



BEA WebLogic Portal

**Commerce Templates
Commerce and Campaign
Features Tour**

Copyright

Copyright © 2002 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, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Portal, BEA WebLogic Server and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Commerce Templates—Commerce and Campaign Features

Part Number	Date	Software Version
N/A	July 17, 2002	WebLogic Portal 7.0

Contents

Preface

vii

1. What Is the Commerce Templates Tour?

About JSP	1-2
About the Database Schemas	1-2
About the Sample Database	1-2
About Tax Calculation	1-3
About Payment Processing	1-3

2. Starting the Tour

Prerequisite for Netscape 4.7x Users	2-1
Starting the Commerce Server and Templates	2-3
Overview of the Home Page	2-4
Features of main.jsp for Anonymous Users (Part 1 of 2)	2-5
Technical Note: Importing Tag Libraries and Java Packages	2-6
More Features of main.jsp for Anonymous Users (Part 2 of 2)	2-8
Technical Note: The Product Catalog	2-9

3. New User Buys a Product

New User Roadmap	3-2
Step 1: Search for Products	3-5
Step 2: Select an Item	3-5
Features of searchresults.jsp	3-6
Step 3: View the Shopping Cart	3-7
Features of shoppingcart.jsp	3-7
Technical Note: Pipeline Components, Pipeline Sessions, and Webflow	

Step 4: Create a New User Profile	3-9
Features of login.jsp	3-10
Features of newuser.jsp (Part 1 of 3)	3-11
Technical Note: Input Processors	3-12
Demographics in newuser.jsp (Part 2 of 3)	3-13
More Features of newuser.jsp (Part 3 of 3)	3-14
Step 5: View Successful Login	3-15
Features of successfullogin.jsp	3-15
Step 6: Choose Shipping Options	3-17
Features of shipping.jsp	3-17
Features of selectaddress.jsp	3-18
Step 7: Enter Payment Information	3-20
Features of payment.jsp	3-21
Features of paymentnewcc.jsp	3-21
Technical Note: Templates Included Within Templates	3-23
Step 8: Select a Credit Card	3-25
More Features of payment.jsp	3-25
Step 9: Purchase the Item	3-26
Features of checkout.jsp	3-27
Step 10: Review the Order Confirmation	3-28
Features of confirmorder.jsp	3-29

4. Registered User Buys a Product

Registered User Roadmap	4-2
Step 1: Browse the Catalog and Select an Item	4-5
Features of browse.jsp (Part 1 of 2)	4-5
More Features of browse.jsp (Part 2 of 2)	4-7
Features of details.jsp	4-8
Step 2: Review the Shopping Cart and Check Out	4-9
Technical Note: The Security Model	4-11
Detour: Review the Home Page for Authenticated Users	4-13
Features of main.jsp for Authenticated Users	4-13
Step 3: Choose Shipping Options	4-15
Step 4: Enter and Select a New Shipping Address	4-16

Features of selecttaxaddress.jsp	4-19
Step 5: Select a Credit Card.....	4-20
Step 6: Purchase the Items.....	4-21
Step 7: Review Order Confirmation.....	4-22
5. Registered User Adds a Credit Card	
User Profile Roadmap	5-1
Step 1: Access Your User Profile.....	5-3
Step 2: Review Your User Profile	5-3
Features of viewprofile.jsp	5-4
Step 3: Add a Credit Card	5-5
Features of profilenewcc.jsp	5-6
Step 4: View Your Order History and Status	5-7
Features of orderhistory.jsp.....	5-9
Features of orderstatus.jsp.....	5-10
Step 5: Log Out	5-11
6. Example of Customizing the Templates	
Step 1: Understand the Add to Cart Link	6-2
How the Link Anchor Invokes Webflow	6-2
How Webflow Resolves the Link	6-3
Step 2: Modify Add to Cart.....	6-5
Directing the Output of the getTopCategoriesIP Input Processor	6-5
Step 3: Verify the Modification.....	6-6
What's Next?.....	6-8
7. A Sample Campaign	
Introduction and Prerequisites	7-1
Examining a Campaign: TourCampaign1	7-2

Index



Preface

Welcome to the BEA Commerce Templates—Commerce and Campaign Features tour. In addition to this document, we encourage you to use the following resources, as well.

Finding documentation online 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>.

Providing documentation feedback 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. Please indicate that you are using the documentation for the WebLogic Portal release **7.0**.

Contacting BEA WebSUPPORT If you have any questions about this version of WebLogic Portal, or if you have problems installing and running WebLogic Portal, contact BEA Customer Support through BEA WebSUPPORT at <http://www.bea.com> or by using the contact information provided on the Customer Support Card in the product package.



1 What Is the Commerce Templates Tour?

BEA WebLogic Portal includes a set of commerce templates, included as long as the sample is selected during installation, that you can use to set up your e-commerce Web site quickly. This tour leads you through many of the templates to demonstrate their scope and design. It uses sample data that simulates an e-commerce Web site. An easy way to get started is to follow along with this tour document, working through the suggested steps to get familiar with the templates.

The templates consist of JavaServer Pages (JSP) and image files. You can modify the template layout and the order in which the product presents pages without requiring Java programming skills. Other lower-level modifications, such as changing the default business logic, require Java programming skills.

If you are not familiar with JSP or the basic product concepts, we recommend that you read the remaining topics in this section:

- [About JSP](#)
- [About the Database Schemas](#)

About JSP

JavaServer Pages (JSP) are text files that contain well-formed HTML and special tags (JSP tags) that, when served by a Java-enabled Web server, extend the capabilities of HTML. With Campaign services and Commerce services, you use JSP tags to provide dynamic content in specific areas of a Web page and to create an interface for exchanging data with your customers.

For more information about JSPs, refer to <http://java.sun.com/products/jsp/index.html>.

About the Database Schemas

Commerce services include schemas for both PointBase and Oracle databases. The schemas support all transactions that are available from the Commerce services JSP templates.

Note: When you install WebLogic Server, it sets up an evaluation copy of the PointBase database. If you want to use a PointBase database for your production Web site, you must purchase your own copy of the PointBase database. If you want to use an Oracle database, you must purchase and install an Oracle platform that WebLogic Portal supports.

About the Sample Database

By default, WebLogic Portal is configured to use the PointBase database. When you install WebLogic Portal, it creates and populates a sample PointBase database. The sample data simulates an e-commerce Web site that sells tools and miscellaneous hardware.

About Tax Calculation

To calculate the amount of sales tax due, WebLogic Portal provides an API that lets you integrate with a tax service vendor. If you want to use the tax API for your production Web site, you must also set up your own account with a tax service vendor. With the account you can download updated data files, which contain information about tax requirements for all tax jurisdictions in the United States and selected other countries.

About Payment Processing

To verify and post credit card transactions, WebLogic Portal provides a payment API that lets you integrate with a payment vendor. If you want to use the payment API for your production Web site, you must also set up your own account with a payment vendor. By default, the integration with a payment vendor is disabled. You can complete purchases with the integration disabled, but WebLogic Portal does not send data to a payment vendor.

1 *What Is the Commerce Templates Tour?*

2 Starting the Tour

Before you can start the tour, you or your system administrator must install WebLogic Portal.

To see the templates and sample data from a Web browser, do one of the following tasks depending on the type of operating system you use:

- [Prerequisite for Netscape 4.7x Users](#)
- [Starting the Commerce Server and Templates](#)

Then go to the last topic in this section:

- [Overview of the Home Page](#)

Prerequisite for Netscape 4.7x Users

The minimum required Netscape browser version is 4.77. If you are going to run the tour in the Netscape 4.7x browser, read this section. There are a few prerequisite steps. These steps do not apply if you are using the Microsoft Internet Explorer browser. This section also does not apply if you are using Netscape 6.0 and later versions.

The sample JSP templates do not work with cookies enabled for Netscape in the default configuration for the product. Before starting the tour, you must set the `CookieDomain` value in the Web application's `weblogic.xml` file and uncomment it.

The application's `weblogic.xml` file includes the following:

```
<!--  
<session-param>  
  <param-name>CookieDomain</param-name>  
  <param-value>.mycompany.com</param-value>
```

```
</session-param>
-->
```

Before using the Web application in Netscape, remove the begin and end comment characters and change the `CookieDomain` parameter value in `weblogic.xml` from `.mycompany.com` to the domain name in which the application will be run.

The `weblogic.xml` file resides in the following directory:

```
<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/WEB-INF/
(UNIX)
```

```
<BEA_HOME>\weblogic700\samples\portal\wlcsDomain\beaApps\wlcsApp\wlcs\WEB-INF\
(Windows)
```

Important: After you make the change, you must restart the server if it is already running. For more information about starting the server and then opening the initial page of the sample application, see the next section.

Notes: Another option, of course, is disabling cookies in Netscape 4.77 or later in order to run the tour. However this approach is not practical in a live Web production environment because you cannot predict which browser the Web site's customers will use, or whether they will have cookies enabled or disabled. Therefore in production you must make the `CookieDomain` change in your Web application's `weblogic.xml` file.

In this chapter and in all of the product documentation, the variable `<BEA_HOME>` is used to represent the parent directory into which you install BEA products. The variable `<BEA_HOME>` is used to represent the top-level of the product installation directory.

Hereafter, the tour uses UNIX syntax for pathnames. If you use WebLogic Portal from a Windows machine, substitute the syntax that Windows requires for pathnames (forward slashes separating pathname elements and percent signs before and after variable names).

Starting the Commerce Server and Templates

Before you begin the tour, make sure the following are installed:

- WebLogic Server 7.0
- WebLogic Portal 7.0, including Portal Examples

To begin the tour you need to first start the server and then launch the Commerce Server.

1. Start WebLogic Portal as follows:

Windows: Start Menu → Programs → BEA WebLogic Platform 7.0 → WebLogic Portal 7.0 → Portal Examples → Commerce Templates → Launch Commerce Server

UNIX: Run `<install directory>/weblogic700/samples/portal/wlcsDomain/startWLCS.sh`

2. After WebLogic Portal is running, start the Commerce Templates as follows:

Windows: Start Menu → Programs → BEA WebLogic Platform 7.0 → WebLogic Portal 7.0 → Portal Examples → Commerce Templates → Start Commerce Templates

UNIX: Log on to the Commerce Templates client. If you installed WebLogic Portal with the default settings, you can use the following URL in a browser that is invoked on the same machine as the server:

`http://localhost:7501/wlcs`

where localhost is the name of the server running the Commerce Server.

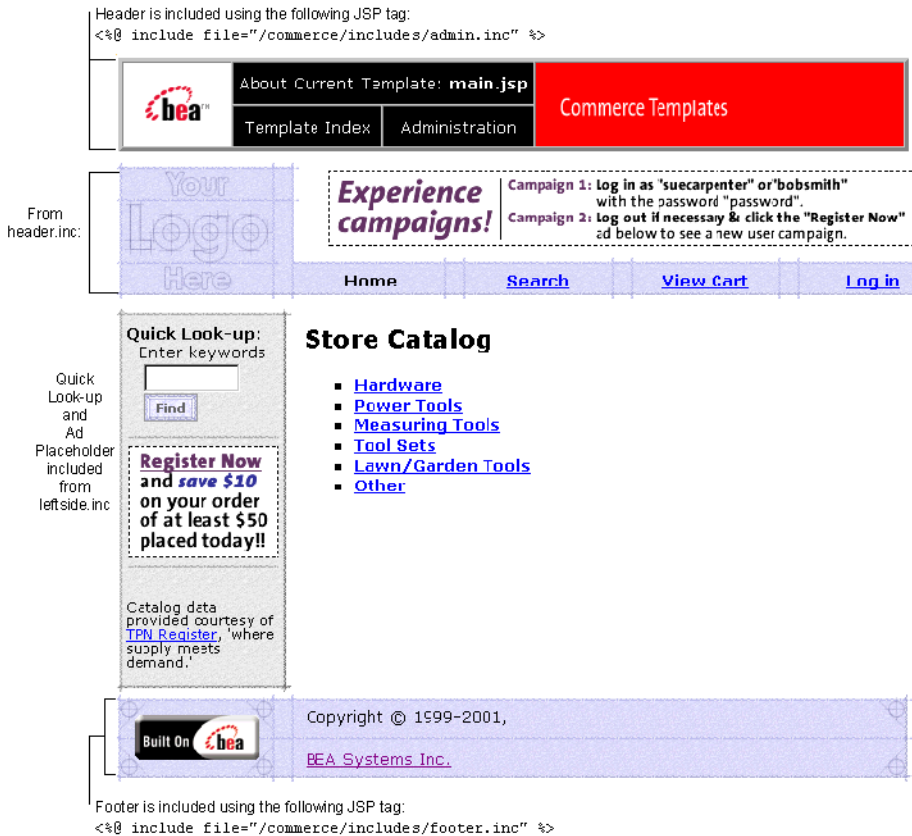
You are now ready to take a tour.

Overview of the Home Page

When you access `http://<WLCS-host:7501>/wlcs`, your browser displays `main.jsp`, which is the Web site's Home page. [Figure 2-1](#) illustrates the information and navigation options that are available to anonymous users. [Chapter 4, “Registered User Buys a Product,”](#) illustrates how `main.jsp` changes for authenticated users.

Features of main.jsp for Anonymous Users (Part 1 of 2)

Figure 2-1 Home Page for Anonymous Users



Notice the following features of main.jsp:

- The file's full pathname is
`<BEA_HOME>\weblogic700\samples\portal\wlcsDomain\beaApps\wlcsApp\wlcs\commerce\main.jsp`
- The top banner, the navigation header, and the footer are defined in separate JSP include files. JSP tags specify that WebLogic Portal includes these files when your browser requests main.jsp.

- In the `admin.inc` include file that appears at the top of `main.jsp`:
 - It displays the JSP filename in the upper banner. This display comes from an included file, `admin.inc`. All templates show the JSP filename in this location.
 - It includes online help, which describes the purpose of the template, navigation options that are available in different user contexts, and notes on customizing the template.
 - Administration links to tools for setting up and maintaining WebLogic Portal features. This tour does not describe the administration tools; to access them, you need the WebLogic Portal administrator username and password. For more information, see “Creating and Administering a Catalog” in the *Administration Guide* at <http://edocs.bea.com/wlp/docs70/admin/commerce.htm#1167188> and “Setting Up Commerce Services” in the *Development Guide* at <http://edocs.bea.com/wlp/docs70/dev/commerce.htm>.
 - Template Index. Links to `templateindex.html`, which briefly describes all JSP templates included in WebLogic Portal.
- The included `header.inc` file contains an ad placeholder. In the `main.jsp` file before it is rendered by the browser, a JSP tag is used to reserve a place for the banner ad. For example:

```
<td colspan="7" height="75" align="center" valign="middle">
<ph:placeholder name="<%= banner %>" />
</td>
```

`header.inc` also includes buttons for navigating to different areas of the site:

- Search links to `search.jsp`, which supports searching with Boolean expressions.
- View Cart links to `shoppingcart.jsp`, which keeps a list of items purchased during the current session.
- Log in links to `login.jsp`, which allows either an existing user to log in, or gives the new user a chance to register with the site.

Technical Note: Importing Tag Libraries and Java Packages

Each JSP template that uses JSP tags begins with a set of lines that import tag libraries. [Listing 2-1](#) shows `main.jsp` importing Pipeline, Catalog (cat), and User Management (um) tag libraries.

Listing 2-1 Importing Tag Libraries

```
<%@ taglib uri="webflow.tld" prefix="webflow" %>
<%@ taglib uri="cat.tld" prefix="catalog" %>
<%@ taglib uri="um.tld" prefix="um" %>
<%@ taglib uri="pz.tld" prefix="pz" %>
<%@ taglib uri="es.tld" prefix="es" %>
<%@ taglib uri="cm.tld" prefix="cm" %>
```

Likewise, templates that use Java classes must import Java packages. [Listing 2-2](#) shows `main.jsp` importing Java packages.

