Up the Database Repository and Load Sample Data

WebLogic Portal uses a database to store your business data such as customer IDs and profiles, catalog items, and payment and order histories.

The reference enterprise applications provide a Cloudscape RDBMS that is loaded with sample data. This database is for demonstration purposes only. For your development and production environments, set up an RDBMS type that is listed in the “Supported Platforms” section of the *Installation Guide*.

To set up your RDBMS, WebLogic Portal provides scripts that create schemas and other data structures. It also provides scripts that can load the sample data into your database. We recommend that you load the sample data and use it to verify that the system functions properly.

For information on setting up the database repository and loading sample data, refer to the *Deployment Guide*.

Set Up Developer Security

When you install WebLogic Portal, it provides a set of default user IDs and passwords that provide access and modification permissions for your application’s resources. Depending on your security needs in the development environment, you can set up additional user IDs and passwords. Different types of resources require different permissions schemes:

- For the file-based resources that you develop for your Web site (such as JSPs and E-Business Control Center data files), use your content-management system or source control system. For information about the security schemes that these systems provide, refer to the systems’ documentation.
- To create additional user IDs for the WebLogic Portal Administration Tools, which you can use to edit the data in the RDBMS repository, log in to the WebLogic Portal Administration Tools as `administrator` and set up user profiles for your developers. (A *user profile* contains user IDs and passwords for a user.)

Add the new user profiles to the `SystemAdministrator` group. Any user profile in the `SystemAdministrator` group can use all the features available in the WebLogic Portal Administration Tools. In other words, authenticated users

are "superusers;" the application permits them to make all possible modifications and there are no restrictions based on the user's organizational role. For information on starting and logging in to the WebLogic Portal Administration Tools, refer to "WebLogic Portal Administration Tools" in the *WebLogic Portal Architectural Overview*.

- To provide privileges to synchronize data and browse ads, e-mail, and catalog data from the E-Business Control Center, log in to the WebLogic Portal Administration Tools as `administrator` and set up user or group profiles for your developers. Add each developer's user profile to the `SystemAdministrator` group or to a subgroup of `SystemAdministrator`.
- To provide access privileges to the WebLogic Server Administration Console, log in to the WebLogic Server Administration Console as `system`, add new user IDs and passwords to the `fileRealm`, and assign the user IDs to a group. Then adjust the ACLs to grant group access to specific resources. For more information about setting up security in the WebLogic Server Administration Console, refer to "Managing Security" in the *WebLogic Server Administration Guide*.

Note: When you start WebLogic Server, it always refers to the user IDs and passwords that are in the `fileRealm` security realm. After the server starts, it retrieves its configuration and uses the `wlcsRealm` security realm for all subsequent requests for authentication.

For more information, refer to the following:

- For information on managing user IDs and passwords in the RDBMS repository, refer to online help in the WebLogic Portal Administration Tools.
- For more information about security realms, refer to "Managing Security" in the *WebLogic Server Administration Guide*.

Verify the Configuration

To verify that your domain and enterprise application are configured properly, start the server and address any error messages that WebLogic Portal displays. For information on debugging your domain, refer to the *Deployment Guide* and the *WebLogic Server Administration Guide*.

Milestone 3: Place Files Under Source Control

After you create a base configuration, we recommend that you place it under source control.

This section contains the following subsections:

- Files for the Domain and Server
- Files for the Enterprise Application and Web Application
- Data from the E-Business Control Center
- Other Files and Directories

Files for the Domain and Server

To control your domain and server configurations, place the files that are listed in Table 1-1 under source control.

Table 1-1 Domain and Server Files

Component	Description
<code>config/yourDomain/config.xml</code>	Contains configuration information for a domain named <i>yourDomain</i> and for all servers that you create in the domain.
<code>config/yourDomain/config.xml/fileRealm.properties</code>	Contains the user ID and password that WebLogic Server requires to start.
<code>config/yourDomain/SerializedSystemIni.dat</code>	Contains encrypted information required to start WebLogic Server. If you modify the system password, this file and <code>fileRealm.properties</code> must be writable.

Files for the Enterprise Application and Web Application

To control the sources for your enterprise application, place all files in the `applications/yourApplication` directory tree under source control with the following exception:

Any directories under `applications/yourApplication/WEB-INF` with names that start with `_tmp_war` contain compiled versions of your Web applications. WebLogic Portal generates these directories when you start a server. You do not need to place these generated directories under source control.

If you created a skeletal enterprise application as described in Milestone 2: Create a Base Configuration, this directory tree contains a deployment descriptor and the beginnings of your Web application. These components will change frequently during development.

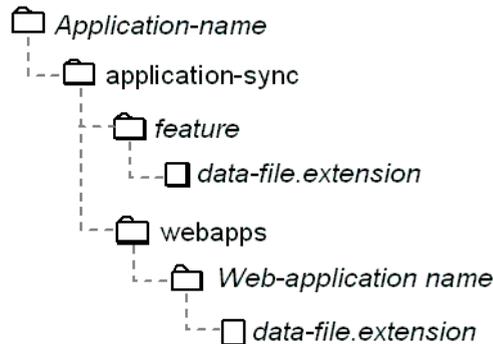
The directory tree also contains EJBs and helper Web applications. While we recommend that you do not modify these modules, they are an integral part of your enterprise application and merit inclusion in a source control system. For example, different enterprise applications that you create might require different sets of EJBs. In

addition, new releases of WebLogic Portal might include new or modified EJBs and Web applications. If this occurs, you can use your source control system to keep track of which version EJB belongs in a specific enterprise application.

Data from the E-Business Control Center

The E-Business Control Center creates and maintains such data as customer types (customer segments), promotional campaigns, portal and portlet data, and logic for displaying personalized ads. The E-Business Control Center organizes its data in a directory tree of XML files. (See Figure 1-4.)

Figure 1-4 Data from the E-Business Control Center



Each installation of the E-Business Control Center creates and maintains its own directory tree on the local host.

Because this data defines the behavior of your Web application, it is essential that you place this directory under source control. For more information about the files and file types that the E-Business Control Center creates, refer to “Synchronizing Application Data” in the *Deployment Guide*.

Other Files and Directories

The `PORTAL_HOME/db` directory contains scripts that you use to create the WebLogic Portal database schema for your development and production database. In most cases you need to modify these files to support your specific environments. To keep track of your modifications, and to maintain separate sets of these scripts, place all of the files in this directory under source control with the following exceptions:

The `PORTAL_HOME/db/cloudscape` and `PORTAL_HOME/db/data` directories contain data for the sample Cloudscape database. Placing these directories under source control is optional, depending on how heavily you rely on the sample database in your development environment.

WebLogic Portal generates log files to contain messages from the various server instances. While it is not necessary to place log files under source control, we recommend that you make sure the log files are backed up or otherwise safeguarded. You determine where log files are located when you create a domain and server configuration.

For More Information

Refer to the following topics:

- For information on the application server's file structure, refer to "Deploying the Application Server" in the *Deployment Guide*.
- For information on a domain's file structure, refer to "Deploying Domains and Server Configurations" in the *Deployment Guide*.

Milestone 4: Set Up a Development Site

When multiple developers work concurrently on a single set of source files and business documents, they can overwrite each others' work or create sets of changes that are incompatible and difficult to debug.

To facilitate concurrent development, we recommend the following model:

- Organize your development tasks into discrete units. For example, one task is creating a promotional campaign and another is modifying the flow of data through the Web site (Webflow).
- Use a separate installation of the E-Business Control Center for each development task, especially for development tasks that have the potential to introduce incompatible changes.
- For each installation of the E-Business Control Center, deploy an instance of your enterprise application under a unique name.
- Use a content management or source-control system to manage the data that the separate installations create, control access to files, and maintain a history of changes.

This section provides an example in which an Application Assembler sets up a site for developing an enterprise application named Bank. The Bank enterprise application includes a Web application named Invest, which provides customers the ability to manage their investments online. A Business Engineer (BE) named Pat creates campaigns and customer segments for the enterprise application while a developer named Mauricio creates Webflows for the Invest Web application. To prevent the introduction of incompatible changes and to facilitate debugging, each developer works with a separate E-Business Control Center and separate instance of the enterprise application.