Listing 2-2 Importing Java Packages

```
<%@ page import="com.beasys.commerce.webflow.HttpRequestConstants" %>
<%@ page import="com.beasys.commerce.webflow.PipelineSessionConstants" %>
<%@ page import="com.beasys.commerce.ebusiness.catalog.Category" %>
<%@ page import="com.beasys.commerce.ebusiness.catalog.ViewIterator" %>
<%--@ page import="com.beasys.commerce.axiom.contact.*" --%>
<%--@ page import="com.beasys.commerce.ebusiness.customer.*" --%>

<%@ page import="com.bea.pl3n.content.ContentHelper"%>
<%--@ page import="java.util.Collection" --%>
<%--@ page import="java.util.Iterator" --%>
```

More Features of main.jsp for Anonymous Users (Part 2 of 2)

Figure 2-2 Top Level of the Product Catalog



main.jsp is also the top level of the product catalog. Notice the following features of the catalog's top level:

- JSP tags create a search mechanism in the left column.
- The information in the Store Catalog section is dynamically generated, based on the information in the product catalog and the user's current location in the catalog hierarchy.

Technical Note: The Product Catalog

WebLogic Portal provides a database schema and set of build scripts for implementing a product catalog. The schema and scripts define commonly used features and attributes found on Web-based catalog sites. For example, the schema specifies metadata that you use to establish a hierarchy of product categories and items. You can use this hierarchy to display product data based on a user's current location in the catalog. [Figure 2-2](#) shows the content that WebLogic Portal generates for the top level of the hierarchy.

[Listing 2-3](#) shows the JSP tags in `main.jsp` that query the product catalog for categories and display the query results. The listing also shows other HTML elements for formatting and displaying content. (If you are not familiar with JSP syntax, look only at the comment tags, which are surrounded by `<!--` and `-->`.)

Listing 2-3 JSP Tags that Query the Product Catalog

```
<webflow:getProperty id="topCategory"
property="<%=PipelineSessionConstants.CATALOG_CATEGORY%>"
type="com.beasys.commerce.ebusiness.catalog.Category" scope="request"
namespace="sampleapp_main" />

<webflow:getProperty id="subcategories"
property="<%=PipelineSessionConstants.CATALOG_CATEGORIES%>"
type="com.beasys.commerce.ebusiness.catalog.ViewIterator" scope="request"
namespace="sampleapp_main" />

<p class="head1">Store Catalog</p>

<ul type="square">

<!-- Declare a String containing common browse parameters -->

<!-- static final String commonParameters =
HttpRequestConstants.CATALOG_SOURCE_KEY + "=" +
PipelineSessionConstants.CATALOG_CATEGORIES + "&" +
HttpRequestConstants.CATALOG_DESTINATION_KEY + "=wlcs_siblings&"; %>

<catalog:iterateViewIterator iterator="<%= subcategories %>"
id="currentCategory"
returnType="com.beasys.commerce.ebusiness.catalog.Category">

<% String browseParameters = commonParameters +
HttpRequestConstants.CATALOG_CATEGORY_ID + "=" +
java.net.URLEncoder.encode(currentCategory.getKey().getIdentifier()); %>
```

2 Starting the Tour

```
<li>
    <a href="<webflow:createWebflowURL event="link.browse"
namespace="sampleapp_main" extraParams="<%=browseParameters%>" />">
    <b><catalog:getProperty object="<%= currentCategory %>"
propertyName="Name" /></b>
</a>
<br>
</catalog:iterateViewIterator>
```

WebLogic Portal also includes a utility for importing data from an existing product catalog database, or you can use other utilities to import data. If you already use a catalog system that you prefer to the WebLogic Portal database, you can modify the JSP templates to use your existing catalog system. (However, this low-level modification requires Java programming skills.)

To continue the tour...

Leave `main.jsp` open in your browser and go to [Chapter 3, “New User Buys a Product.”](#) This section shows the JSP templates that new users are likely to encounter.

Other sections of the tour, [Chapter 4, “Registered User Buys a Product,”](#) and [Chapter 5, “Registered User Adds a Credit Card,”](#) show additional scenarios for using the Commerce services JSP templates.

3 New User Buys a Product

In this section of the tour, assume that you want to buy a hammer for a home improvement project. You have heard of a new e-commerce Web site with a huge selection and low prices, so you decide to visit it. Your trek through the Web site takes you along the following path:

Note: Consider opening the New User Roadmap link in a separate browser window; you may want to refer to it frequently during the tour.

- [New User Roadmap](#)
- [Step 1: Search for Products](#)
- [Step 2: Select an Item](#)
- [Step 3: View the Shopping Cart](#)
- [Step 4: Create a New User Profile](#)
- [Step 5: View Successful Login](#)
- [Step 6: Choose Shipping Options](#)
- [Step 7: Enter Payment Information](#)
- [Step 8: Select a Credit Card](#)
- [Step 9: Purchase the Item](#)
- [Step 10: Review the Order Confirmation](#)

New User Roadmap

Before starting this section of the tour, review the following maps of the JSP templates (Figure 3-1, Figure 3-2, and Figure 3-3). The arrows in the maps represent navigation options on the Web site; the darker arrows show the path that this section of the tour describes.

Figure 3-1 JSPs for Finding and Selecting an Item

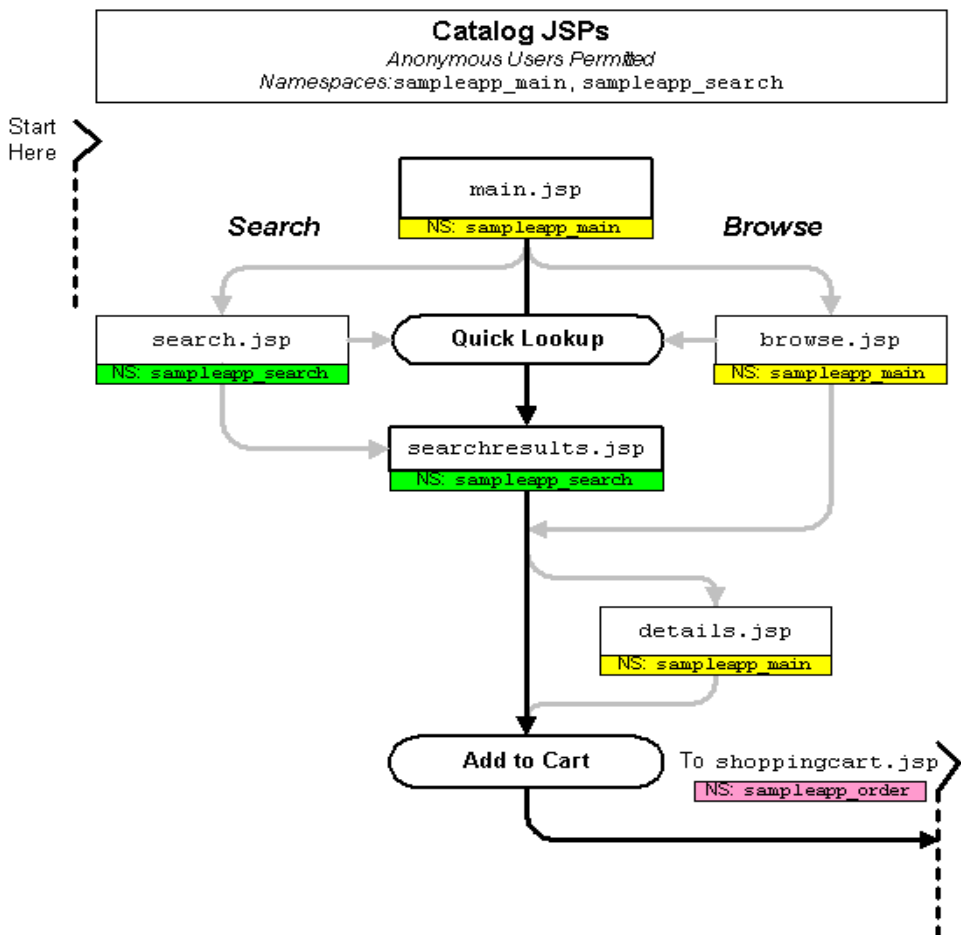


Figure 3-2 JSPs for the Shopping Cart and New User Registration

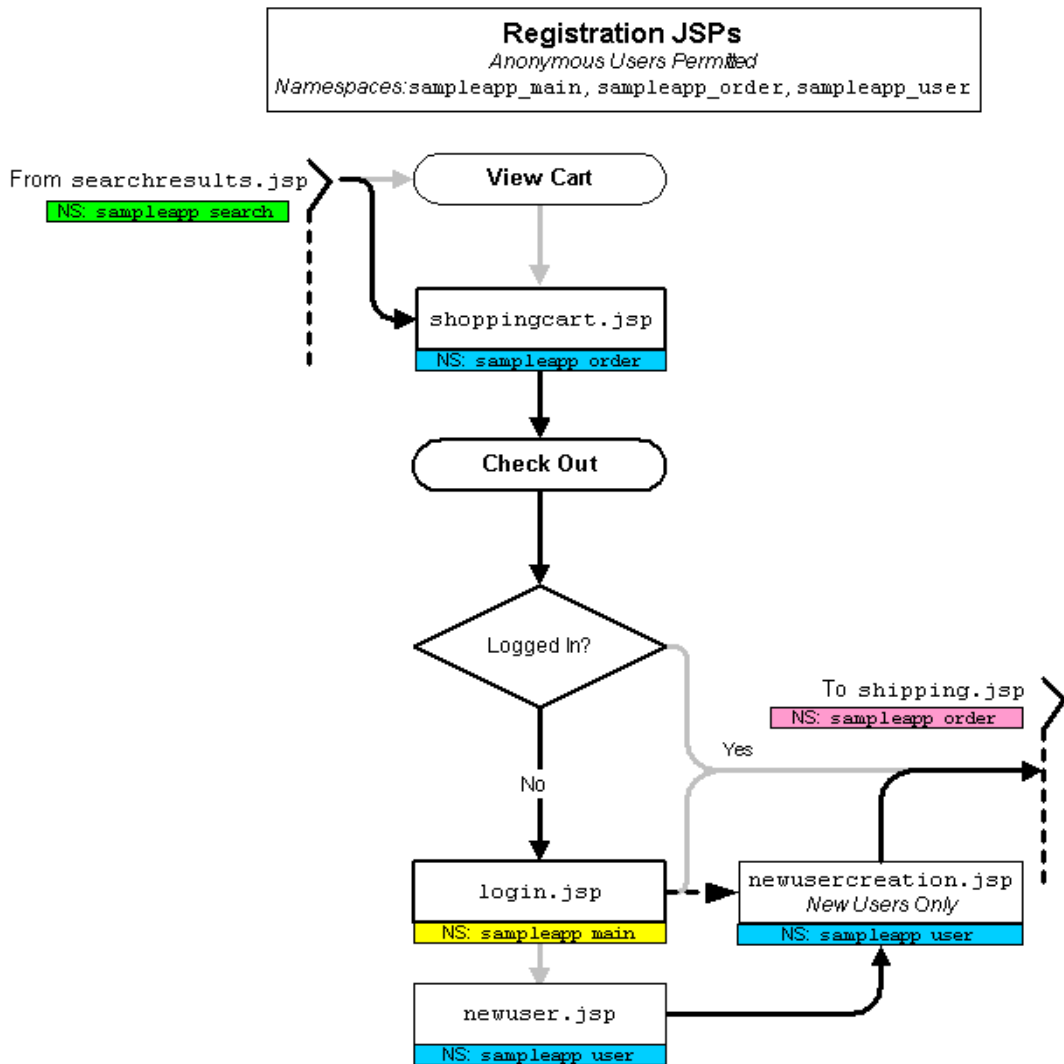
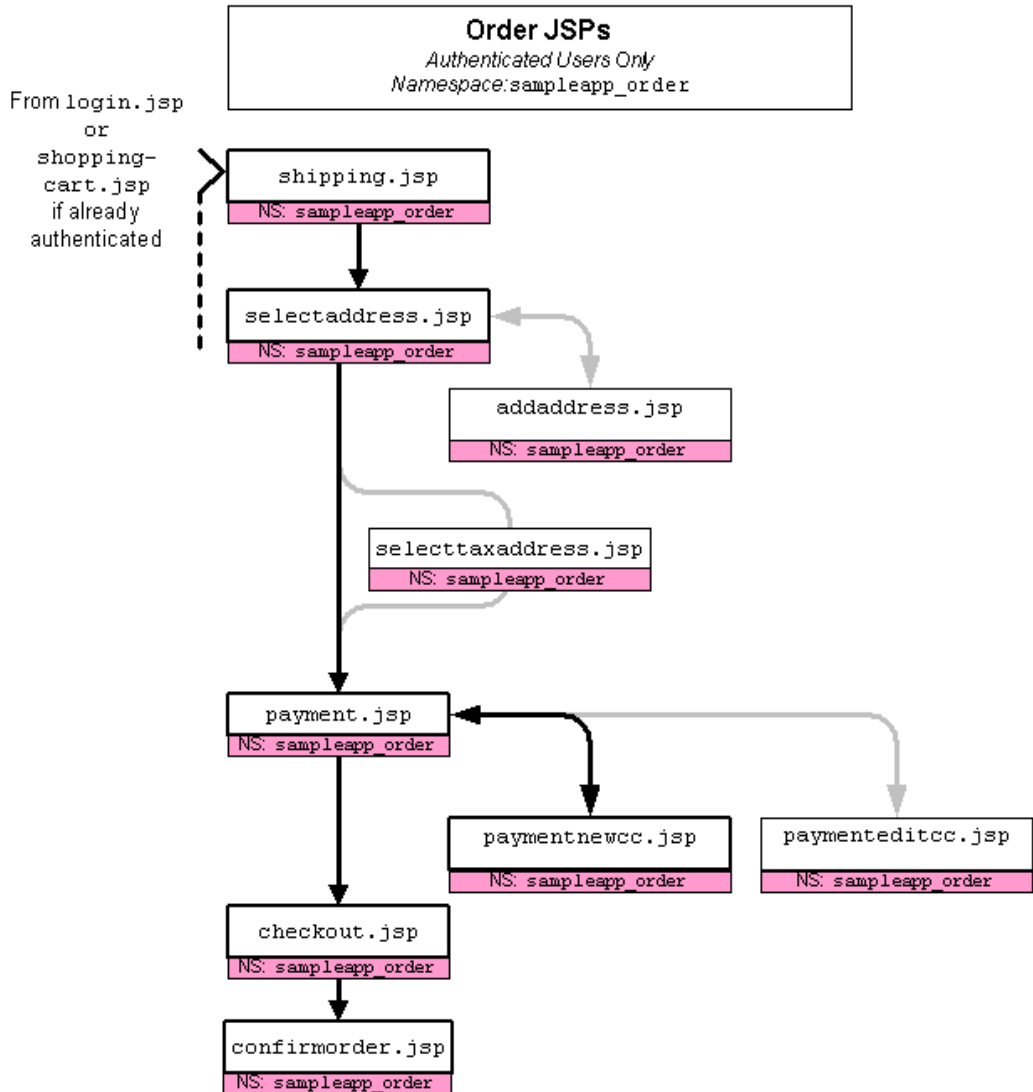


Figure 3-3 JSPs for Completing the Order



Step 1: Search for Products

When you first access the Web site's Home page, you notice that you could either browse or search through the product catalog. Because you are looking specifically for a hammer, and because you are unfamiliar with the scope of this new Web site, you decide to use the Quick Look-up box (see [Figure 3-4](#)).

Figure 3-4 Searching in main.jsp



To continue the tour...

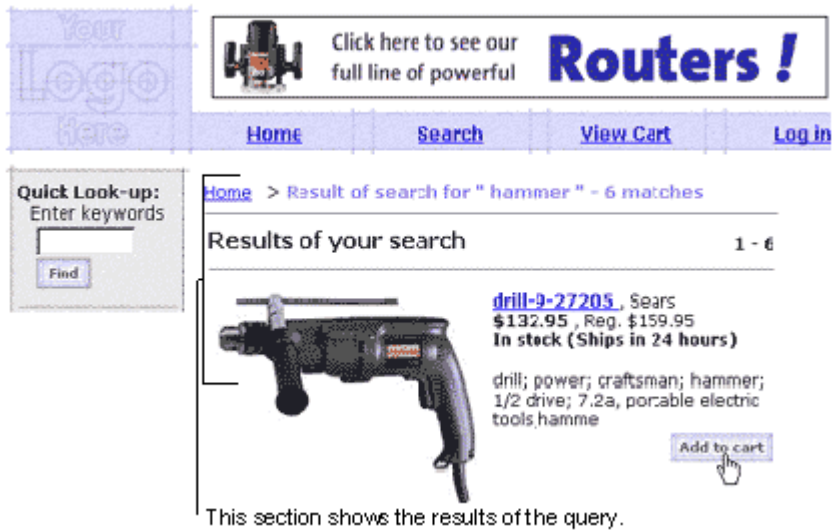
On the Home page, enter hammer in the Quick Look-up box. Then click **Find**.