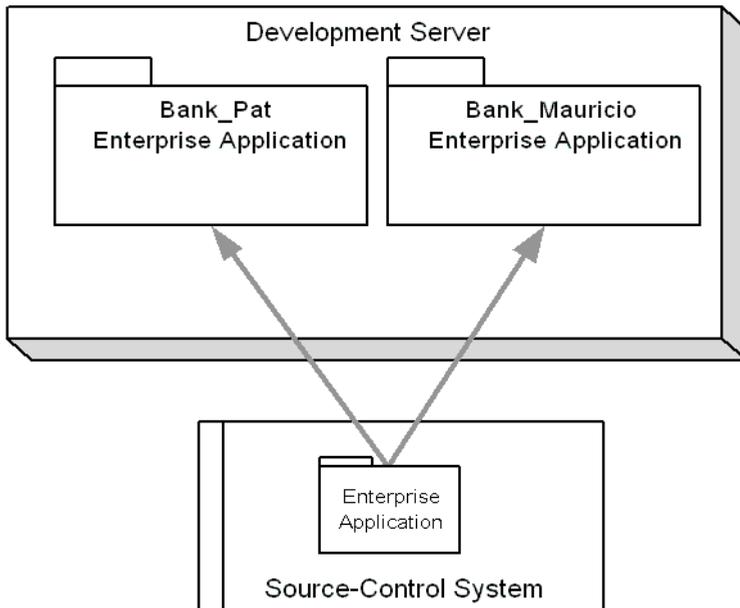
This section contains the following subsections:

- Setting Up the Development Site (Example)
- Concurrent Development (Example)

Setting Up the Development Site (Example)

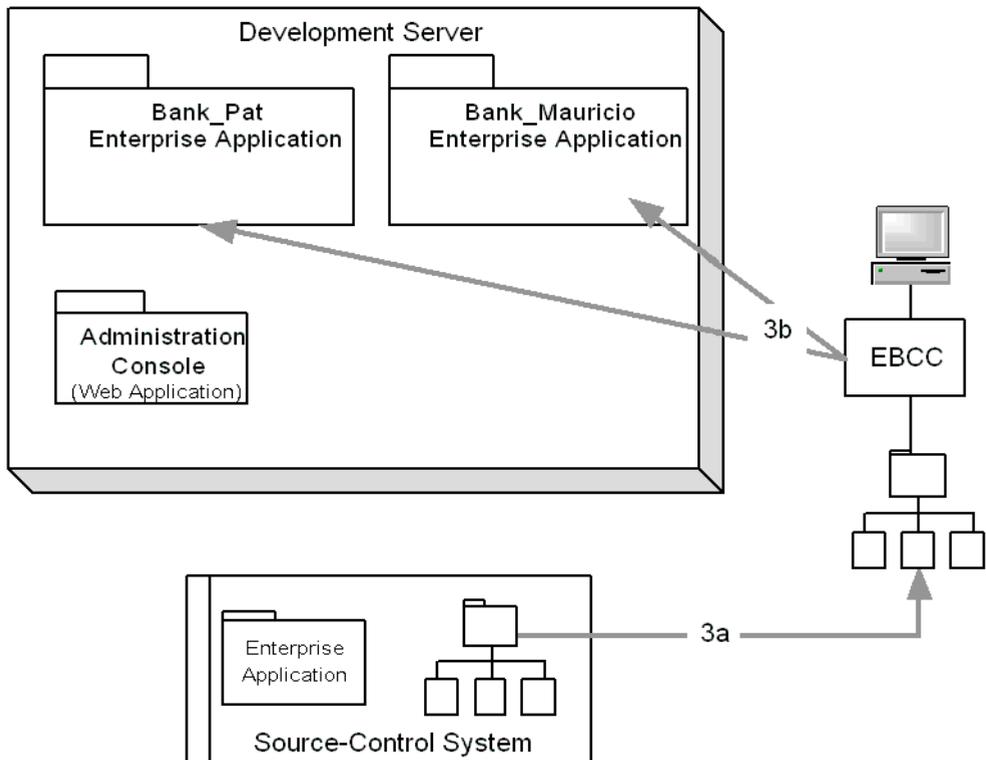
1. A System Administrator installs WebLogic Portal on a computer that will be used to host the development server.
2. An Application Assembler takes the latest version of the Bank application from the source control system and deploys two instances of the enterprise application under different names, Bank_Pat and Bank_Mauricio. (See Figure 1-5.)

Figure 1-5 Two Instances of Bank Enterprise Application



3. The Application Assembler does the following (see Figure 1-6):
 - a. Uses the source control system to copy the latest version of the E-Business Control Center data onto a workstation.
 - b. Then the Application Assembler uses the E-Business Control Center to synchronize the data with the applications. This assures that each instance of the enterprise application contains the same set of definitions for Webflows, customer segments, campaigns, and other application data.

Figure 1-6 Deploy Application Data



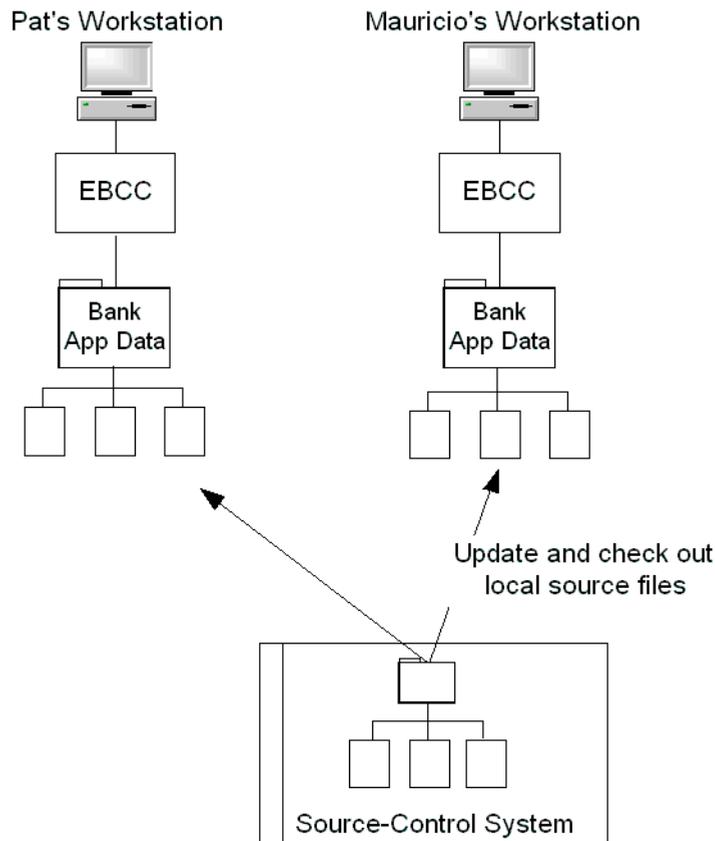
Concurrent Development (Example)

After a System Administrator sets up the development environment, Pat and Mauricio begin working concurrently on the Bank application.

1. Both developers install the E-Business Control Center and use the source-control system to copy the E-Business Control Center data onto their workstations. Then they check out the files. (See Figure 1-7.)

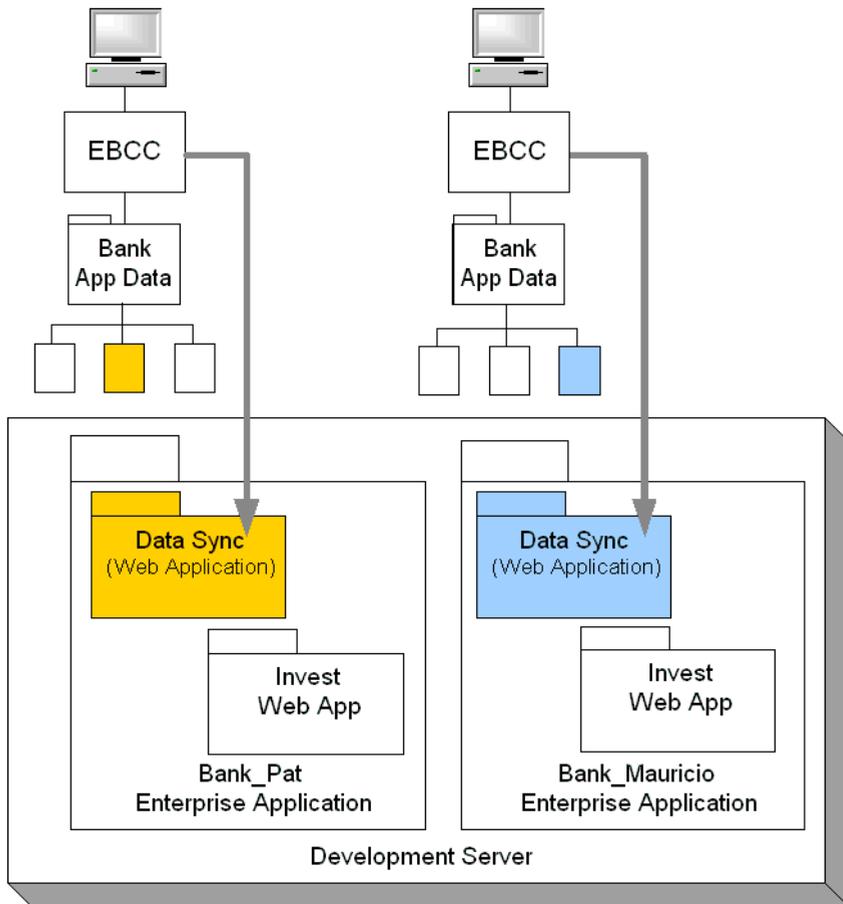
Note that each developer starts with the same data that the Application Assembler used to synchronize the enterprise applications.

Figure 1-7 Update and Check Out Source Files



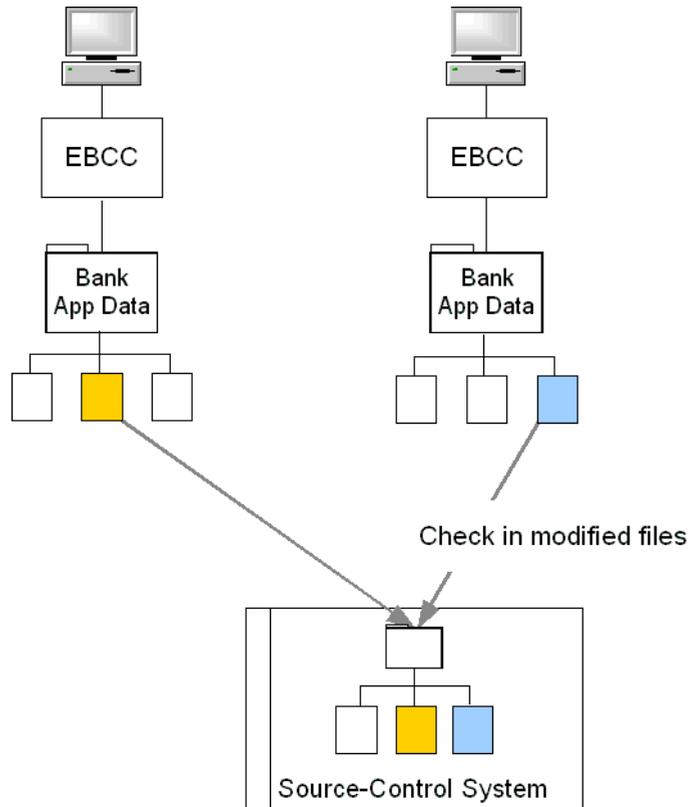
2. The developers use the E-Business Control Center to create and modify campaigns, customer segments, and Webflows.
3. When the developers are ready to see the results of their modifications, they use the E-Business Control Center and the DataSync Web application to synchronize with their private instantiations of the enterprise application. The DataSync Web application helps the E-Business Control Center to exchange data with an application. All applications include their own copy of this helper Web application.

Figure 1-8 Deploy Modifications to Web Applications



4. The developers test their modifications on their individual enterprise applications. They iteratively modify, deploy, and test until they have successfully completed the development task.
5. The developers check their sources into the source-control system. (See Figure 1-9.)

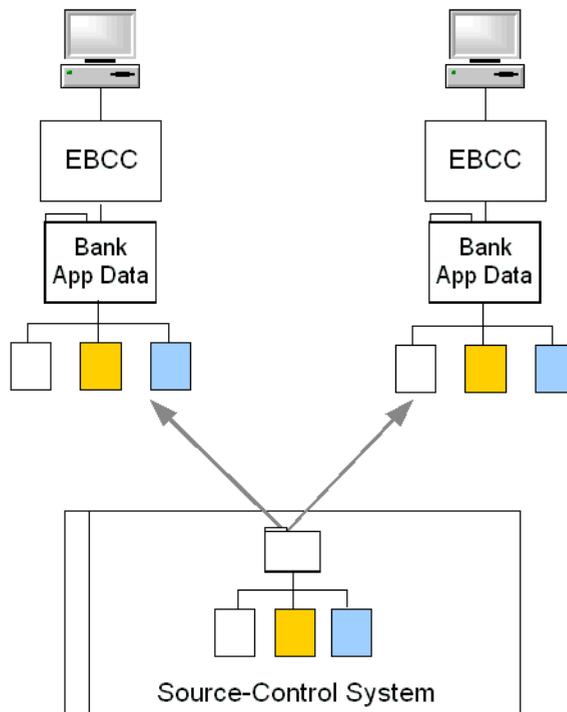
Figure 1-9 Check In Modified Files



6. To update their workstations with each others' work, the developers use the source control system. In some cases, dependencies between developer tasks require developers to update their workstations to complete their tasks.

For example, Pat sets up a placeholder that Maceo, another BA, uses in a campaign. Pat creates the placeholder and checks it in to the source control system. Then Maceo updates his workstation and uses Pat's placeholder in his campaign. (See Figure 1-10.)

Figure 1-10 Update Workstations



For more information, refer to the following:

- For information on deploying enterprise applications, refer to “Assembling and Deploying Enterprise Applications” in the *Deployment Guide*.
- For more information about deploying application data, refer to “Synchronizing Application Data” in the *Deployment Guide*.

Milestone 5: Set Up Basic Services

WebLogic Portal provides a collection of services with specialized functions. You can limit your development efforts to only the services that you need to support your business model. This milestone sets up the basic services upon which most other services depend. It includes the following tasks:

- Design the Presentation for Your Site
- Design the Data Flow for Your Site
- Set Up Customer Profiles
- Develop Customer Segments
- Set Up the Product Catalog
- Set Up a Content Management System

Design the Presentation for Your Site

A Web designer or usability engineer uses a library of components from the E-Business Control Center to design portals and portlets for your Web application.

The library includes design elements (such as graphics, skins, and layouts) and a set of functional portlets. Designers can modify the default components in the library or add their own design elements and portal definitions.

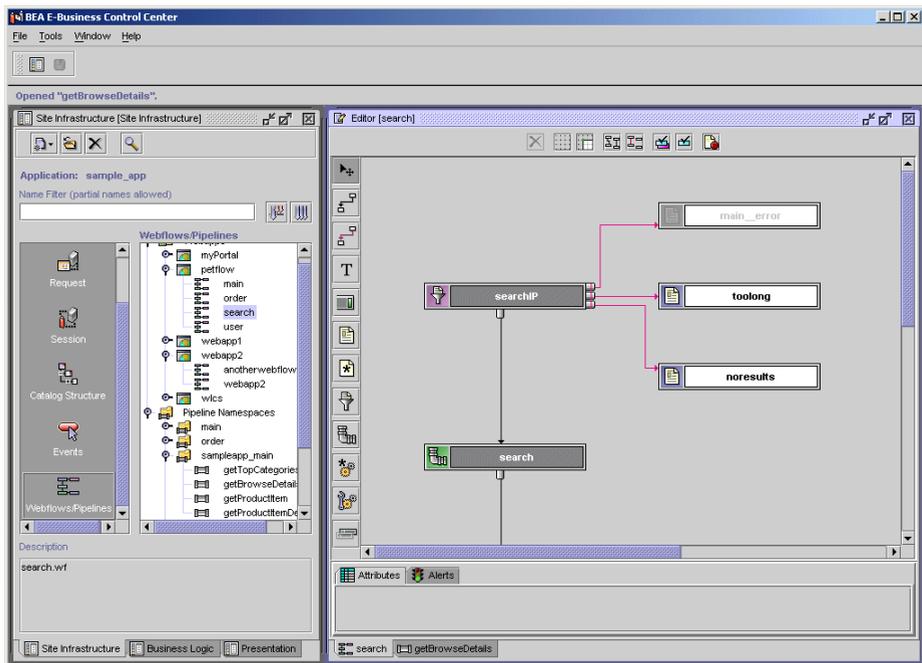
For more information, see *Getting Started with Portals and Portlets*.