Step 2: Select an Item

When you click Find, the WebLogic Portal search engine queries the product catalog for items with keywords that match hammer and displays the results in `searchresults.jsp` (see [Figure 3-5](#)).

Features of searchresults.jsp

Figure 3-5 Search Results in searchresults.jsp



Notice the following features of `searchresults.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/catalog/searchresults.jsp` (UNIX)
- All of the content is dynamically generated. The item's description, price, availability, and URL to its related image come from the item record in the product catalog.

To continue the tour...

Choose a hammer and click **Add to Cart**.

Step 3: View the Shopping Cart

When you click Add to Cart, WebLogic Portal lists all items you have selected during the current visit to the Web site in `shoppingcart.jsp` (see [Figure 3-6](#)).

Features of shoppingcart.jsp

Figure 3-6 Shopping Cart

Shopping Cart

Please review the items in your cart before clicking Check Out. Click Delete to remove an item from the cart altogether. Change an amount in the Quantity column to order two or more of an item, then click Update Totals before clicking Check Out.

Empty cart Check out >

Quantity	Item	List Price	Our Price	You Save	Subtotal
1	drill-9-27205	\$ 159.95	\$ 132.95	\$ 27.00	\$ 132.95
					Remove
Total					\$ 132.95
(before shipping and taxes)					

Static HTML.

Generated with JSP tags and scriptlets.

Notice the following features of `shoppingcart.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/shoppingcart.jsp` (UNIX)
- It contains a section of static text that is formatted as HTML.

- The list of items, prices, the subtotal, and all cart-related buttons are dynamically generated based on information in the current Pipeline session.

Technical Note: Pipeline Components, Pipeline Sessions, and Webflow

So far this tour has focused on HTML elements and JSP tags, which WebLogic Portal uses to present information in a format that Web browsers can display.

Behind this presentation layer, Pipeline components, Pipeline sessions, and Webflow exchange data with users, apply business logic and process data, maintain state in individual shopping sessions, and direct your users through the Web site based on a set of context-specific rules. The following list describes each of the mechanisms:

- *Pipeline components*: Discrete units of server-side business logic, such as logic for calculating tax or committing an order. A succession of Pipeline components, in which one component feeds data to another component, is called a Pipeline. The `namespace.pln` file controls the order of Pipeline components in the Pipeline.
- *Pipeline sessions*: Storage locations for information about the current session (such as the current shopping cart) or more transient data (such as error messages about a customer's most recent input).
- *Webflow*: Controls the order of JSPs that WebLogic Portal presents to users based on the current state of the Pipeline session. WebLogic Portal uses a `namespace.wf` file to determine the sequence of pages on your Web site. Webflow centralizes control of Web site navigation, making it easy for you to maintain and update the Web site.

For example, [Listing 3-1](#) shows JSP tags from `shoppingcart.jsp` that retrieve the contents of the shopping cart from the current Pipeline session.

Listing 3-1 JSP Tags for Retrieving Shopping Cart Content

```
<webflow:getProperty id="shoppingCart"
property="<%=PipelineSessionConstants.SHOPPING_CART%>"
type="com.beasys.commerce.ebusiness.shoppingcart.ShoppingCart" scope="session"
namespace="sampleapp_main" />

<webflow:getProperty id="savedShoppingCart"
property="<%=PipelineSessionConstants.SAVED_SHOPPING_CART%>"
```

```
type="com.beasys.commerce.ebusiness.shoppingcart.ShoppingCart" scope="session"
namespace="sampleapp_main" />
```

To continue the tour...

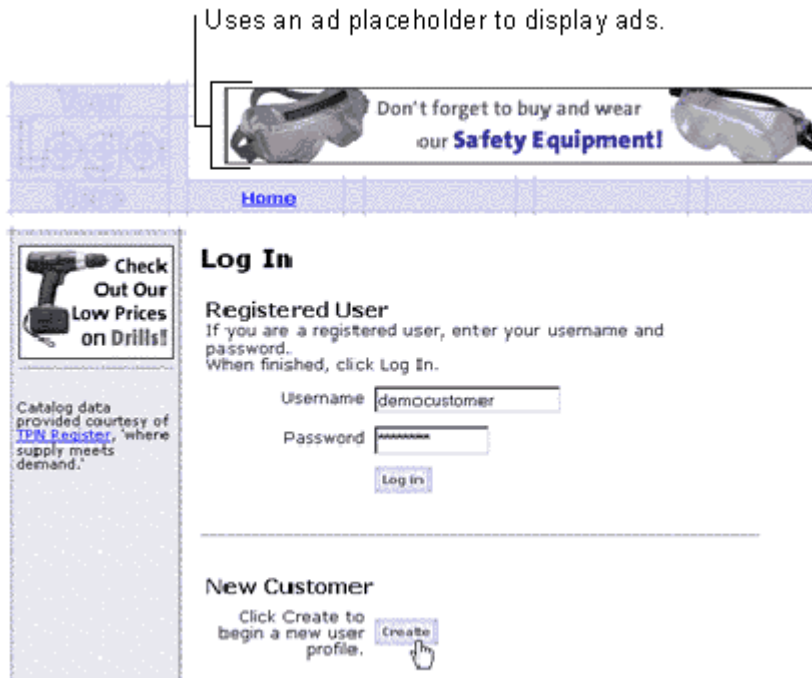
Click **Check Out**.

Step 4: Create a New User Profile

When you click Check Out from the shopping cart, WebLogic Portal prompts you to log in from `login.jsp`. Because you are a new user, you must first create a user profile (see [Figure 3-7](#)).

Features of login.jsp

Figure 3-7 New User Logging In



- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/login.jsp (UNIX)`
- It uses an ad placeholder to display ads in the header. A Power User can specify which types of ads display to specific types of customers in a given ad placeholder.
- To facilitate browsing the sample templates, it specifies default values in the Username and Password boxes. The sample database includes a user profile for the democustomer username.

To continue the tour...

Click **Create** to see the templates for creating a new user profile.

When you click Create on the Log In page, WebLogic Portal prompts you to provide information for your user profile in `newuser.jsp` (see [Figure 3-8](#) and [Figure 3-10](#)).

Features of `newuser.jsp` (Part 1 of 3)

Figure 3-8 Entering Address Information

Register Now
and **save \$10**
on your order
of at least \$50
placed today!!

Catalog data
provided courtesy of
[TPN Register](#), 'where
supply meets
demand.'

Create a New Account

If you already have an account, use your browser's Back button to return to the Login page and log in from there.

First name

Middle initial

Last name

Address

Address 2

City

State / Province (Required for U.S. and Canadian address)

Zip/Postal Code

Country

Home phone

Business phone

Email address

☒ Yes, I want to be offered specials and notified of discounts. Please send me promotional e-mail.

Shipping Address

Same as above ☒

*) Indicates a required field, which is validated by an input processor.

List of states comes from `states.inc`.

List of countries comes from `countries.inc`.

Allows scenario actions to generate email for a customer.

Notice the following features of the first half of `newuser.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/register/newuser.jsp` (UNIX)
- It uses an input processor to determine if the user completed all required fields (the next topic in this section, [Technical Note: Input Processors](#), describes input processors).
- It contains fields for entering a contact address and a shipping address. Users can enter additional shipping addresses from `selectaddress.jsp` or from `viewprofile.jsp` (see [Figure 5-2](#)).
- It includes the following files to provide values for the State and Country lists:
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/includes/states.inc` and
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/includes/countries.inc`

The State and Country lists are defined in separate include files because they are used by several JSP templates.
- It includes a check box for which a customer can specify a preference for receiving email. If you use Campaign services, you can create scenario actions that use JSPs to generate personalized email.

Technical Note: Input Processors

Input processors are Java classes that handle form submission. Some may perform validation of customer data, but the primary role of an input processor is to store customer data into the Pipeline session for subsequent use by a Pipeline component. Webflow determines whether user input is directed to an input processor and where the input processor directs its output.

For example, [Listing 3-2](#) is an excerpt from `newuser.jsp` specifying that when a user clicks the **Save** button, the browser posts data to the Webflow.

Listing 3-2 Form that Sends Data to an Input Processor

```
<!-- begin form to input customer data.
```

```
This form posts to the Webflow Advisor a "button(save)" event from  
origin "newuser.jsp" -->
```

```
<form method="post" action="<webflow:createWebflowURL
event="button.save" httpsInd="calculate"
namespace="sampleapp_user" />">

<input type="hidden" name="origin" value="newuser.jsp">

<input type="hidden" name="event" value="button.save">

<input type="hidden" name="httpsInd" value="calculate">

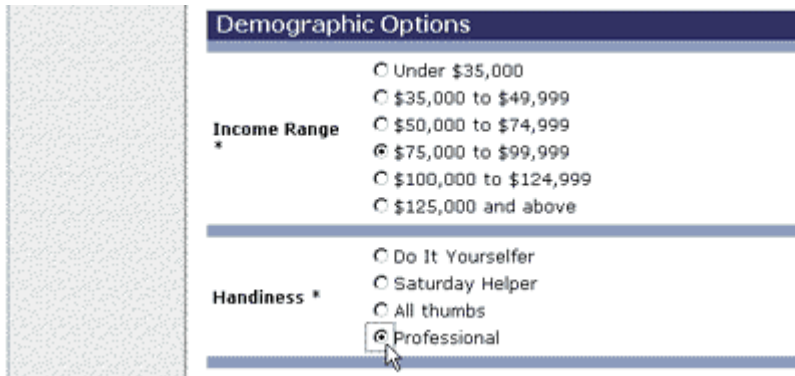
<input type="hidden" name="namespace" value="sampleapp_user">
```

Demographics in newuser.jsp (Part 2 of 3)

`newuser.jsp` collects demographic information about your customers. (See [Figure 3-9](#).) You can use this information to create customer segments, which are used by scenarios and content selectors.

A scenario identifies patterns of events, system states, and customer segments and carries out specific actions. Content selectors to display personalized information to a customer or customer segment.

Figure 3-9 Entering Demographics Information



The screenshot shows a web form titled "Demographic Options". It contains two sections: "Income Range *" and "Handiness *". Each section has a list of radio button options. In the "Income Range" section, the option "\$75,000 to \$99,999" is selected. In the "Handiness" section, the option "Professional" is selected, and a mouse cursor is visible over it.

Demographic Options	
Income Range *	<input type="radio"/> Under \$35,000
	<input type="radio"/> \$35,000 to \$49,999
	<input type="radio"/> \$50,000 to \$74,999
	<input checked="" type="radio"/> \$75,000 to \$99,999
	<input type="radio"/> \$100,000 to \$124,999
	<input type="radio"/> \$125,000 and above
Handiness *	<input type="radio"/> Do It Yourselfer
	<input type="radio"/> Saturday Helper
	<input type="radio"/> All thumbs
	<input checked="" type="radio"/> Professional

More Features of newuser.jsp (Part 3 of 3)

Figure 3-10 Entering username and Password

The screenshot displays a web form titled "newuser.jsp" with two main sections: "Payment Information (Optional)" and "Username and Password".

Payment Information (Optional):

- Credit card type:** A dropdown menu with "SELECT" and an asterisk (*).
- Name on card:** A text input field with an asterisk (*).
- Card number:** A text input field with an asterisk (*).
- Expiration date (mm/yyyy):** Two dropdown menus with "SELECT" and an asterisk (*).
- Card billing address:** A text input field with an asterisk (*).
- Address 2:** A text input field with an asterisk (*).
- City:** A text input field with an asterisk (*).
- State / Province:** A dropdown menu with "SELECT" and a checked checkbox. Text below reads "(Required for U.S. and Canadian addresses)".
- Zip/Postal Code:** A text input field with an asterisk (*).
- Country:** A dropdown menu with "SELECT" and an asterisk (*).

Annotations for the Payment section:

- A line points from the "Credit card type" dropdown to the text: "List values are in newcctemplate.inc."
- A line points from the "State / Province" dropdown to the text: "List values come from include files."

Username and Password:

- Username:** A text input field containing "pet.d.gomes" with an asterisk (*).
- Password:** A text input field with an asterisk (*).
- Repeat password to confirm:** A text input field with an asterisk (*).

Annotations for the Username and Password section:

- A line points from the asterisks on the Username, Password, and Repeat password fields to the text: "The sample JSP templates do not restrict character types in user names and passwords."

At the bottom of the form, there is a note: "Fields marked with (*) are required." followed by "Save" and "Cancel" buttons.

Notice the following features of the second half of `newuser.jsp`:

- It includes `newcctemplate.inc`, which contains fields for new credit card information.

The values for the Credit Card Type and Expiration Date lists are defined in `newcctemplate.inc` because `newcctemplate.inc` is the only template that uses them. The values for the State and Country lists are defined in separate JSP files because they are used by several JSP templates.

- It does not require you to enter payment information to set up your user profile.

- The maximum number of characters allowed for usernames and passwords is set in the JSP file as 50. The sample JSP templates do not impose other restrictions, such as required character types or disallowed character types. If you want to impose such restrictions, you must set up your own field validator, possibly in the form of an input processor.

To continue the tour...

Complete the required fields and click **Save**. The tour automatically logs you in.

Step 5: View Successful Login

Features of successfullogin.jsp

Figure 3-11 successfullogin.jsp



Notice the following features of `successfullogin.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/user/successfullogin.jsp` (UNIX)

- It displays only when a user creates a new user profile, confirming that the user is now registered and authenticated.
- It displays the Checkout button only if there are items in the shopping cart for the current Pipeline session. [Listing 3-3](#) shows the JSP tags that query the Pipeline session and the Java scriptlets that evaluate the results of the query and toggle the Checkout button.

Listing 3-3 Displaying the Checkout Button

```
<!-- Get the shopping cart so we know if checkout is an option -->

<pipeline:getPipelineProperty
propertyName="<%=PipelineSessionConstants.SHOPPING_CART%>"
returnName="shoppingCart"
returnType="com.beasys.commerce.ebusiness.shoppingcart.ShoppingCart"/>

<p>
You have successfully logged in.
</p>
<p>
You may now go to your

<a href="<%=WebflowJSPHelper.createWebflowURL(pageContext,
"successfullogin.jsp",
"link(shoppingcart)", true) %>">shopping cart</a>,

<% if (shoppingCart != null && shoppingCart.isEmpty() == false) { %>
<a href="<%=WebflowJSPHelper.createWebflowURL(pageContext,
"successfullogin.jsp",
"button(checkout)", true) %>">checkout</a>,
<% } %>

or return to the
<a href="<%=WebflowJSPHelper.createWebflowURL(pageContext,
"successfullogin.jsp",
"link(home)", true) %>">main page</a> to continue shopping.
</p>
```

To continue the tour...

Click [Checkout](#).

Step 6: Choose Shipping Options

When you click Checkout, WebLogic Portal starts the checkout process by displaying shipping options in `shipping.jsp` (see [Figure 3-12](#)).

Features of `shipping.jsp`

Figure 3-12 Choose Shipping Options

Shipping

How do you want this order shipped?

☒ Second Day Air

☐ Standard Shipping - 4 to 7 days

Ship all at once, or 'as available'?

☒ Please wait until the entire order is ready before shipping.

☐ Please ship items I ordered as they become available (you may incur additional shipping charges).

Special Instructions

Please leave the package under the doormat.

[< Back](#) [Continue >](#)

Generated by JSP tags.

Static HTML.

Notice the following features of `shipping.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/shipping.jsp` (UNIX)
- It uses Pipelines JSP tags to display the shipping methods that are defined in your database. To enhance performance, WebLogic Portal caches the list of methods instead of accessing the database each time the JSP tags request the data.

- It uses HTML to implement the Ship All At Once or As Available? radio buttons and the Special Instructions input box.

To continue the tour...