Design the Data Flow for Your Site

Designing the data flow consists of two phases:

1. JSP Developers use text editors or an IDE to create the JSPs that represent all of the nodes that customers can visit on your site. They may have already created these JSPs in Milestone 2: Create a Base Configuration. In addition, JSP Developers use the E-Business Control Center to create data-processing nodes, which are discrete units of server-side business logic that validate input data and process the data.
2. BAs use the E-Business Control Center to connect all of these nodes, both JSP and data-processing nodes, into a Webflow. Webflow determines how customers and data flow through your site. (See Figure 1-11.)

Figure 1-11 WebFlow Editor in the E-Business Control Center



To set up Webflows, analyze your business use cases and determine how you want customers and data to flow through your site. For example, after customers place an item in the shopping cart, determine whether you want them to go to the product catalog or to the checkout page.

When designing the Webflow, make sure that you consider which pages are available to anonymous customers and which pages require customers to log in to your site. Balance the need for securing data with the need to make your site easy to navigate and perform quickly.

For information on designing Webflow and Pipelines, refer to the *Guide to Managing Presentation and Business Logic: Using Webflow and Pipeline*.

Set Up Customer Profiles

If you want to store secured data, such as shipping addresses and credit card information, or provide access to sensitive business data, you must set up customer profiles.

A *customer profile* is a schema that organizes the data that you collect and store about a customer. WebLogic Portal uses customer profiles to do the following:

- Authenticate customers.

At a minimum, a customer profile must contain a user ID and password and it must belong to a group profile. You use group profiles to specify access privileges for a collection of customers. For example, you can place your retail customers in one group and your wholesale customers in another group. You can give each group access to different sets of resources on your site. When a customer logs in to your site, WebLogic Portal uses the user ID and password from the customer profile to verify the customer's identity. After WebLogic Portal authenticates the customer, it determines which group profile the customer belongs to.

WebLogic Portal security consists of several layers, including customer and group profiles and J2EE security roles, which enable you to declare a set of pages in your Web application that can be accessed only by customers in specific roles. For more information about WebLogic Portal security, refer to "Security" in the *WebLogic Portal Architectural Overview*.

- Provide a personalized experience.

In addition to user IDs and passwords, a customer profile can store other information that is pertinent to your business needs. For example, you can store credit card information and billing and shipping addresses. You can also collect and store data that you use to personalize the information that your customer sees. For example, if you store data 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.

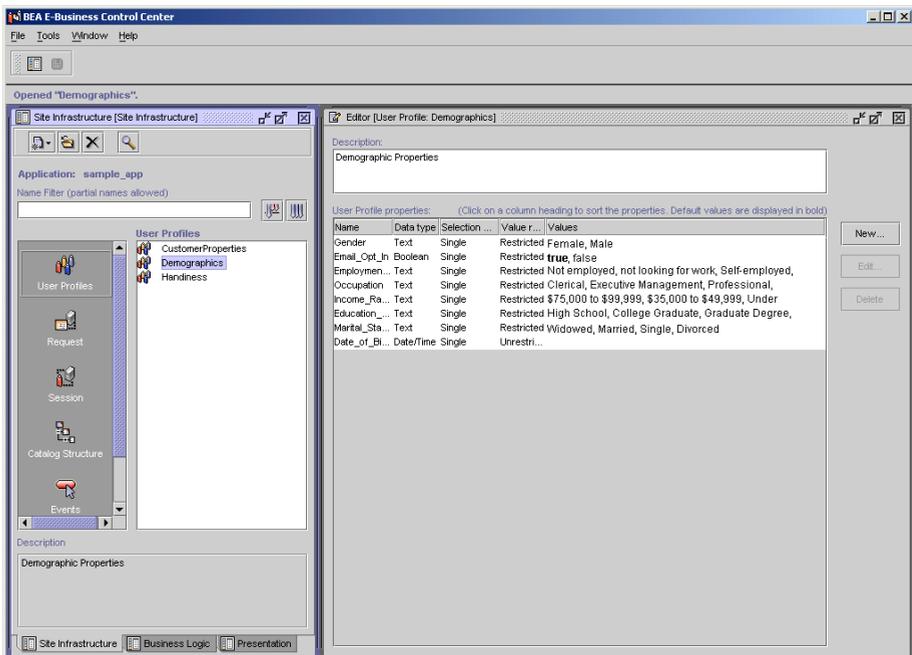
Developing Profiles and Property Sets

To develop customer and group profiles, analyze your business needs and determine which information (properties) you need to store about your customers. Make sure that your profiles contain sufficient properties for you to accurately target campaigns, discounts, and advertisements and other types of content.

Then organize the properties into sets of related information. For example, you could create one property set that stores information about customers' investment preferences and another property set that stores customers' billing and shipping preferences. Property sets modularize the data in your profiles. For any given application, you can use only the property sets that the application requires for its particular business model. All properties in your customer and group profiles must belong to a property set.

To create property sets, use the E-Business Control Center. (See Figure 1-12.)

Figure 1-12 Property Sets in the E-Business Control Center



For more information, refer to "Creating and Managing Property Sets" in the *Guide to Building Personalized Applications*.

Adding Customers and Customer Profile Data

After you create profiles and property sets, you can do the following to collect data and add organize it according to the profile schema:

- If your e-business site is new, you can use the customer registration and management services to collect user information. To set up these services, use the JSP, Webflow, and Pipeline templates from the reference e-commerce enterprise application. Figure 1-13 shows a JSP template for creating customer profiles. For more information, refer to the *Guide to Registering Customers and Managing Customer Services*.

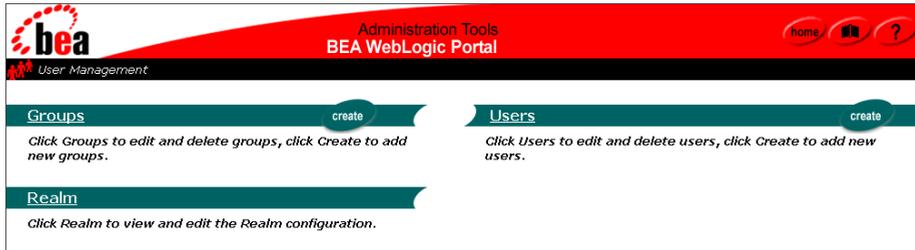
Figure 1-13 newuser.jsp Template

The screenshot shows a web browser window displaying the 'newuser.jsp' registration form. The browser's address bar shows 'http://www.ibm.com/...'. The page has a header with the IBM logo and navigation links for 'Template Index' and 'Administration'. A red banner at the top right reads 'IBM WebLogic Commerce Server Commerce Templates'. A central message box states: 'This registration form populates the Unified Customer Profile (UCP). Through the Campaign Manager you can use this information to create rules that target content based upon specific user profiles. Register now to see for yourself.' The main form area is titled 'Create a New Account' and includes a note: 'If you already have an account, use your browser's Back button to return to the Login page and log in from there.' The form fields are: First name, Middle initial, Last name, Address, Address 2, City, State / Province (dropdown), Zip/Postal Code, Country (dropdown), Home phone, Business phone, and Email address. A checkbox is checked with the text: 'Yes, I want to be offered specials and notified of discounts. Please send me promotional e-mail.' Below the main form is a 'Demographic Options' section with radio buttons for Gender (Female, Male), Date of Birth (text input with '(mm/dd/yyyy)' hint), and Occupation (radio buttons for Clerical, Executive Management, Professional, Engineering, Management, Sales). The form is styled with a blue and white color scheme.

- If you already have a large collection of existing users in the form of customers or employees, you can use the Unified User Profile to access the existing profiles through products like Lightweight Directory Access Protocol (LDAP) servers.

To manage customers and the data in the customer and group profiles, use the WebLogic Portal Administration Tools. Figure 1-14 shows the User Management screen in the WebLogic Portal Administration Tools.

Figure 1-14 WebLogic Portal Administration Tools—User Management Screen



Note: The E-Business Control Center creates the schema for profile data; the WebLogic Portal Administration Tools manages the data that is in the profile.

Develop Customer Segments

A *customer segment* is a set of criteria for dynamically grouping customers. You can use data from customer profiles, such as product preferences and age, as well as dynamic properties, such as the number of products a customer has purchased from the site during the current session.

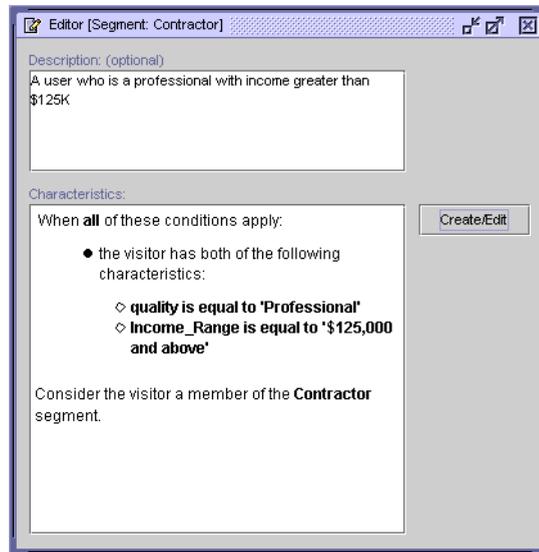
Note these important distinctions between customer and group profiles and customer segments:

- A **customer profile** is a collection of data about a customer. It is 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.
- 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.

You create customer segments in the E-Business Control Center. (See Figure 1-15.)

Figure 1-15 Customer Segments Window



Use customer segments in combination with the Event and Behavior Tracking service to trigger actions on your Web site and to keep track of how customer segments react to product offerings on your site.

The Event service is a mechanism for notifying other WebLogic Portal services when something has happened on your Web site. It keeps track of such things as when a customer begins a session and when a customer clicks on an ad banner. Events provide a detailed and comprehensive view of the entire customer life cycle across your e-business site.

The Event service passes messages to the Behavior Tracking service, which records the event data to the RDBMS repository. This information can then be used by data-mining systems to analyze the behavior of your Web site visitors.

For more information, refer to the following:

- For more information about User Registration, refer to the *Guide to Registering Customers and Managing Customer Services*.
- For more information about User Management, refer to “Creating and Managing Users” in the *Guide to Building Personalized Applications*.
- For more information on creating and using customer segments, refer to “Using Customer Segments to Target High-Value Markets” in *Guide to Using the E-Business Control Center*.
- For more information about Events and Behavior Tracking, refer to the *Guide to Events and Behavior Tracking*.

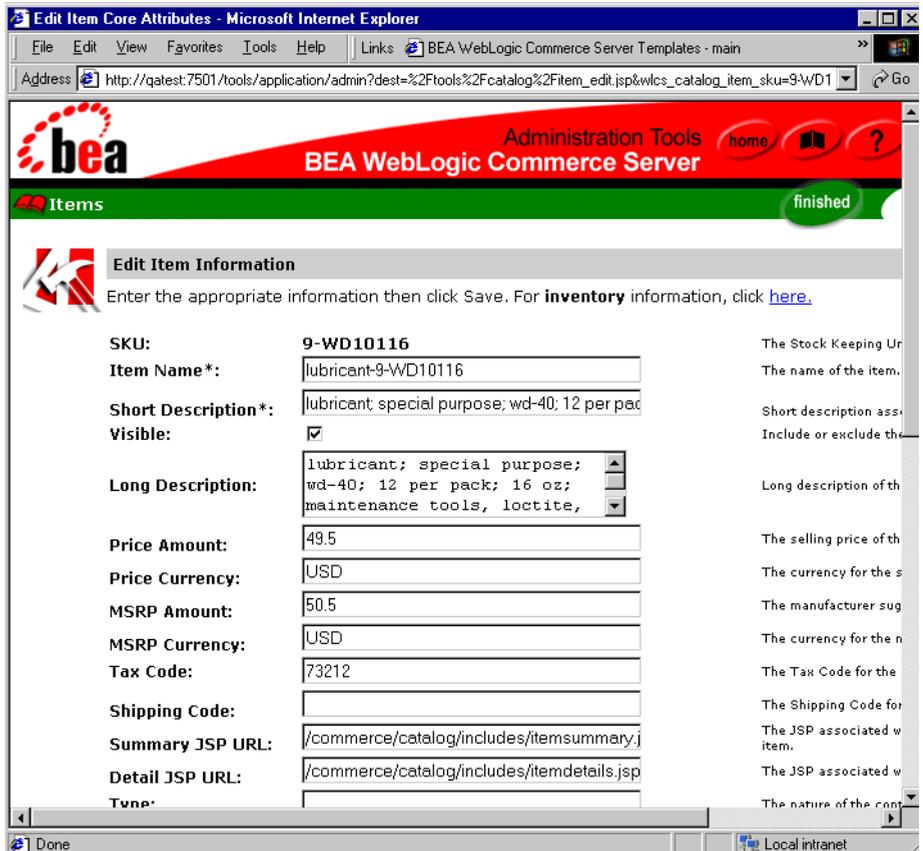
Set Up the Product Catalog

All commerce features rely on the product catalog. It organizes the products that you offer and provides descriptions and pricing information. The WebLogic Portal catalog provides tables that are located in the server’s RDBMS repository. The tables and schema determine which categories are in the catalog’s hierarchical organization and which information the catalog stores about your products. WebLogic Portal also provides sample data that you can use while developing your catalog and the JSPs that display its data to your customers.

To determine whether you need to customize the catalog, take the product tours and read the *Guide to Building a Product Catalog*. For information about customizing the catalog, refer to “Using the API to Extend the Product Catalog” in the *Guide to Building a Product Catalog*.

You can use the WebLogic Portal Administration Tools to add product categories and items to the catalog. Figure 1-16 shows an administration screen that a BE uses to edit the values for an existing item.

Figure 1-16 Sample Administration Screen



If you plan to add a large number of product categories and items to the catalog, use the DBLoader program. For more information, refer to the *Guide to Building a Product Catalog*.

Set Up a Content Management System

While you can use JSPs to present some information to your customers, to take advantage of all of WebLogic Portal features you must store the bulk of your site's content as documents in a content management system. Within the content management system, you use metadata (attributes) to describe each document. Then, to retrieve and display documents, developers use JSP tags that query the metadata.

Before you set up a content management system, analyze the types of documents that your site displays and develop a schema for consistently describing them. Using a well-defined schema will help your developers create precise queries to retrieve the appropriate set of documents for a given context. For example, you can use a query in a JSP tag to find and display documents about sailing along the Maine coast. Depending on the data that you collect about your customers, you could create a query for one group of customers that finds documents about racing; for another group of customers you could create a query for documents about cruising.

For sites with limited content personalization needs, WebLogic Portal includes a command-line utility called the `BulkLoader`. The `BulkLoader` can parse a directory of HTML files and store their URL address and metadata attributes in the WebLogic Portal RDBMS repository.

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.

For more information about using content management systems to retrieve documents, refer to "Creating and Managing Content" in the *Guide to Building Personalized Applications*.