1. Select shipping options. You may also enter a comment in the Special Instructions box.
2. Click **Continue**.

When you click Continue, WebLogic Portal displays your shipping addresses in `selectaddress.jsp` (see [Figure 3-13](#)).

Features of `selectaddress.jsp`

Figure 3-13 Select Shipping Address



Notice the following features of `selectaddress.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/selectaddress.jsp` (UNIX)

- It uses Java scriptlets to iterate through the list of shipping addresses in the user profile (see [Listing 3-4](#)).
- It invokes a **Taxation Service**, a WebLogic Portal implementation of business logic and presentation logic.

Listing 3-4 Java Scriptlets in selectaddress.jsp

```
<um:getProperty propertySet="CustomerProperties"
propertyName="shippingAddressMap" id="shippingAddressMap" />

<%
if(shippingAddressMap == null) shippingAddressMap = new HashMap();
%>

<p><font color="red"><webflow:getException/></font></p>
  <!-- Iterate through all addresses -->

  <%
    Iterator iterator = ((Map) shippingAddressMap).keySet().iterator();
    while(iterator.hasNext())
    {
      String addressKey = (String) iterator.next();
      Address shippingAddress = (Address) ((Map) shippingAddressMap).get(addressKey);
    }
  %>

  <table width="90%" border="0" cellpadding="6" cellspacing="0">

    <tr>
      <td align="left" valign="top" width="40%" nowrap>
```

3 *New User Buys a Product*

```
<p><%= shippingAddress.getStreet1() %><br>

<% if( shippingAddress.getStreet2().length() != 0) {%>
<%= shippingAddress.getStreet2() %><br>
<% } %>

<%= shippingAddress.getCity() %><br>
<%= shippingAddress.getState() %> <%= shippingAddress.getPostalCode()
%><br>

<%= shippingAddress.getCountry() %>

</td>

<td align="left" valign="top" width="5%" >
<%
String extraParams = HttpRequestConstants.ADDRESS_KEY + "=" + addressKey;
%>
```

To continue the tour...

Click **Use**.

Step 7: Enter Payment Information

When you click Use, WebLogic Portal displays `payment.jsp`, which lists any credit cards that are in your user profile (see [Figure 3-14](#)).

Features of payment.jsp

Figure 3-14 Select Add Credit Card



Notice the following features of `payment.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/payment.jsp` (UNIX)
- Because you did not add payment information when you set up your user profile, your only option on this page is to add credit card information.

To continue the tour...

Click **Add Card**.

Features of paymentnewcc.jsp

When you click Add Card, WebLogic Portal prompts you to enter payment information in `paymentnewcc.jsp` (see [Figure 3-15](#)).

Figure 3-15 Enter Credit Card Information

Your Logo Here

[Home](#)

Welcome Pat Gomes
[View Profile](#)
[Logout](#)

[View History](#)
[Orders](#)
[Payments](#)

New Credit Card

Credit card type: *

Name on card: *

Card number: *

Expiration date (mm/yyyy): *

Card billing address: *

Address 2:

City: *

State / Province: (Required for U.S. and Canadian addresses)

Zip/Postal Code: *

Country: *

Fields marked with (*) are required.

Adds to user profile.

This section comes from
the following include file:
`/commerce/includes/newcctemplate.inc`

Notice the following features of `paymentnewcc.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/paymentnewcc.jsp` (UNIX)
- This page is similar to
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/user/`

profilenewcc.jsp and

```
<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsAp  
p/wlcs/commerce/register/  
newuser.jsp.
```

All three JSP files use an include file,

```
<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsAp  
p/wlcs/commerce/includes/  
newcctemplate.inc, to define the credit card input fields. For more  
information about template include files, see the next topic in this section,  
“Technical Note: Templates Included Within Templates.”
```

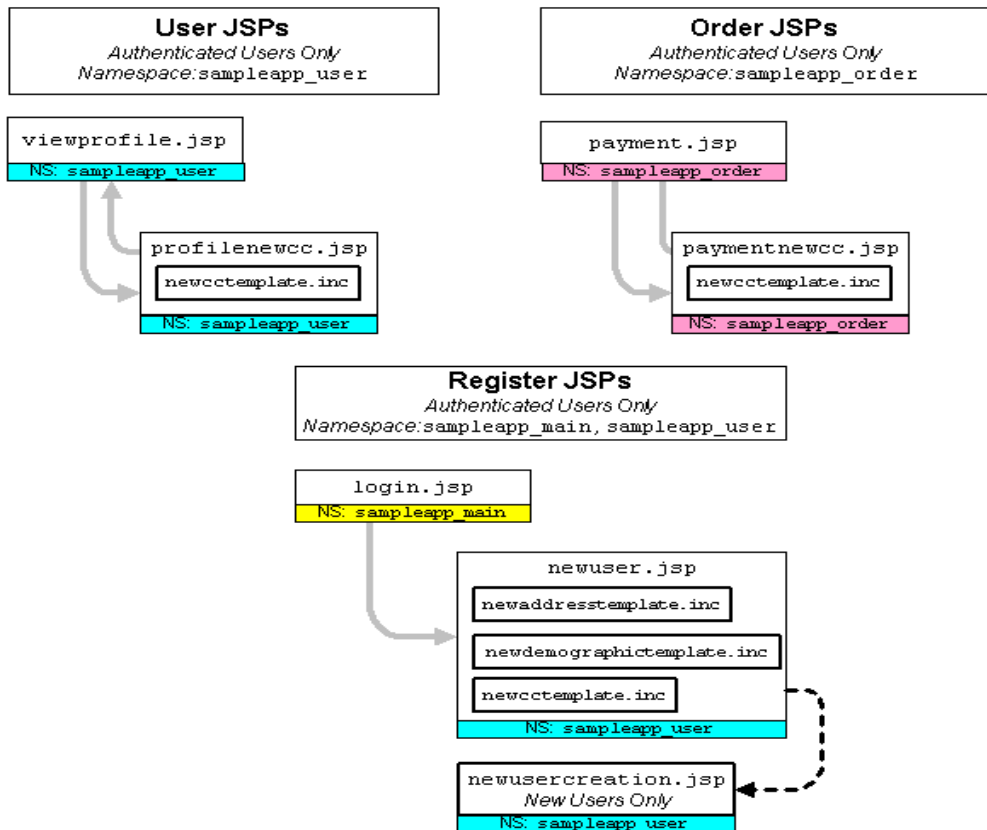
- Clicking **Save** updates the user profile.

Technical Note: Templates Included Within Templates

Rather than repeating input fields and other information across several files, the WebLogic Portal JSP templates assign re-used JSP tags and HTML elements to a set of include files, which are, in turn, JSP templates that you can use to develop your Web site. For example, `paymentnewcc.jsp` (described in the previous section), `profilenewcc.jsp`, and `newuser.jsp` are parent JSP pages that include `newcctemplate.jsp` as a child page to provide consistent input fields for credit card information. You can include `newcctemplate.jsp` in other JSP files that you create.

Webflow uses the three parent JSP pages to determine a user's context and to determine which pages to display after a user enters new credit card information. For example, [Figure 3-16](#) shows that after a user completes `paymentnewcc.jsp`, Webflow displays `payment.jsp`; after a user completes `profilenewcc.jsp`, Webflow displays `viewprofile.jsp`, and after a user completes `newuser.jsp`, `login.jsp` is displayed.

Figure 3-16 Template Include Files



To continue the tour...

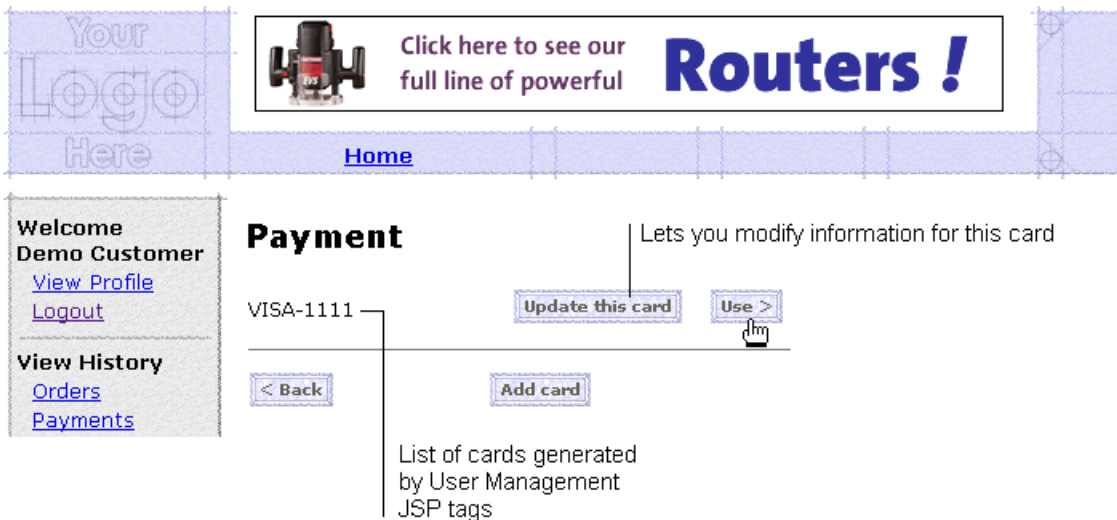
1. Enter information for a credit card.
2. Complete the remaining required fields and click **Save**.

Step 8: Select a Credit Card

When you click Save in `paymentnewcc.jsp`, WebLogic Portal displays your newly added credit card information in `payment.jsp` (see [Figure 3-17](#)).

More Features of `payment.jsp`

Figure 3-17 Select Credit Card



Notice the following features of `payment.jsp`:

- User Management JSP tags generate the credit card list based on your user profile. Earlier in the tour when you accessed `payment.jsp` for the first time (see [Figure 3-14](#)), the User Management JSP tags returned zero credit card definitions from your user profile.

To continue the tour...

Click **Use**.

Step 9: Purchase the Item

When you click **Use** in `payment.jsp`, WebLogic Portal displays `checkout.jsp`, from which you review the tax information and complete the purchase.

Features of checkout.jsp

Figure 3-18 Purchase the Item

Welcome
Pat Gomes
[View Profile](#)
[Logout](#)

View History
[Orders](#)
[Payments](#)

Don't Forget
Extension
Cards!
click here

Catalog data
provided courtesy of
[TPN Register](#), where
supply meets
demand.

Final Checkout Review

Shipping Destination
Pat Gomes
1 Yawkey Way
BOSTON
MA-02110
United States

Shipment Splitting Preferences
Ship all at once

Special Instructions
Please leave the package under the doormat.

Ship Via
Second Day Air

Method of Payment
Credit Card: XXXXXXXXXXXX1111

ID	Description	Quantity	Our Price	SubTotal
9-27205	drill-9-27205	1	\$ 132.95	\$ 132.95

Discount
(Discount of \$ 10 off \$ 50) **\$ -10.00**

Shipping & Handling **\$ 4.95**

Total tax **\$ 6.90**

Total due \$ 134.80

Shows tax information received from _____ the Taxation Service.

[< Back](#) [Complete purchase](#)

Invokes the Payment Service and
Taxation Service.

Notice the following features of checkout.jsp:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsAp
p/wlcs/commerce/order/
checkout.jsp (UNIX)`

- It uses JSP tags to gather and display information from the Pipeline, including the tax information that a **Taxation Service** calculates.
- Clicking **Complete Purchase** invokes a **Payment Service**, a WebLogic Portal implementation of business logic that processes payments.

To continue the tour...

Click **Complete Purchase**.

Step 10: Review the Order Confirmation

When you click Complete Purchase, WebLogic Portal completes the purchase and displays the confirmation number in `confirmorder.jsp` (see [Figure 3-19](#)).

Features of confirmorder.jsp

Figure 3-19 Review the Order Confirmation

Welcome
Pat Gomes
[View Profile](#)
[Logout](#)

View History
[Orders](#)
[Payments](#)

Quick Look-up:
Enter keywords



Catalog data provided courtesy of [TPN Register](#), "where supply meets demand."

Confirm Order

Please print this page for your records.

Thank you for shopping. Don't forget to come back for more great deals, contests, new store openings and specials.

Order Confirmation #1

Will be billed to card:
XXXXXXXXXXXX1111

Will be shipped to:
Pat Gomes
1 Yawkey Way
BOSTON
MA-02110
United States

Shipping Preferences:
Second Day Air

Ship all at once

Please leave the package under the doormat.

ID	Description	Quantity	Unit Price	Subtotal
9-27205	drill-9-27205	1	\$ 132.95	\$ 132.95
Discount (Discount of \$10 off \$50)				\$ -10.00
Shipping & Handling				\$ 4.95
Total Tax				\$ 6.90
Total Billed				\$ 134.80

*** indicates discounts or adjustments associated with a particular item ID.

Notice the following features of `confirmorder.jsp`:

- The file's full pathname is
<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/
confirmorder.jsp
(UNIX)

3 *New User Buys a Product*

- The Pipeline mechanism generates the confirmation number.
- Pipeline JSP tags gather and display the information on this page.

To continue the tour...

This concludes the tour for new users. To continue the tour of the JSP templates, click **Home** in the navigation bar and go to the next section, [Chapter 4, “Registered User Buys a Product.”](#)

4 Registered User Buys a Product

Because you were impressed by the large selection and low prices when you visited the e-commerce Web site, you decide to return and buy a drill for a friend who is starting a home improvement project. Your return trip to the Web site takes you along the following path:

- [Registered User Roadmap](#)

Note: Consider opening the [Registered User Roadmap](#) link in a separate browser window; you may want to refer to it frequently during the tour.

- [Step 1: Browse the Catalog and Select an Item](#)
- [Step 2: Review the Shopping Cart and Check Out](#)
- [Detour: Review the Home Page for Authenticated Users](#)
- [Step 3: Choose Shipping Options](#)
- [Step 4: Enter and Select a New Shipping Address](#)
- [Step 5: Select a Credit Card](#)
- [Step 6: Purchase the Items](#)
- [Step 7: Review Order Confirmation](#)

Note: In this chapter, the product name is referred to as WebLogic Portal. If you are licensed for BEA Campaign services, this product package includes Commerce services.

Registered User Roadmap

Before starting this section of the tour, review the following maps of the JSP templates (Figure 4-1, Figure 4-2, and Figure 4-3). The arrows in the maps represent navigation options that Webflow provides; the darker arrows show the path that this section of the tour describes.