Milestone 6: Set Up Personalization, Commerce, and Campaign Services

After you set up basic services, you can complete any of the following tasks, depending on your business needs:

- Add JSP Tags for Retrieving Personalized Content
- Set Up the Payment, Order, and Shipping Services
- Set Up Campaigns

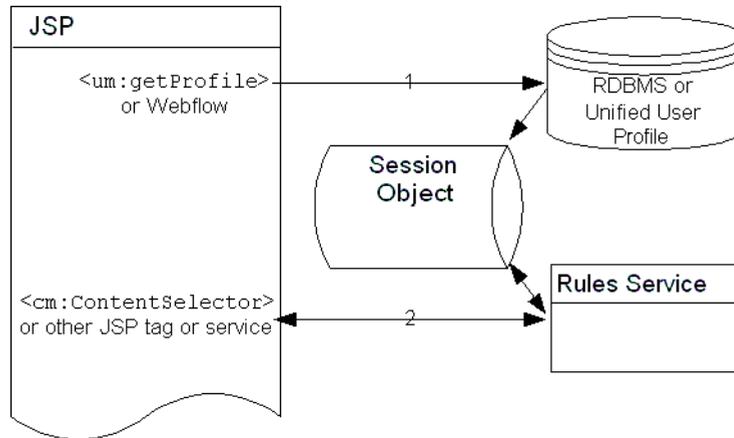
Add JSP Tags for Retrieving Personalized Content

WebLogic Portal provides several JSP tags that select and display content that is tailored to a customer. Before you can use these tags on a given JSP, you must retrieve the customer's profile data and place it in the customer's session object. Once the data is in the session object, other tags and services can use the Rules service to determine whether the customer is in a specific segment.

For example (see Figure 1-17):

1. A customer requests a JSP that includes a `<um:getProfile>` JSP tag. The JSP tag retrieves the customer's profile from the RDBMS and places it in the customer's session object.
2. A `<cm:ContentSelector>` JSP tag (which is a tag that runs queries to retrieve personalized documents from the content management system) contacts the Rules service to determine whether the customer fits into a specific customer segment. The Rules service uses the data in the customer session and returns its assessment to the content selector.

Figure 1-17 Place Customer Profile in the Session



Once you place profile data in the session, you can use several methods to retrieve documents from the content management system. Table 1-2 lists some of the methods.

Table 1-2 Methods for Retrieving and Displaying Documents

Use This Method...	When You Want To...
Content selectors and <pz:contentSelector> tags	<ul style="list-style-type: none">■ Use a centrally maintained infrastructure for matching Web site content with events, 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.</p>
<pz:contentQuery> 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 1-2 Methods for Retrieving and Displaying Documents (Continued)

Use This Method...	When You Want To...
Ad placeholders and <code><ph:placeholder></code> tags	<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.</p>
<code><ad:adTarget></code> tag	<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 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>

For more information, refer to the following topics in the *Guide to Building Personalized Applications*:

- “Working with Content Selectors”
- “Working with Ad Placeholders”
- “Creating and Managing Content”

Set Up the Payment, Order, and Shipping Services

The Payment, Order, and Shipping services provide default implementations for the most common e-business order-related functions:

- A shopping cart, which stores items that a customer decides to purchase from your e-business.
- Shipping services, which record the shipping information related to a customer's order and calculate shipping costs.
- The Taxation service, which provides support for using a Web service to calculate sales taxes for transactions on your site.
- The Payment service, which provides support for using a Web service to validate and complete credit card transactions.
- Order review and order summary, which provides customers with information about the items they have decided to purchase, as well as the shipping, payment, and tax information that is related to their order.
- The Order Status list, which indicates the location of an order in your order fulfillment process.

To set up these services, use the JSP, Webflow, and Pipeline templates from the reference e-commerce enterprise application.

For more information, refer to the *Guide to Managing Purchases and Processing Orders*.

Set Up Campaigns

You can use your WebLogic Portal Web site to run promotional campaigns. Campaigns use customer segments and real-time events to generate e-mail, ad, and discount actions that create personalized messages usually designed to motivate specific behavior. Most campaigns conceived in your organization will be developed within the context of a strategic marketing objective. For example, you may want to run campaigns aimed at increasing your average order size by employing cross-sell or up-sell tactics.

To create and deploy a campaign, a Business Engineer adds tags to JSPs, loads ads and other documents into the content management system, and sets up the Mail Service for campaigns.

Then BAs use the E-Business Control Center to complete the following tasks:

- Create user and global discounts, which can adjust the price for individual product items or for the order subtotal. For example, you can discount shipping charges for customers who have clicked on an specific ad banner on your site.
- Create ad placeholders, which reserve a location on a JSP to show ads. Each placeholder can run a variety of queries to retrieve personalized ads.
- Create email for customers
- Design scenario actions

For more information, refer to the following:

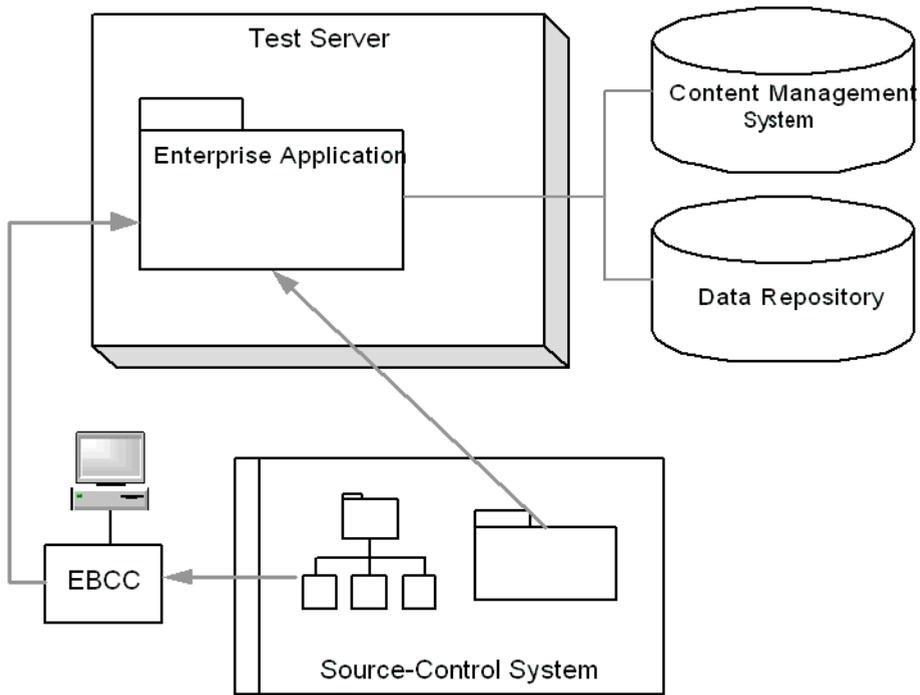
- For information about how a BE supports campaign development, refer to the *Guide to Developing Campaign Infrastructure*.
- For more information about a BA's tasks, refer to "Roadmap for Creating Promotional Campaigns" in *Guide to Using the E-Business Control Center*.

Milestone 7: Deploy in a Test Environment

After your developers complete significant development tasks, deploy their modifications onto a testing server and implement system tests. To set up a testing environment, do the following (see Figure 1-18):

1. Install WebLogic Portal on a test machine.
2. Set up an RDBMS repository and populate it with data. You can migrate data from the development environment or use sample data from the reference application.
3. Copy your domain and server configuration from your source control system to the testing environment. Then start your test domain.
4. Copy your enterprise application from your source control system to the testing environment. Then deploy the enterprise application.
5. Use the E-Business Control Center to synchronize application data from your source control system to the test environment.
6. Initiate testing.

Figure 1-18 Set Up Test Environment



Milestone 8: Tune Performance for a Production Environment

Most development environments trade Web site performance for development conveniences. For example, time-to-live intervals are biased towards showing modifications immediately, which can adversely affect performance but is desirable for developers.

When you have completed a development iteration, refer to the *Performance Tuning Guide* and “Configuring the Rules Framework” under “Introducing the Rules Manager” in the *Guide to Building Personalized Applications* for information on tuning your server and application for production-level performance.

Consider running performance tests periodically to identify potential bottlenecks as soon as possible.