Figure 4-1 JSPs for Browsing the Catalog

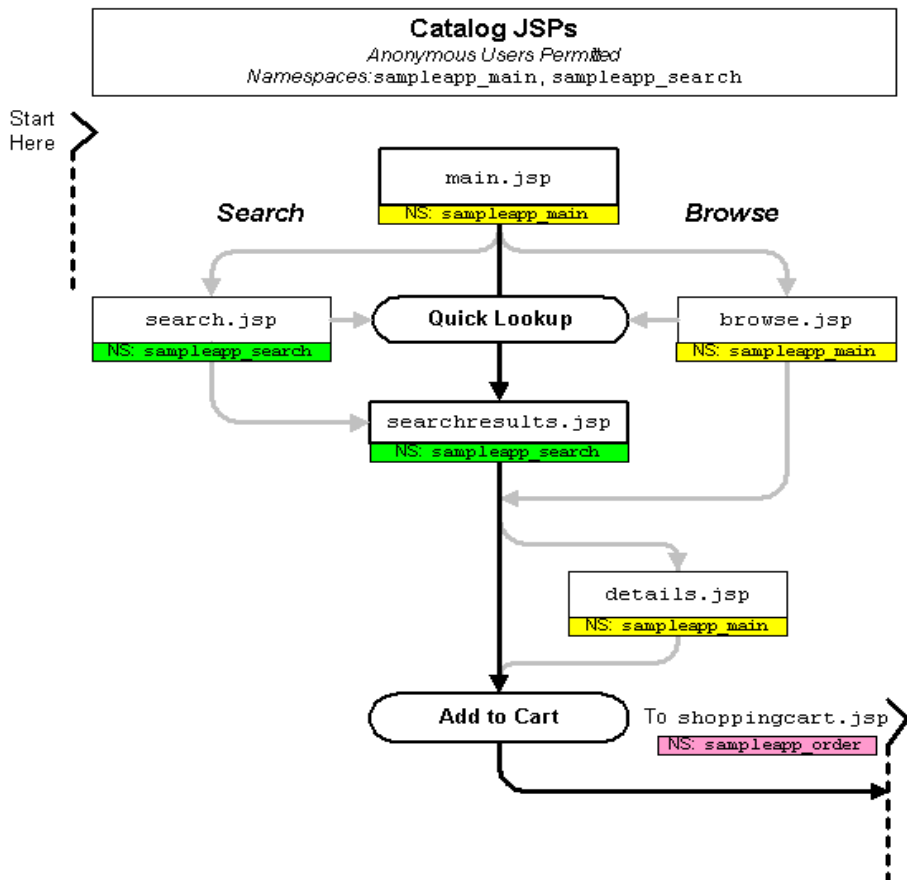


Figure 4-2 JSPs for the Shopping Cart and User Login

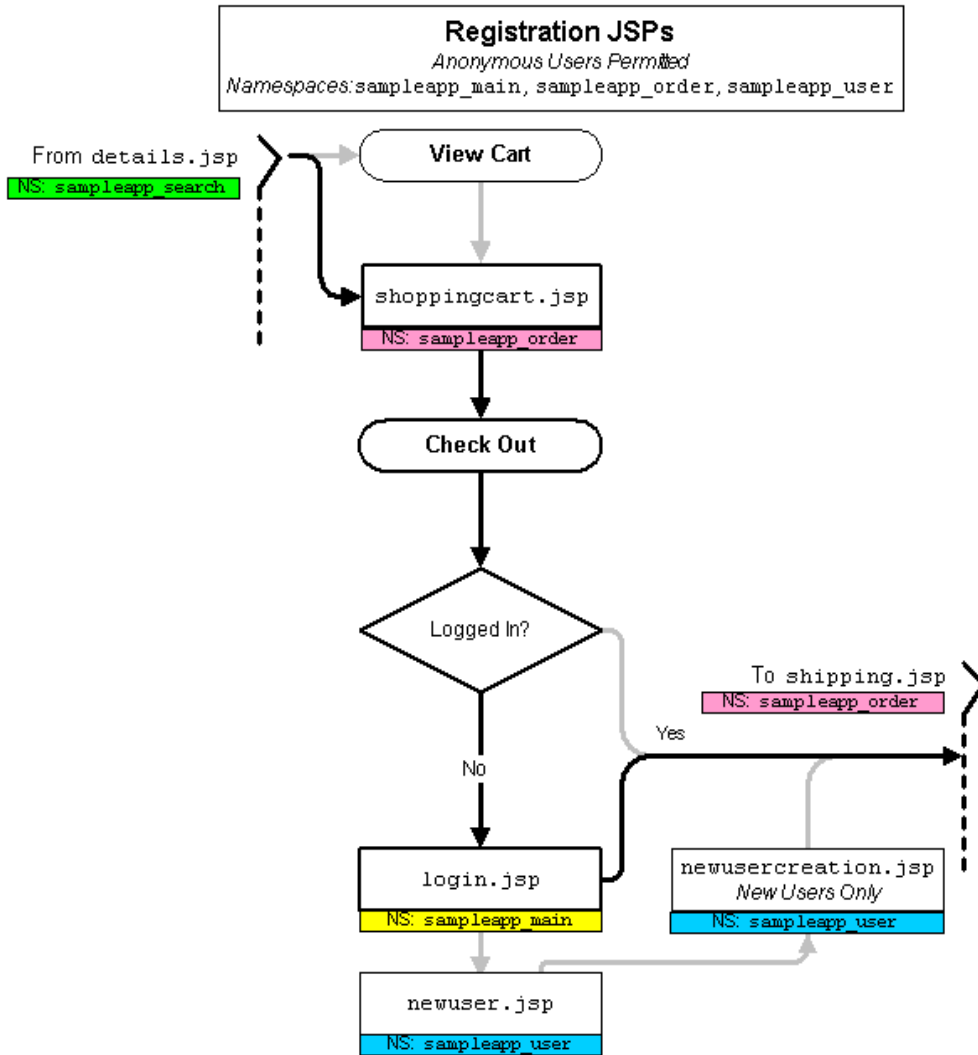
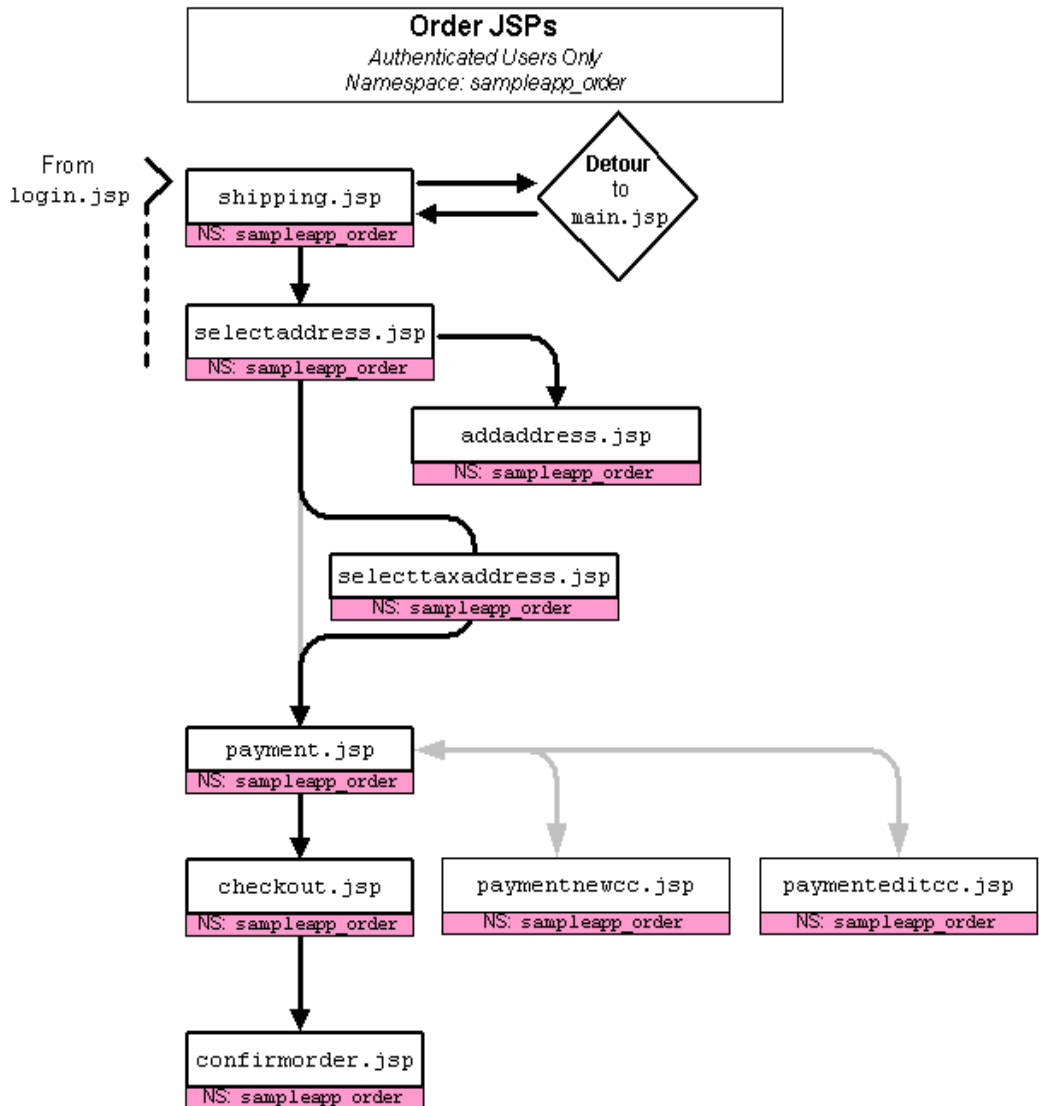


Figure 4-3 JSPs for Completing the Order



Step 1: Browse the Catalog and Select an Item

If you are not still logged in to the tour, log in again. When you arrive at the Home page for your return shopping trip, you decide to browse for drills in the catalog.

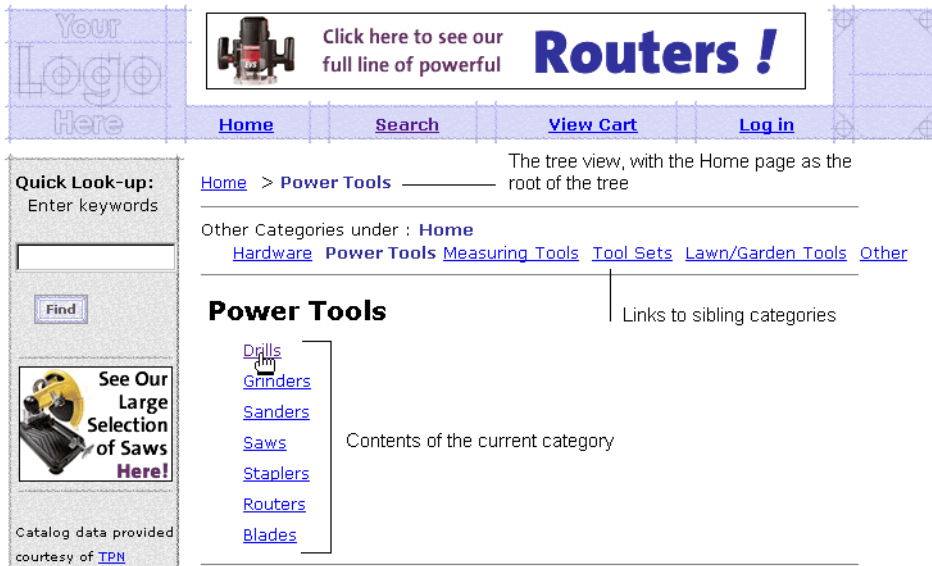
To continue the tour...

Click **Power Tools**.

When you click Power Tools, WebLogic Portal displays the contents of the Power Tools category in `browse.jsp` (see [Figure 4-4](#)).

Features of `browse.jsp` (Part 1 of 2)

Figure 4-4 Second Level of the Catalog



Notice the following features of `browse.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/catalog/browse.jsp` (UNIX)
- Pipeline JSP tags query the product catalog for a list of sibling categories. The name of the current category and links to its siblings are displayed above the category name.
- A JSP tag in the center column constructs a tree view from the current category to the top level of the catalog:

```
<jsp:include page="/commerce/catalog/includes/navigation.jsp"
flush="true"/>
```
- [Listing 4-1](#) shows that `browse.jsp` includes a JSP file that the Catalog generates to describe the contents of the current category.

Listing 4-1 Catalog Tags to Display Current Category

```
<!-- main content -->

<%-- Get the summary JSP from the current category --%>

<catalog:getProperty object="<%= category %>"
propertyName="Jsp"
id="displayJsp"
returnType="com.beasys.commerce.ebusiness.catalog.JspInfo"/>

<%-- Include the category layout JSP for this category --%>
<jsp:include page="<%= displayJsp.getUrl() %>" flush="true"/>
```

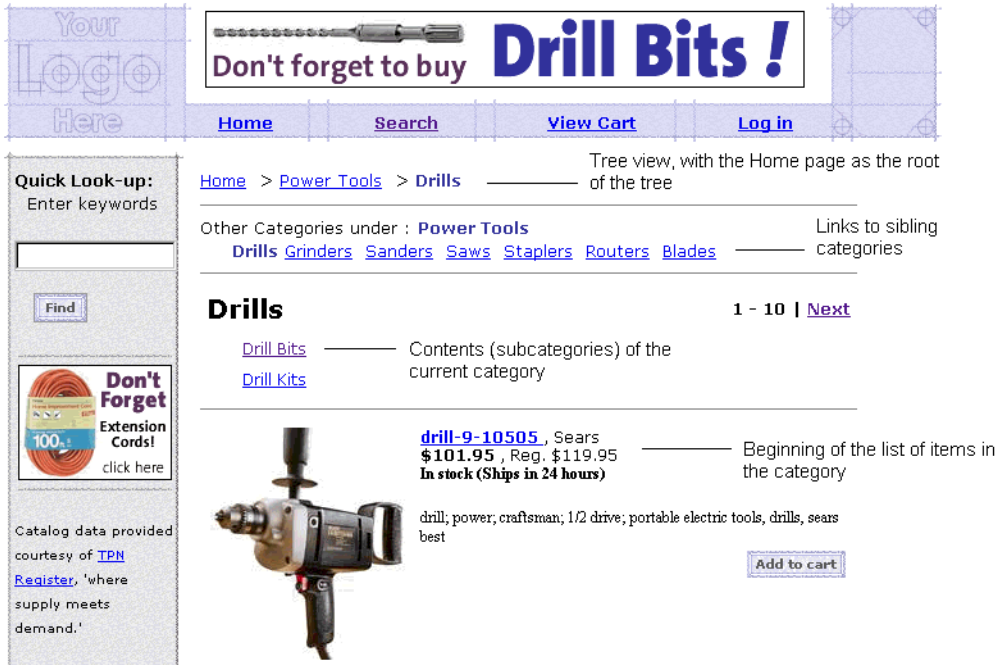
To continue the tour...

Click **Drills**.

When you click Drills, WebLogic Portal displays the contents of the Drills category in `browse.jsp` (see [Figure 4-5](#)).

More Features of `browse.jsp` (Part 2 of 2)

Figure 4-5 Third Level of the Catalog



Notice the following additional features of `browse.jsp`:

- It is the same source file that [Figure 4-4](#) illustrates, but the content is different because the user's context is different.
- The Pipeline JSP tags query the product catalog for the list of sibling categories. Because the page now displays a category different from the category illustrated in [Figure 4-4](#), the list changes.

4 Registered User Buys a Product

- In the center column, the JSP tags modify the tree view to reflect the current category.
- In the second row of the center column, the Catalog generates a different JSP file than in [Figure 4-4](#), hence the contents and layout in [Figure 4-5](#) changes.
- All the items in the drill category are displayed using category and summary JSPs. Each drill includes a link to detailed product information.

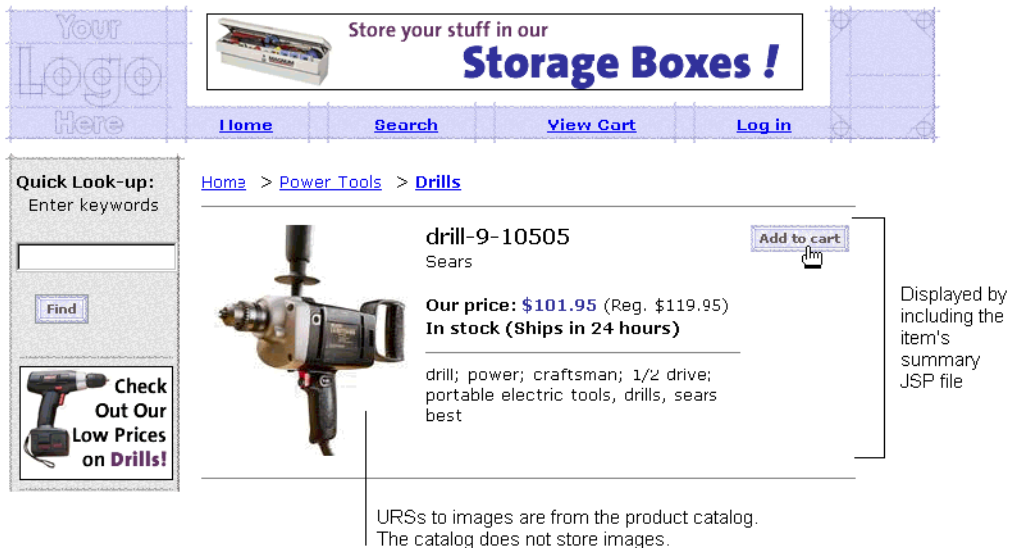
To continue the tour...

Click the name of a drill.

When you click the name of a drill, WebLogic Portal displays a product description in `details.jsp` (see [Figure 4-6](#)).

Features of details.jsp

Figure 4-6 Item Details



Notice the following features of `details.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/catalog/details.jsp` (UNIX)
- It can be accessed from either `browse.jsp` or `searchresults.jsp` (see [Figure 4-1](#)).
- It includes a JSP file (`navigation2.jsp`) that constructs a tree view **only if** users access the page from `browse.jsp`. If the Pipeline mechanism determines that a user accessed the page from `searchresults.jsp`, the JSP file does not construct the tree view.
- It includes a summary JSP file for the item being displayed. The product catalog generates this summary JSP file based on the information it stores about the item.

To continue the tour...

Click **Add to Cart**.

Step 2: Review the Shopping Cart and Check Out

When you click Add to Cart, WebLogic Portal displays your selections in `shoppingcart.jsp` (see [Figure 4-7](#)).

Figure 4-7 Shopping Cart



For a description of `shoppingcart.jsp`, see “Features of `shoppingcart.jsp`” on page 3-7.

To continue the tour...

Click **Check Out**.

When you click Check Out, WebLogic Portal prompts you to log in (see Figure 4-8). Because you created a user profile in Chapter 3, “New User Buys a Product,” you are now a registered user.

Figure 4-8 Log In

Don't Forget
Extension Cords!
click here

Catalog data provided courtesy of [TPN Register](#), 'where supply meets demand.'

Log In

Registered User
If you are a registered user, enter your username and password.
When finished, click Log In.

Username

Password

For a description of `login.jsp`, see [“Features of login.jsp”](#) on page 3-10.

Technical Note: The Security Model

How does WebLogic Portal determine if users must log in when they click a link? Instead of using Webflow to implement security, WebLogic Portal follows the J2EE specification for security. With this specification, the Web site administrator determines which set of files can be accessed only by authenticated users. By default, all JSP templates below the `order` and `user` directories can be accessed only by authenticated users.

When an anonymous user tries to access one of these files, WebLogic Portal displays `login.jsp`, from which the user can either log in or create a new user account. After authentication, Webflow determines the remaining succession of pages.

To continue the tour...

Enter the username and password you set up in “[Step 4: Create a New User Profile](#)” on [page 3-9](#). Then click **Log In**.

When you click Log In, WebLogic Portal displays shipping options in `shipping.jsp` (see [Figure 4-9](#)).

Figure 4-9 Shipping Page Detour



Your Logo Here

Click here to see our full line of powerful **Routers !**

[Home](#)

See Our Large Selection of Saws Here!

Catalog data provided courtesy of [TPN Register](#), 'where supply meets demand.'

Shipping

How do you want this order shipped?

☒ Second Day Air

☐ Standard Shipping - 4 to 7 days

Ship all at once, or 'as available'?

☒ Please wait until the entire order is ready before shipping.

☐ Please ship items I ordered as they become available (you may incur additional shipping charges).

For a description of `shipping.jsp`, see “[Features of shipping.jsp](#)” on [page 3-17](#).

Taking a detour...

Before continuing the tour, take a minute to review the Home page. Now that you have logged in (authenticated), the Home page shows options that are available only to authenticated users.

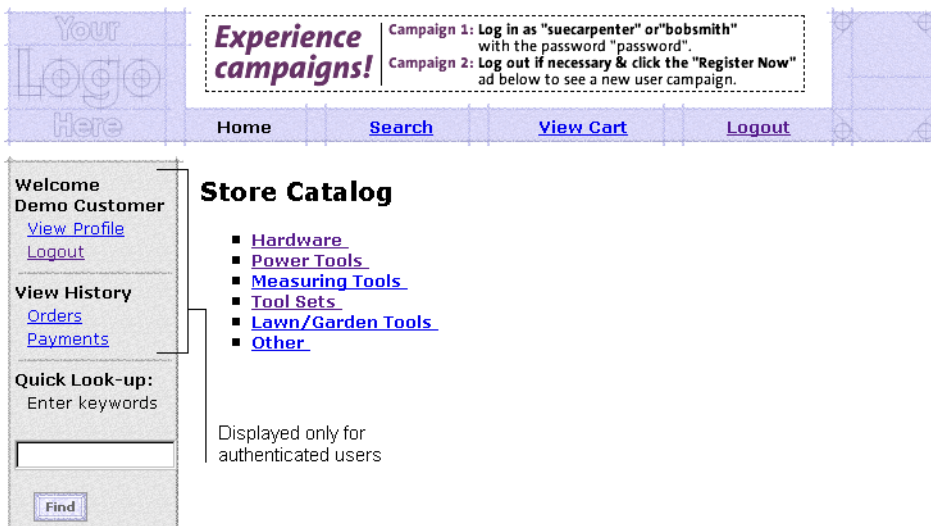
To take the detour, click **Home** (see [Figure 4-9](#)).

Detour: Review the Home Page for Authenticated Users

When you click Home, WebLogic Portal displays `main.jsp` (see [Figure 4-10](#)).

Features of `main.jsp` for Authenticated Users

Figure 4-10 Home Page for Authenticated Users



Notice the following features of `main.jsp`:

- Now that you are authenticated, `main.jsp` displays your username along with links to your order and payment history. If a name is associated with the Pipeline session, User Management JSP tags in the `leftside.inc` file query the database for your username and display the results.

From the `header.inc` file

```
<!-- Get the username --%>
```

4 *Registered User Buys a Product*

```
<pipeline:getPipelineProperty
propertyName="<%= PipelineSessionConstants.USER_NAME %>"
returnName="userName"
returnType="String" />

<!-- if the user is not null, get the customer's profile --%>

<% if(userName != null) { %>

<um:getProfile profileKey="<%=request.getRemoteUser()%>"
profileType="WLCS_Customer" />

<% } %>
```

From the leftside.inc file

```
<!-- Get the username. The customer 's profile should have already been loaded --%>
<!-- obtained from the um:getProfile tag in the header.inc --%>

<pipeline:getPipelineProperty
propertyName="<%= PipelineSessionConstants.USER_NAME %>"
returnName="userName"
returnType="String" />

<% if (userName != null && userName.length() != 0) { %>

<p><b>Welcome<br>

<um:getPropertyAsString
propertyName="firstName" />
<um:getPropertyAsString
propertyName="lastName" /></b></p>

<p>&nbsp;&nbsp;&nbsp;<a href="<%=
WebflowJSPHelper.createWebflowURL(pageContext,
pageContext.getAttribute("AboutFileName") + ".jsp", "link(viewCustomerProfile)",
true) %>">View Profile</a></p>

<p>&nbsp;&nbsp;&nbsp;<a href="<%= WebflowJSPHelper.createWebflowURL(pageContext,
"main.jsp", "link(logout)", true) %>">Logout</a></p>
```

```
<div class="mid" align="center">
" width="120" height="14"
align="absmiddle"></div>

<p><b>View History</b></p>

<p>&nbsp;&nbsp;&nbsp;<a href="<%= WebflowJSPHelper.createWebflowURL(pageContext,
"main.jsp", "link(viewOrderHistory)", true)%>">Orders</a></p>

<p>&nbsp;&nbsp;&nbsp;<a href="<%= WebflowJSPHelper.createWebflowURL(pageContext,
"main.jsp", "link(viewPaymentHistory)", true) %>">Payments</a></p>

<div class="mid" align="center">
" width="120" height="14"
align="absmiddle"></div>

<% } %>

<!-- End of log-in conditional links. --%>
```

To continue the tour...

1. Click **View Cart**.
2. In the shopping cart, click **Check Out**.

Step 3: Choose Shipping Options

When you click Check Out from `shoppingcart.jsp`, WebLogic Portal displays shipping options in `shipping.jsp` (see [Figure 4-9](#)).

For a description of `shipping.jsp`, see [“Features of shipping.jsp” on page 3-17](#).

To continue the tour...

From `shipping.jsp`, click **Continue**.

Step 4: Enter and Select a New Shipping Address

When you click Continue in `shipping.jsp`, WebLogic Portal displays shipping addresses in `selectaddress.jsp`.

To ship the item to someone other than yourself, you must enter a new address.

For a description of `selectaddress.jsp`, see [“Features of selectaddress.jsp” on page 3-18](#).

To continue the tour...

Click **Add Address**.

When you click Add Address, WebLogic Portal prompts you to enter a new shipping address in `addaddress.jsp` (see [Figure 4-11](#)).

Figure 4-11 Enter the New Address

The screenshot shows a web application interface for adding a shipping address. At the top, there is a banner for 'Storage Boxes!' with a 'Home' link. On the left, a sidebar contains a 'Welcome Demo Customer' message with links for 'View Profile', 'Logout', 'View History', 'Orders', and 'Payments'. Below this is an advertisement for 'Check Out Our Low Prices on Drills!' and a note about catalog data provided by TPN Register. The main content area is titled 'Add Shipping Address' and contains the following form fields:

- Street address: 23 Bronco Blvd. *
- Address 2: (empty)
- City: Denver *
- State/Province: CO
- Zip/Postal Code: 80212 *
- Country: United States *

Below the form fields, a note states: 'Fields marked with (*) are required. (State/Province is required for U.S. or Canadian addresses)'. At the bottom of the form, there are two buttons: '< Back' and 'Save'. A mouse cursor is pointing at the 'Save' button.

To continue the tour...

Enter an address that contains the following information:

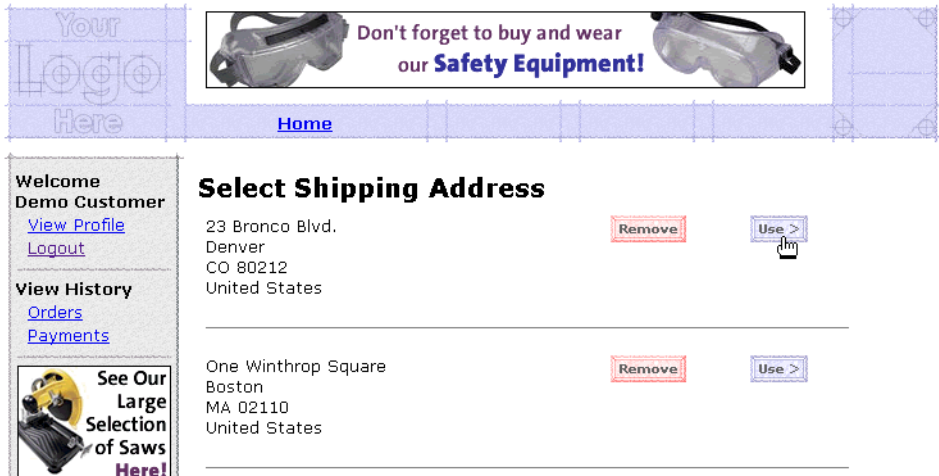
- City: Denver
- Zip Code: 80212
- Country: United States

Then click **Save**.

This particular zip code spans multiple tax jurisdictions. Later in this step of the tour, you must select which tax jurisdiction applies.

When you click Save, WebLogic Portal displays the addresses from your user profile in `selectaddress.jsp` (see [Figure 4-12](#)).

Figure 4-12 Select the New Address



For a description of `selectaddress.jsp`, see “Features of `selectaddress.jsp`” on page 3-18.

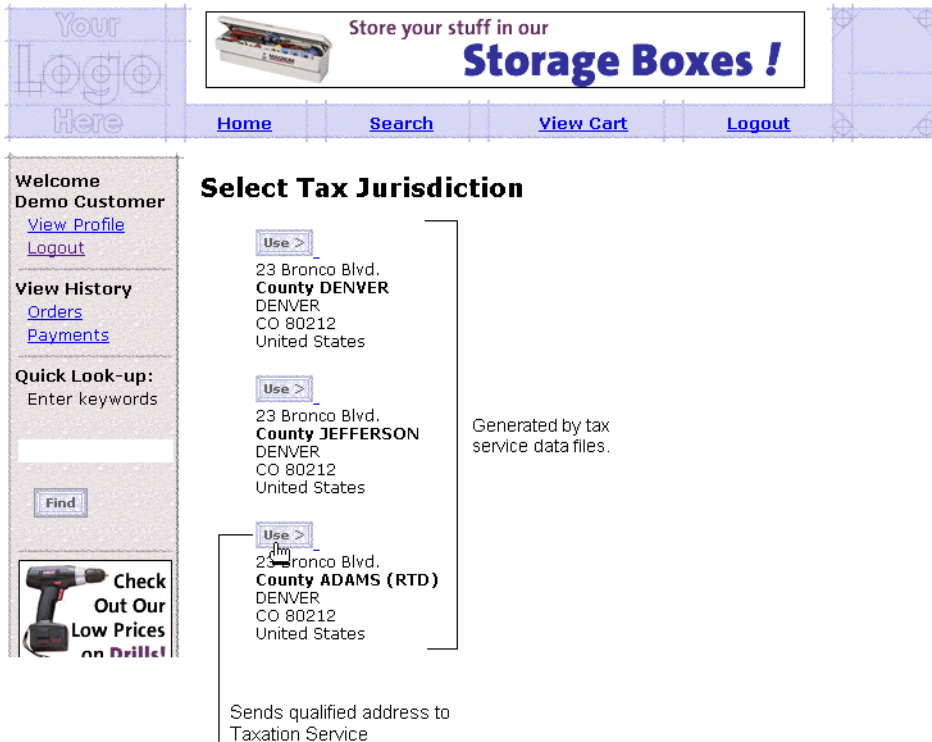
To continue the tour...

Click **Use** next to the Denver, CO address that you created.

When you click Use, the Pipeline sends data to a Taxation Service. Because the zip code 80212 covers different tax jurisdictions, WebLogic Portal prompts you to choose a specific area within the zip code from `selecttaxaddress.jsp` (see Figure 4-13).

Features of selecttaxaddress.jsp

Figure 4-13 Select Tax Jurisdiction



Note the following features of `selecttaxaddress.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/selecttaxaddress.jsp` (UNIX)
- It uses a Pipeline JSP tag to collect the tax jurisdiction data from the Pipeline session.
- It uses Java scriptlets and HTML to display and format the data.

4 Registered User Buys a Product

- Clicking Use sends data to the WebLogic Portal Taxation Service for tax calculation.

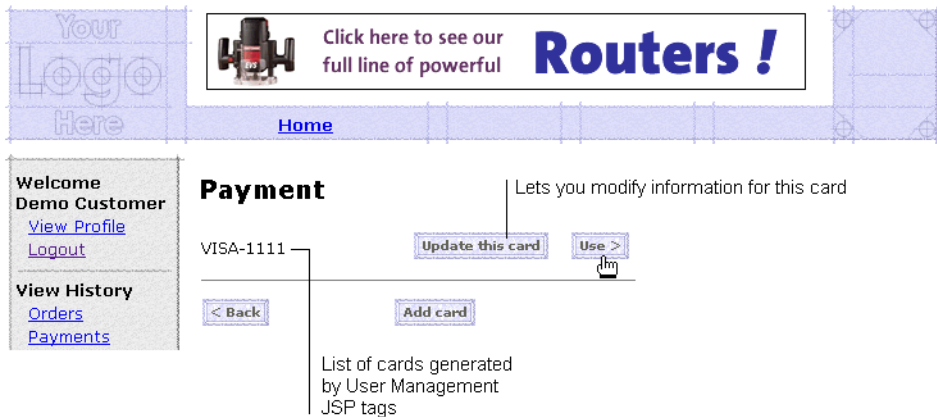
To continue the tour...

Click **Use** next to an address.

Step 5: Select a Credit Card

When you click Use, WebLogic Portal displays `payment.jsp`, which lists the credit cards in your user profile (see [Figure 4-14](#)).

Figure 4-14 Select a Credit Card



For a description of `payment.jsp`, see [“Features of payment.jsp” on page 3-21](#).

To continue the tour...

Click **Use**.

Step 6: Purchase the Items

When you click Use in payment.jsp, WebLogic Portal displays the final purchase information in checkout.jsp (see [Figure 4-15](#)).

Figure 4-15 Purchase the Items

Welcome
Demo Customer
[View Profile](#)
[Logout](#)

View History
[Orders](#)
[Payments](#)

Catalog data provided courtesy of [TPN Register](#), 'where supply meets demand.'