Set Up a Clustered Environment

A WebLogic cluster is a group of WebLogic servers that work together to provide a powerful and reliable Web application platform. A cluster appears to its clients as a single server but it is, in fact, a group of servers acting as one. It provides two key benefits that are not provided by a single server: scalability and availability.

If you plan to set up a clustered production environment, we recommend that you test your application in a single-server environment first. Verify that the application functions properly in the less complex environment before testing the application on a cluster.

For information on setting up a cluster, refer to “Deploying Clusters” in the *Deployment Guide* and “Configuring WebLogic Servers and Clusters” in the *WebLogic Server Administration Guide*.

Milestone 9: Deploy in a Production Environment

When you are ready to make your application available to customers, consider compressing it into an EAR (Enterprise Application aRchive) file before you deploy it onto your production servers. Archiving your application simplifies administration, especially in a clustered environment. Instead of keeping track of the hundreds of files that make up your application, you need to manage only a single EAR file.

Note that you cannot incrementally update an EAR file. That is, you cannot insert a modified JSP into a deployed EAR file. Instead, you must re-archive the application and redeploy the EAR file.

For information on archiving an enterprise application, refer to "Assembling and Deploying Enterprise Applications" in the *Deployment Guide*.

2 Documentation

Roadmap for WebLogic Portal

WebLogic Portal provides a set of documents to guide your team through the process of developing an e-business Web site. One group of documents provide summaries and guided tours of WebLogic Portal features and are intended for all members of your development team.

Another group of documents address tasks that require specific skill sets, which developers in different roles may perform.

This topic summarizes the development process and indicates which document supports each major milestone. It includes the following sections:

- Documents for Development Milestones
- Development Roles
- Visitor, Customer, and User Roles

Documents for Development Milestones

Table 2-1 provides a quick look at the milestones and roles that are needed to build your personalized e-business presence on the Internet. The next section, Development Roles, describes each role in detail.

Table 2-1 Milestones and Roles

Milestone	Roles	Documents
Milestone 1: Analyze Business Needs and Design a Prototype	<ul style="list-style-type: none">■ Business Analyst (BA)■ Web or User Interface Designer (WD)■ Java/EJB Developer■ System Administrator	<ul style="list-style-type: none">■ <i>Installation Guide</i>■ <i>Portal Tour</i>■ <i>JSP Commerce and Campaign Tour</i>■ <i>Guide to Developing and Managing Portals</i>
Milestone 2: Create a Base Configuration	<ul style="list-style-type: none">■ Application Assembler/Deployer■ System Administrator	<ul style="list-style-type: none">■ <i>Installation Guide</i>■ <i>Deployment Guide</i>
Milestone 3: Place Files Under Source Control	<ul style="list-style-type: none">■ Business Engineer (BE)	<ul style="list-style-type: none">■ <i>Deployment Guide</i>
Milestone 4: Set Up a Development Site	<ul style="list-style-type: none">■ Business Engineer (BE)■ Application Assembler/Deployer	<ul style="list-style-type: none">■ <i>Deployment Guide</i>

Table 2-1 Milestones and Roles (Continued)

Milestone	Roles	Documents
Milestone 5: Set Up Basic Services	<ul style="list-style-type: none"> ■ Web or User Interface Designer (WD) ■ Business Engineer (BE) ■ HTML/JSP Developer 	<ul style="list-style-type: none"> ■ <i>Guide to Developing and Managing Portals</i> ■ <i>Guide to Managing Presentation and Business Logic: Using Webflow and Pipeline.</i> ■ <i>Guide to Building a Product Catalog</i> ■ <i>Guide to Registering Customers and Managing Customer Services</i> ■ <i>Guide to Building Personalized Applications</i> ■ <i>Guide to Using the E-Business Control Center</i> ■ <i>Guide to Events and Behavior Tracking</i>
Milestone 6: Set Up Personalization, Commerce, and Campaign Services	<ul style="list-style-type: none"> ■ Business Engineer (BE) ■ HTML/JSP Developer 	<ul style="list-style-type: none"> ■ <i>Guide to Building Personalized Applications</i> ■ <i>Guide to Managing Purchases and Processing Orders.</i> ■ <i>Guide to Developing Campaign Infrastructure</i> ■ <i>Guide to Using the E-Business Control Center</i>
Milestone 7: Deploy in a Test Environment	<ul style="list-style-type: none"> ■ Business Engineer (BE) 	<ul style="list-style-type: none"> ■ <i>Deployment Guide</i>
Milestone 8: Tune Performance for a Production Environment	<ul style="list-style-type: none"> ■ System Administrator 	<ul style="list-style-type: none"> ■ <i>Performance Tuning Guide</i>
Milestone 9: Deploy in a Production Environment	<ul style="list-style-type: none"> ■ Business Engineer (BE) ■ System Administrator 	<ul style="list-style-type: none"> ■ <i>Deployment Guide</i>

Development Roles

Each milestone in the workflow requires a distinct set of skills. To help you identify these skill sets and match them with developers on your team, we refer to each skill set as one of the following roles:

- Web or User Interface Designer (WD)
- Business Analyst (BA)
- Business Engineer (BE)
- Developer
 - Java/EJB Developer
 - HTML/JSP Developer
 - Application Assembler/Deployer
 - System Administrator

Web or User Interface Designer (WD)

Web and User Interface Designers design the user experience and interaction with a Web application. This role may merge with that of the HTML/JSP developer or Business Engineer, depending on the size of your organization or project. The WebLogic Portal tools allow the design or implementation of a Web application based on a library of web component metadata, including skins, layouts, buttons, and graphics. The designer or creative role can add to this library or modify existing components. For more information see the *Guide to Developing and Managing Portals* documentation.

A WD may also want to map out the larger design elements, working with the BA to scope a robust Web application that addresses both the business objectives and the visitor usability.

Business Analyst (BA)

Business analysts, marketing professionals, line-of-business managers, and other non-technical professionals are people who implement customized e-business strategies to capture and maintain sizable online audiences. As such, the **Business Analyst (BA)** is typically responsible for business-level changes to a site including campaign management, customer profile management, customer segmentation, catalog management, order management and content targeting, and will use the documentation to learn about and perform activities associated with these topics.

A BA's strength lies in conceiving and executing effective models for e-business, and therefore is not required to understand the technical details of how a Web site is implemented. Business analysts may need to collaborate with technical resources, but should remain in control of e-business efforts through tools with intuitive graphical user interfaces, such as the E-Business Control Center (EBCC).

Business Engineer (BE)

The **Business Engineer (BE)** is the resource BAs most often collaborate with when they require technical assistance. The BE is a person in an organization who is part BA and part Developer. The BE may:

- Manage the operations of major pieces of the Web site, or the entire site itself.
- Coordinate major and minor change processes.
- Be an active participant in overall business effectiveness monitoring and system performance monitoring.
- Be responsible for major, structural updates to an e-business Web site, for management of user profiles, user permissions, and so on.
- Perform many behind-the-scenes programming tasks that keep a Web site functioning.

Developer

Developer is a generic role name used to describe any technical individuals in your organization who participate in creating or modifying Web applications by writing code. In the Java 2 Platform Enterprise Edition (J2EE) Specification, these individuals

are referred to as Application Component Providers. Developers have special access privileges that allow them to perform these tasks—privileges that individuals in other organizational roles may not have.

Because there are numerous technical specialties within the Developer role, this role has a number of distinctions:

- Java/EJB Developer
- HTML/JSP Developer
- Application Assembler/Deployer
- System Administrator

Java/EJB Developer

The **Java/EJB Developer** is a technical resource with in-depth knowledge of Java and/or other object-oriented programming languages who may be responsible for extending a BEA product's out-of-the-box capabilities. For example, the Java/EJB Developer may extend the Webflow and Pipeline mechanisms by creating new Pipeline components or input processors and making them available to other Developers.

HTML/JSP Developer

The **HTML/JSP Developer** is an individual primarily focused on the front-end aspects of Web applications. HTML/JSP Developers may be engineers or technical Web designers that collaborate with graphic designers and use various tools to create a site design that meets their organization's marketing requirements. HTML/JSP Developers are proficient in HTML and scripting languages such as JavaServer Pages (JSP), although they may not have in-depth knowledge of Java/EJB programming or the J2EE specification. As such, HTML/JSP Developers may collaborate with other Developers to ensure that information is passed appropriately to back-end components.