Final Checkout Review

Shipping Destination
Demo Customer
23 Bronco Blvd.
DENVER
CO-80212
United States

Shipment Splitting Preferences
Ship all at once

Special Instructions

Ship Via
Second Day Air

Method of Payment
Credit Card: xxxxxxxxxxxxx1111

Order				
ID	Description	Quantity	Our Price	SubTotal
9-10505	drill-9-10505	1	\$ 101.95	\$ 101.95
Shipping & Handling				\$ 4.95
Shows tax information received from _____ the Taxation Service				Total tax \$ 8.56
				Total due \$ 115.46

[< Back](#)
[Complete purchase >](#)

Invokes the Payment Service and Taxation Service

For a description of checkout.jsp, see [“Features of checkout.jsp” on page 3-27](#).

To continue the tour...

Click **Complete Purchase**.

Step 7: Review Order Confirmation


When you click **Complete Purchase**, WebLogic Portal completes the purchase and displays the confirmation number in `confirmorder.jsp` (see [Figure 4-16](#)).

Figure 4-16 Review the Order Confirmation

Welcome Demo Customer
[View Profile](#)
[Logout](#)

View History
[Orders](#)
[Payments](#)

Quick Look-up:
Enter keywords

See Our Large Selection of Saws Here!


Catalog data provided courtesy of [TPN Register](#), 'where supply meets demand.'

Confirm Order

Please print this page for your records.

Thank you for shopping. Don't forget to come back for more great deals, contests, new store openings and specials.

Order Confirmation #4001

Will be billed to card:
xxxxxxxxxxxx1111

Will be shipped to:
Demo Customer
23 Bronco Blvd.
DENVER
CO-80212
United States

Shipping Preferences:
Second Day Air
Ship all at once

ID	Description	Quantity	Unit Price	Subtotal
9-10505	drill-9-10505	1	\$ 101.95	\$ 101.95
Shipping & Handling				\$ 4.95
Total Tax				\$ 8.56
Total Billed				\$ 115.46

“*” indicates discounts or adjustments associated with a particular item ID.

For a description of `confirmorder.jsp`, see “[Features of confirmorder.jsp](#)” on page 3-29.

To continue the tour...

This concludes the tour for a registered user buying a product.

To continue the tour of the JSP templates, click **View Profile** in the header and go to [Chapter 5, “Registered User Adds a Credit Card.”](#)

5 Registered User Adds a Credit Card

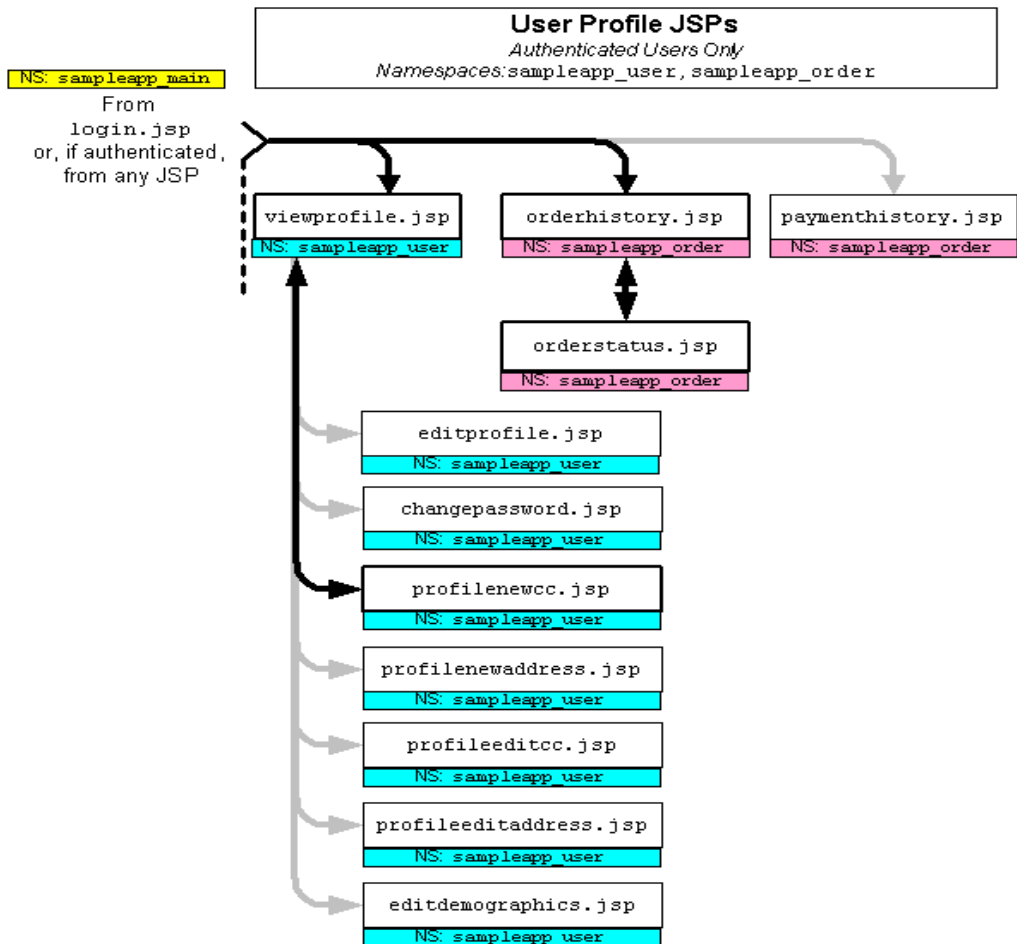
In this section of the tour, assume that you want to access the Web site to add credit card information to your user profile and to see the status of your last order. (This probably is not a realistic scenario for your customers, but it highlights additional features that are available in the JSP templates.) This section of the tour takes you along the following path:

- [User Profile Roadmap](#)
- [Step 1: Access Your User Profile](#)
- [Step 2: Review Your User Profile](#)
- [Step 3: Add a Credit Card](#)
- [Step 4: View Your Order History and Status](#)
- [Step 5: Log Out](#)

User Profile Roadmap

Before starting this section of the tour, review [Figure 5-1](#), which is a map of the user profile JSP templates. The arrows in the map represent navigation options that Webflow provides; the darker arrows show the path that this section of the tour describes.

Figure 5-1 JSPs for Modifying the User Profile



Step 1: Access Your User Profile

You can access your user profile from any page on the Web site by clicking **View Profile** in the left side of the display. This portion of each page comes from the `leftside.inc` include file. If you are accessing the Web site anonymously WebLogic Portal prompts you to log in before it displays your profile.

To continue the tour...

Log in if WebLogic Portal prompts you.

For information about WebLogic Portal security, see “[Technical Note: The Security Model](#)” on page 4-11.

Step 2: Review Your User Profile

When you click View Profile as an authenticated user, or when you click Log In from `login.jsp`, WebLogic Portal displays `viewprofile.jsp` (see [Figure 5-2](#)).

Features of viewprofile.jsp


Figure 5-2 Your User Profile

Secure session using SSL through port 7502



BEA WebLogic Commerce Server Templates - viewprofile - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address https://localhost:7502/wlcs/application/commercewif


 About Current Template: **viewprofile.jsp**
 Commerce Templates


Template Index
 Administration


 Don't forget to buy and wear
our **Safety Equipment!**


[Home](#)

Welcome
Demo Customer
[View Profile](#)
[Logout](#)

View History
[Orders](#)
[Payments](#)


 Check
Out Our
Low Prices
on Drills!

Catalog data
provided courtesy of
[TPN Register](#), where
supply meets
demand.

Demo Customer's Profile

Contact Info

[Update](#)

Name Demo Customer
Address One Winthrop Square
 Boston
 MA 60237
 United States
Home Phone 617-555-5555
Business Phone 708-555-5555
Email Address democustomer@bea.com
OK to E-mail? Yes, I want to be offered specials
 and notified of discounts. Please
 send me promotional e-mail.

Shipping Addresses

[Add address](#)

One Winthrop Square
 Boston
 MA 02110
 United States
[Update](#) [Remove](#) — Added by default in the
 first shopping session.

One Main Street
 Denver
 CO 80212
 United States
[Update](#) [Remove](#) — Added in the second
 shopping session.

Credit Cards

[Add card](#)

Select an account on file to update or delete.

VISA-1111 [Update this card](#) [Remove](#) — Added by default in the
 first shopping session.

User Name & Password

[Update](#)

User Name democustomer

Demographic Data

[Update](#)

Gender Female
Date of Birth October 8, 1948
Occupation Executive Management
Employment Status Employed
Marital Status Married
Education Level College Graduate
Income Range \$75,000 to \$99,999
Handiness Do It Yourselfer

Notice the following features of `viewprofile.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/user/viewprofile.jsp` (UNIX)
- It creates an HTTPS session on port 7502 to send encrypted data through SSL.
- It displays the shipping addresses you added in previous shopping sessions. The first address was added by default when you indicated that the shipping address was the same as your contact address (see [“Step 4: Create a New User Profile” on page 3-9](#)).
- It displays the credit card information you entered in your first shopping session.

To continue the tour...

Click **Add Card**.

Step 3: Add a Credit Card

When you click Add Card, WebLogic Portal prompts you to enter credit card information in `profilenewcc.jsp` (see [Figure 5-3](#)).

Features of profilnewcc.jsp

Figure 5-3 Adding a Credit Card

Secure session using SSL through port 7502

BEA WebLogic Commerce Server Templates - profilnewcc.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <https://localhost:7502/wlp/commerce/user/profilnewcc.jsp> Go Links >>

About Current Template: **profilnewcc.jsp**

Template Index Administration Commerce Templates

Your Logo Here

Store your stuff in our **Storage Boxes!**

[Home](#)

Add Credit Card

Check Out Our Low Prices on Drills!

Catalog data provided courtesy of [TEN Register](#), where supply meets demand.

Credit card type: VISA *

Name on card: Demo Customer *

Card number: 4111111111111111 *

Expiration date (mm/yyyy): 4 2004 *

Card billing address: 4 Fourth Street *

Address 2: *

City: Former *

State / Province: OK (Required for U.S. and Canadian addresses) *

Zip/Postal Code: 76444 *

Country: United States *

Fields marked with (*) are required.

[Save](#)

[< Back](#)

newcctemplate.jsp defines these data-entry fields.

Notice the following features of `profilenewcc.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/user/profilenewcc.jsp` (UNIX)
- A JSP tag includes the file that defines the data-entry fields:
`<%@ include file="/commerce/includes/newcctemplate.jsp" %>`

For information about template include files, see [“Technical Note: Templates Included Within Templates” on page 3-23](#).

To continue the tour...

1. Enter information for a credit card.
2. Complete the remaining required fields. Then click **Save**.

The next part of the tour shows your order history and status, which you access from the Home page for authenticated users.

3. Click **Home** in the header.

Step 4: View Your Order History and Status

When you click Home, WebLogic Portal displays `main.jsp`, which is the Web site's Home page (see [Figure 5-4](#)). Notice that `main.jsp` uses the HTTP protocol through port 7501; WebLogic Portal does not use SSL for the Home page.

Figure 5-4 Click Orders



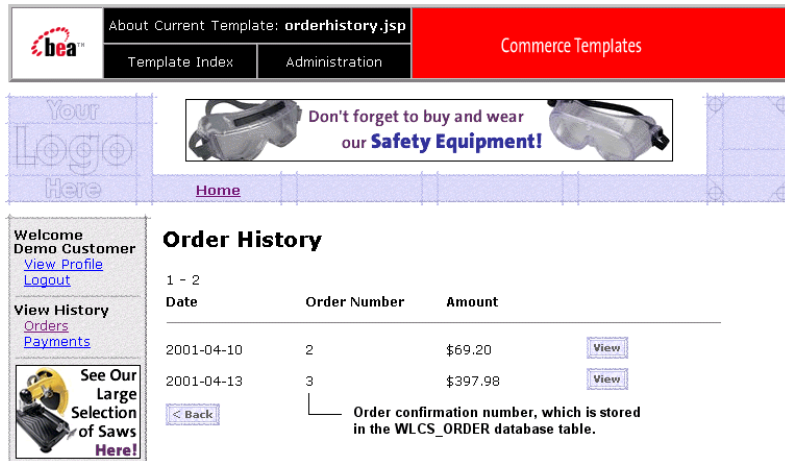
For details about `main.jsp` for authenticated users, see “Features of `main.jsp` for Authenticated Users” on page 4-13.

To continue the tour...

Click **Orders**, WebLogic Portal displays `orderhistory.jsp` (see Figure 5-5).

Features of orderhistory.jsp

Figure 5-5 Order History



Notice the following features of `orderhistory.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/orderhistory.jsp` (UNIX)

- A Pipeline JSP tag retrieves the history.

```
<!-- Extract the scrollable model -->
```

```
<webflow:getProperty id="orderHistory"
property="<%=PipelineSessionConstants.SCROLLABLE_MODEL%"
type="com.beasys.commerce.ebusiness.util.ScrollableModel"
scope="session" namespace="sampleapp_main" />
```

- WebLogic Portal stores this history in the `WLCS_ORDER` database table until the database administrator archives the data.

To continue the tour...

Click **View** to see the order status.

When you click View, WebLogic Portal displays `orderstatus.jsp` (see [Figure 5-6](#)).

Features of orderstatus.jsp

Figure 5-6 Order Status

Order Status

Confirmation number: 2
Order status: Submitted
Date ordered: 2001-04-13
Splitting preferences: Ship all at once
Special instructions:
Shipping address: One Winthrop Square, BOSTON, MA-02110, United States

Order status is stored in the WLCS_ORDER table, along with other data related to the order.

ID	Description	Quantity	Subtotal
71-86393	shears-71-86393	1	\$ 60.95
Shipping & Handling			\$ 4.95
Total tax			\$ 3.30
Total due			\$ 69.20

[< Back](#)

Notice the following features of `orderhistory.jsp`:

- The file's full pathname is
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/orderstatus.jsp` (UNIX)
- A Pipeline JSP tag collects properties of the order:

```
<!-- define bean to retrieve the CustomerValue -->

<webflow:getProperty id="orderValue"
property="<%=PipelineSessionConstants.SELECTED_ORDER%>"
type="com.beasys.commerce.ebusiness.order.OrderValue"
scope="session" namespace="sampleapp_main" />

<%
```

- The order properties, which are stored in the order database table, include the order status.

Note: The Pipeline assigns only the New or Submitted state to an order. If you want to add states, you must modify the Pipeline.

To continue the tour...

Click **Home**.

Step 5: Log Out

To end your authenticated session, click **Logout**. Logging out ends the Pipeline session, clearing your shopping cart and disassociating your username from subsequent events on the Web site (until you log in again).

What's next?

To see an example of how to modify the JSP templates and the Webflow, see [Chapter 6, “Example of Customizing the Templates.”](#)

6 Example of Customizing the Templates

In previous sections of the tour, when you clicked the **Add to Cart** link in the commerce templates, WebLogic Portal displayed `shoppingcart.jsp` to prompt you to check out. However, your business model may want to encourage customers to continue shopping by returning them to the product catalog after they click **Add to Cart**.

You can change the behavior of the Add to Cart link by modifying the Webflow associated with the `wlcs` Web application. Before you modify the Webflow, however, you need some background information. This section of the tour takes you along the following path:

- [Step 1: Understand the Add to Cart Link](#)
- [Step 2: Modify Add to Cart](#)
- [Step 3: Verify the Modification](#)

Step 1: Understand the Add to Cart Link

Instead of specifying a static filename as a URL, link anchors in WebLogic Portal invoke the Webflow mechanism to determine the destination resource. The link anchor for the Add to Cart link (illustrated in [Listing 6-1](#)) is in the include file

```
<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApps  
/wlcs/commerce/catalog/includes/itemdetails.jsp.
```

Listing 6-1 Add to Cart Anchor

```
<%-- Add the 'Put in Cart' link --%>  
  
<%  
  
allParams = HttpRequestConstants.CATALOG_ITEM_SKU + "=" +  
productItem.getKey().getIdentifier();  
  
%>  
  
<td align="right" valign="top" width="5%"><a href="<webflow:createWebflowURL  
event="link.add" namespace="sampleapp_order" extraParams="<%= allParams %>" />">  
  
  " border="0" alt="Add to Shopping  
Cart"></a>  
  
</td>
```

This step of the tour describes the following actions:

- [How the Link Anchor Invokes Webflow](#)
- [How Webflow Resolves the Link](#)

How the Link Anchor Invokes Webflow

The anchor uses the `<webflow:createWebflowURL>` JSP tag to pass data to Webflow.