Application Assembler/Deployer

The **Application Assembler/Deployer** is responsible for assembling components created by other Developers into complete J2EE applications delivered in the form of a Enterprise ARchive (.EAR) files, and for deploying Web applications and Enterprise JavaBeans components into a specific operational environment. This process may

involve moving the media to a server, generating the additional container-specific classes and interfaces that enable the container to manage the application components at runtime, and installing the application components, additional classes, and interfaces into the J2EE containers. The Application Assembler/Deployer may also resolve external dependencies and map security roles to the user groups and accounts that exist in the operational environment into which the application components are deployed.

System Administrator

The **System Administrator** is responsible for starting up, configuring, administering, and maintaining the J2EE applications as part of the enterprise's computing and networking infrastructure. The System Administrator is also responsible for overseeing the runtime well-being of the deployed J2EE applications, and typically uses runtime monitoring and management tools to accomplish these tasks.

Note: The distinctions made for the Developer role are taken from the *Java 2 Platform Enterprise Edition 1.3 Specification Final Draft Proposal*.

Visitor, Customer, and User Roles

In addition to development roles, BEA documentation and user interfaces use the following terms to describe customers with specific characteristics:

- Visitor
- Customer
- User

Visitor

Visitor describes anyone who interacts with a Web site your organization builds using BEA products. Visitors may arrive at your site at their own initiative or as a result of a targeted e-business effort (for example, a recipient of a campaign e-mail). Visitors interact with your Web sites primarily to locate information. In other words, a visitor's primary intent is in performing some type of information-gathering activity, which may be performed prior to a purchase, or may not involve purchases at all.

Customer

Customers are in the process of purchasing products or services from your e-business Web sites. Depending on the context, customers may be further qualified anonymous or authenticated. **Anonymous customers** have not registered or logged in; therefore their identity is unknown. Alternatively, **authenticated customers** have previously completed a registration process by filling out an online form, have selected a username and password combination, and have logged in or have otherwise been authenticated.

User

User is a generic role name that typically refers to users of BEA products (that is, it refers to you, a BEA customer), in places like the Javadoc and code samples. However, sometimes a technology or implementation will use the role "user" in a different context, in accordance with industry standards. An example of this alternate usage would be in discussions of the Unified User Profile (UUP), references to "user" in the context of security realms (as in users and groups), and so on.

Index

Symbols

<ad:adTarget> 1-39
<ph:placeholder> 1-39
<pz:contentQuery> 1-38
<pz:contentSelector> 1-38

A

ACLs (Access Control Lists) 1-13
Ad Conflict Resolver 1-39
ads
 data in E-Business Control Center 1-16
 in campaigns 1-41
 JSP tags to display 1-39
 synchronizing 1-13
 triggering events 1-32
anonymous customers 1-27, 2-8
APIs 1-5
Application Assembler 1-18, 2-6
authentication 1-28, 2-8

B

banners 1-8
Behavior Tracking 1-32
BulkLoader 1-35
Business Analysts (BAs) 1-10, 2-5
Business Engineers (BEs) 1-10, 1-18, 2-5

C

cache 1-38
campaigns, promotional 1-18, 1-29, 1-41
categories, product catalog 1-33
checkout process 1-27
Cloudscape database 1-12, 1-17
clusters 1-10, 1-44
config.xml 1-15
content management system 1-12, 1-18,
 1-35–1-41
content selectors 1-37
content, loading 1-35
credit cards 1-28, 1-40
customer profiles 1-28
customer profiles, where stored 1-12
customer segments 1-16, 2-5
customer segments in campaigns 1-41
customer segments, defined 1-31
customer support contact information vii

D

data synchronization 1-13, 1-20
DataSync Web application 1-22
DBLoader 1-34
debugging 1-18
deployment descriptors 1-9, 1-15
developer roles 1-2, 2-4
discounts 1-29, 1-41
documentation, where to find it v
documents, loading 1-35

domains 1-10

domains, files for source control 1-15

E

EAR (Enterprise application ARchive) files
1-45, 2-6

E-Business Control Center 1-1, 1-21

- data for 1-16

- designing portals and portlets 1-25

- developing customer profiles 1-31

- developing customer segments 1-32

- security 1-12

- separate installations 1-18

- synchronizing data 1-42

EJBs 1-1, 1-10, 1-15, 2-6

e-mail 1-5, 1-13, 1-41

enterprise applications 1-1, 1-42

enterprise applications, files for source
control 1-15

events 1-32, 1-38, 1-41

F

fileRealm 1-13

fileRealm.properties 1-15

G

groups, customer profile 1-13, 1-28–1-31

H

HTML/JSP Developer 2-6

HTTP 1-6, 1-10

I

image files 1-8, 1-39

include files 1-8

input processors 2-6

IP addresses 1-6

J

J2EE 1-28

Java/EJB Developer 2-6

JDBC connection pools 1-10

JSP tags 1-32, 1-35–1-37, 1-41

JSP templates 1-30

JSPs 1-1, 1-9, 1-26, 1-33, 1-35, 1-45

L

LDAP 1-30

listen ports 1-10

log file 1-17

M

messages 1-6, 1-13, 1-17

metadata 1-35, 2-4

MIME types 1-38

O

order fulfillment 1-40

order histories 1-12

output, messages 1-6, 1-13, 1-17

P

p13nApp sample application 1-6

passwords 1-12, 1-28

payment histories 1-12

performance tests 1-44

Pipelines 1-30, 1-40

placeholders 1-24

portal sample application 1-6

portals and portlets 1-1

portals and portlets, data in E-Business
Control Center 1-16, 1-25

portlets and portlets, prototype 1-8

printing product documentation vi

product catalog 1-33

- product catalog, loading sample data 1-12
- product catalog, location of data 1-10
- product catalog, modification privileges 1-13
- product catalog, who manages 2-5
- properties and property sets 1-29
- prototype 1-3, 1-9

Q

- queries 1-35–1-38

R

- RDBMS repository, in test environment 1-42
- RDBMS repository, setting up 1-12
- RDBMS repository, what it stores 1-10
- reference applications 1-5
- related information vi
- rules 1-32, 1-36

S

- sample applications, p13nApp 1-6
- sample applications, portal 1-6
- sample applications, wlcsApp 1-6
- sample data 1-12, 1-33, 1-42
- scenarios 1-39, 1-41
- schemas 1-1, 1-12, 1-17, 1-33
- security realms 1-13
- security roles 1-28, 2-7
- security, ACLs 1-13
- security, passwords 1-12
- segments, customer 1-33
- SerializedSystemIni.dat 1-15
- sessions 1-31, 1-32, 1-36, 1-37
- shipping addresses 1-28
- shipping costs 1-40
- Shipping Services 1-40
- Shockwave files 1-39
- shopping cart 1-27, 1-40
- skins 1-25, 2-4

- source control system 1-12, 1-20–1-23, 1-42
- starting the server 1-6
- startPortal.sh 1-6
- storyboard 1-3
- support
 - technical vii
- synchronizing data 1-13, 1-20
- System Administrator 1-19, 2-7
- SystemAdministrator group 1-12

T

- Taxation Service 1-40
- test environment 1-42
- transactions 1-1, 1-40

U

- URLs 1-6
- user IDs 1-12
- User Management component 1-31
- user profiles. See customer profiles.
- UUP (unified user profile) 1-30, 2-8

W

- Web applications
 - building blocks 1-1
 - choosing a template 1-8
 - in context 1-10
 - placing under source control 1-15
- Web Designer 1-25, 2-4
- Webflow templates 1-30, 1-40
- Webflow, defined 1-18
- Webflow, designing 1-26, 1-27
- Webflow, extending 1-7
- Webflow, who develops 2-6
- WEB-INF 1-15
- WebLogic Portal Administration Tools 1-1, 1-10, 1-12, 1-31

WebLogic Server Administration Console

1-1, 1-10, 1-13

wlcsApp sample enterprise application 1-6

X

XML 1-9, 1-10, 1-16