Note the following three data items that the `<webflow:createWebflowURL>` JSP tag passes to Webflow:

- The event type and name. For the Add to Cart link, the event type is `link` and the name is `add`.
- The namespace name in which the file that contains the link source resides. For the Add to Cart link, the namespace name is `sampleapp_order`.
- Any extra parameters that should be included.

After sending this data through one or more Pipelines, Webflow returns a URL that resolves the link.

How Webflow Resolves the Link

When the `<webflow:createWebflowURL>` JSP tag sends data to Webflow, Webflow searches the `sampleapp_order.wf` for the `link.add` event in the event list of the wildcard presentation node, as shown in [Listing 6-3](#).

Listing 6-2 The Add to Cart Link in `sampleapp_order.wf`

```
<wildcard-presentation-origin node-type="jsp">
  <event-list>

    ...

    <event event-name="link.add">
      <destination namespace="sampleapp_order"
        node-name="addProductItemToShoppingCartIP" node-type="inputprocessor"/>
    </event>

    ...

  </event-list>
</wildcard-presentation-origin>
```

6 Example of Customizing the Templates

The destination shown for the `link.add` event for this case is to invoke the `addProductItemToShoppingCartIP` Input Processor, which is in `sampleapp_order`. Therefore, Webflow locates the `addProductItemToShoppingCartIP` Input Processor in the list of processor node origins, as shown in [Listing 6-3](#).

Listing 6-3 The `addProductItemToShoppingCartIP` Input Processor in `sampleapp_order.wf`

```
<processor-origin node-name="addProductItemToShoppingCartIP"
  node-type="inputprocessor">

  <node-processor-info
    class-name="examples.wlcs.sampleapp.catalog.webflow.GetProductItemIP"/>

  <event-list>
    <event event-name="success">
      <destination namespace="sampleapp_order"
        node-name="addProductItemToShoppingCart" node-type="pipeline"/>
    </event>
  </event-list>

  ...
</processor-origin>
```

If the `addProductItemToShoppingCartIP` Input Processor executes successfully, the Webflow invokes the `addProductItemToShoppingCart` Pipeline. Webflow searches `sampleapp_order` and locates the `addProductItemToShoppingCart` Pipeline, as shown in [Listing 6-4](#).

Listing 6-4 The `addProductItemToShoppingCart` Pipeline in `sampleapp_order.wf`

```
<processor-origin node-name="addProductItemToShoppingCart" node-type="pipeline">
  <node-processor-info pipeline-name="addProductItemToShoppingCart"/>
  <event-list>
    <event event-name="success">
      <destination namespace="sampleapp_order"
        node-name="addProductItemToShoppingCartTracker" node-type="pipeline"/>
    </event>
  </event-list>
</processor-origin>
```

```
</event>
</event-list>

...

</processor-origin>
```

Similarly, if the `addProductItemToShoppingCart` Pipeline executes successfully, it sends data to another Pipeline named `addProductItemToShoppingCartTracker`. If the `addProductItemToShoppingCartTracker` Pipeline is successful, then Webflow invokes the `refreshSavedList` Pipeline. When the `refreshSavedList` Pipeline executes successfully, it reloads the `shoppingcart.jsp`.

Step 2: Modify Add to Cart

After understanding how the Webflow determines the user-visible behavior of the Add to Cart link, you can modify it to direct customers to the top level of the catalog.

Recall from [Chapter 3, “New User Buys a Product,”](#) that the top level of the product catalog is displayed in `main.jsp`. However, changing the `refreshSavedList` Pipeline’s success event to load `main.jsp` as a destination (rather than `shoppingcart.jsp`) would not produce desirable results. This is because `main.jsp` does not contain the standard HTML tags that Web browsers can render. Instead, the file contains JSP tags. If you were to change the output of the `refreshSavedList` Pipeline’s success event to display `main.jsp` directly, the Web browser would display an empty page.

Instead, you must invoke an Input Processor named `getTopCategoriesIP`, which resides in the `sampleapp_main` namespace. This Input Processor gathers data about categories in the product catalog and displays it in `main.jsp`.

Directing the Output of the `getTopCategoriesIP` Input Processor

You can direct output from the `refreshSavedList` Pipeline to the `getTopCategoriesIP` Input Processor using the Webflow and Pipeline Editors.

Open the `sampleapp_order.wf` file using the Editor, then move the destination of the `refreshSavedList` Pipeline Node's success event by dragging the end connector from the `shoppingcart.jsp` Presentation Node to the `sampleapp_main_getTopCategoriesIP` Proxy Node. This will point the event destination to the correct namespace.

Step 3: Verify the Modification

After you save your modifications to `sampleapp_order.wf`, you must synchronize the modifications to the WebLogic Portal server using the E-Business Control Center by choosing Tools → Synchronize in the menu.

After you have synchronized, you can see the results by clicking the **Add to Cart** link from `details.jsp` (see [Figure 6-1](#)). You do not need to restart the WebLogic Portal server.

Figure 6-1 Click Add to Cart in details.jsp



When you click Add to Cart, WebLogic Portal displays the top level of the product catalog in `main.jsp`.

However, to verify that WebLogic Portal added your item to the shopping cart, do the following from `main.jsp`:

1. Click **View Cart** in the header.
2. If WebLogic Portal prompts you to log in, enter the username and password you set up in the previous sections of the tour.

When you click View Cart, WebLogic Portal displays `shoppingcart.jsp`, which contains the item you added to the cart.

What's Next?

To continue the tour and learn about the campaign features that are used in the JSP templates, and how they were defined in the E-Business Control Center, see [Chapter 7, “A Sample Campaign.”](#)

7 A Sample Campaign

In previous sections of the tour, the focus was on the commerce features of the sample Web application. The descriptions were primarily intended as an overview for the Java enterprise developers or corporate developers who use the JSP templates and tags to build WebLogic Portal applications.

This final chapter of the tour switches the focus to the campaign features, especially from the perspective of the power users and marketing professionals who use the BEA E-Business Control Center.

The following topics are discussed:

- [Introduction and Prerequisites](#)
- [Examining a Campaign: TourCampaign1](#)

Introduction and Prerequisites

This tour takes you through a basic sample campaign, and shows you the simple JSP tags that have been inserted in the sample JSP templates. These JSP tags do things like provide placeholders on the Web pages for content that is either associated with a campaign, or operates independently of a campaign. The characteristics of that content is determined by the person who defines the campaigns in the E-Business Control Center, the profiles of people using the Web site, and the events that occur while people are using the Web site.

First, a few basic facts about the BEA E-Business Control Center and campaigns:

- A broad range of users, including power users, marketing professionals, and corporate enterprise developers can use the E-Business Control Center to define

and manage promotional campaigns, campaign discounts, stand-alone discounts, customer segments, content selectors, ad placeholders, and custom events.

- A campaign is a named group of scenarios that work together to achieve strategic business goals. A campaign coordinates several WebLogic Portal services to create and track marketing goals on an e-commerce Web site.
- A scenario is the primary building-block of campaigns. A scenario is composed of one or more scenario actions, all of which take place if their criteria are met. For example, if a particular event occurs and if a customer is part of a specified customer segment.

Examining a Campaign: TourCampaign1

To learn about a basic campaign, take the following steps

1. On a supported client Windows system, use the Start menu to start the E-Business Control Center. For example:

Start → Programs → BEA WebLogic Platform → WebLogic Portal 7.0 →
E-Business Control Center

The E-Business Control Center client displays a splash screen with a number of *Initializing...* messages, and then opens its windows on the desktop.

2. In the E-Business Control Center, choose File → Open Application in the menu.

Note: You must load an application in the E-Business Control Center before you can see sample data.

3. Open the following project:

<BEA_HOME> → weblogic700 → samples → portal → wlcsDomain → beaApps
→ wlcsApp-project → wlcsApp-project.

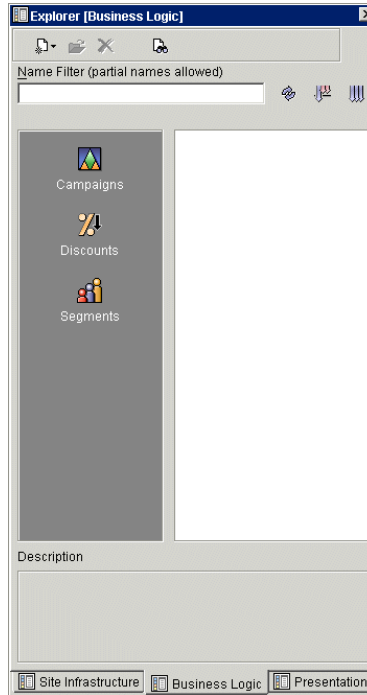
The *wlcsApp* sample application loads, as indicated in the E-Business Control Center's title bar.

This application is a sample enterprise application that ships with WebLogic Portal. You are not loading the enterprise application itself. Rather, you are loading a grouping of XML files that are associated with each tool in the

E-Business Control Center. These files, when synchronized to the WebLogic Portal server, will help control the behavior of the enterprise application.

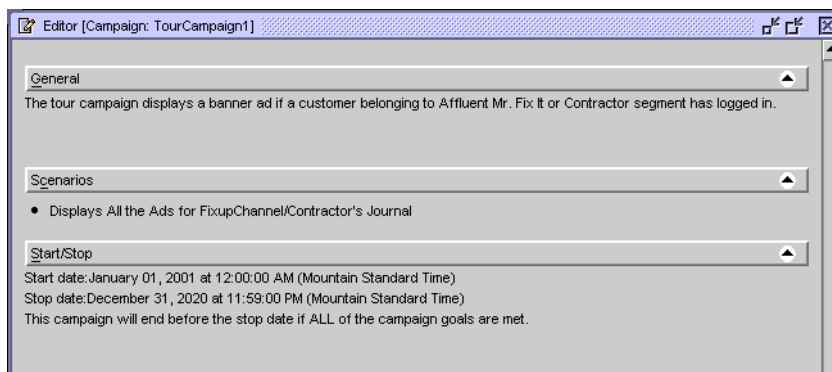
4. In the Explorer window ([Figure 7-1](#)), click the **Business Logic** tab, and then the **Campaigns** icon in the Explorer window.

Figure 7-1 Explorer Window



5. Single-click **TourCampaign1** file. The description explains that in the Web application, a banner ad will be displayed if a customer belonging to the “Affluent Mr/Ms Fix-It” or “Contractor” customer segments has logged in. We will try the login accounts on the sample JSP templates later. For now, though, let’s look at the definitions related to TourCampaign1.
6. Double-click “**TourCampaign1**.” In a few seconds, an editor window is displayed, as shown in [Figure 7-2](#).

Figure 7-2 Campaign Editor Window



7. Click the **Start/Stop** bar.

In the Start/Stop pane, notice the existing start and stop dates for the campaign.

The End Criteria section of the pane is where you can set goals that, when met, will stop the campaign prior to its end date. Goals are either based on ad impressions (when a visitor simply views an ad on a Web page) or ad clickthroughs (when a visitor actually clicks a displayed ad). You determine the number of impressions or clickthroughs required to meet the goal and end the campaign. In TourCampaign1, this goal is set to 25 ad impressions.

8. To get a closer look at the goal setup, click the goal and then click **Edit**. The Goal Detail window is displayed, as shown in [Figure 7-3](#).

Figure 7-3 End Criteria Detail Window

Goal Detail

Goal Type: Ad Impressions

Target number: 25

Target Basis
Target number is based upon the total of:

☒ All ads in the tracking list
☐ Any given ad in the tracking list

Target Scope
Count ad events based on:

☒ Placements from this campaign only
☐ Placements from all campaigns

Specify Ads

Get Ad IDs... Go online to get IDs of ads placed by this campaign or enter IDs directly in the entry field.

Enter ad IDs: [Text Box]

Add

Ads to be tracked:

Ads/wlscsm_ads/tourBannerFixup.jpg
Ads/wlscsm_ads/tourBannerJournal.gif

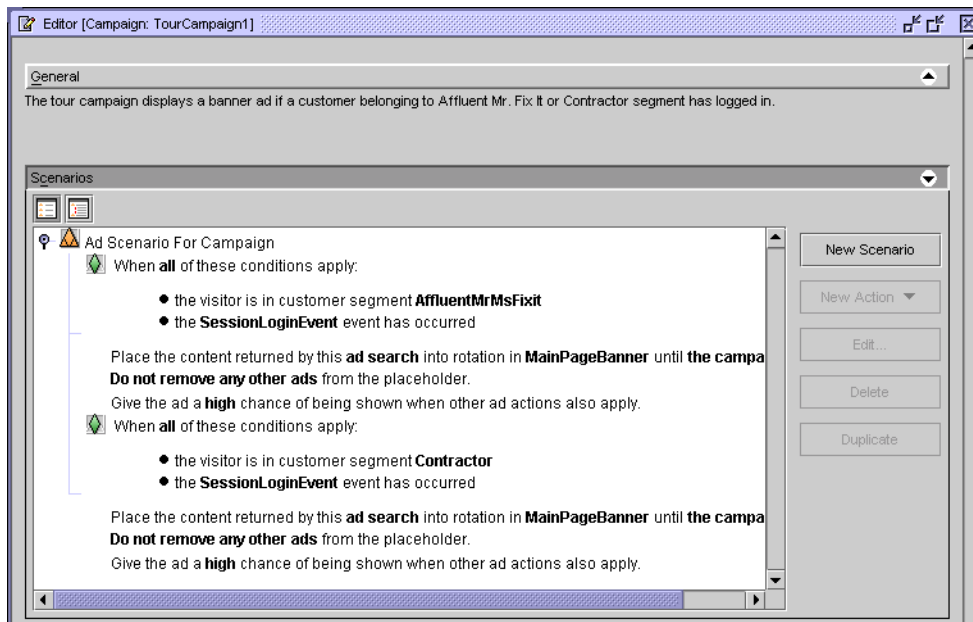
Remove
Preview

OK Cancel

Notice how this particular campaign is specifying two banner ad graphics that will be displayed. On this screen, the power user or marketing professional can set the target number and characteristics for the campaign, the goal scope, and the goal type.

9. Click **OK** or **Cancel** on the End Criteria Detail window. Back on the Editor window for TourCampaign1, click the **Start/Stop** bar again to minimize it.
10. Click the **Scenarios** bar to expand its pane. The Ad Scenario for Campaign is listed in the pane.
11. Click the **Expand Node** button next to the name of the scenario to show the scenario details, as shown in [Figure 7-4](#).

Figure 7-4 Details Display for Ad Scenario for Campaign



Again, a scenario is the primary building-block of campaigns. A scenario is composed of one or more scenario actions, all of which take place if their criteria are met: if a particular event occurs or if a customer is part of a specified customer segment. As you can see, two actions are defined for this scenario. The scenario occurs if:

- The Web site visitor who has just logged in is part of the “Affluent Mr/Ms Fixit” customer segment.
- Or if the Web site visitor who has just logged in is part of the “Contractor” customer segment

As you will recall from [“Demographics in newuser.jsp \(Part 2 of 3\)” on page 3-13](#), the `newuser.jsp` template includes a `newdemographictemplate.inc` file. It contains form fields that allow the person registering with the Web site to characterize themselves in one of several categories. [Figure 7-5](#) shows a portion of the Demographic Options on the `newuser.jsp`:

Figure 7-5 Demographic Options from newuser.jsp Template

Marital Status *	<input type="radio"/> Widowed <input type="radio"/> Married <input type="radio"/> Divorced <input type="radio"/> Single
Education Level *	<input type="radio"/> College Graduate <input type="radio"/> High School <input type="radio"/> Graduate Degree <input type="radio"/> Professional Degree <input type="radio"/> Some College
Income Range *	<input type="radio"/> Under \$35,000 <input type="radio"/> \$35,000 to \$49,999 <input type="radio"/> \$50,000 to \$74,999 <input type="radio"/> \$75,000 to \$99,999 <input type="radio"/> \$100,000 to \$124,999 <input type="radio"/> \$125,000 and above
Handiness *	<input type="radio"/> Do It Yourselfer <input type="radio"/> Saturday Helper <input type="radio"/> All thumbs <input type="radio"/> Professional

For the sample campaigns, there are two registered customers you need to know about:

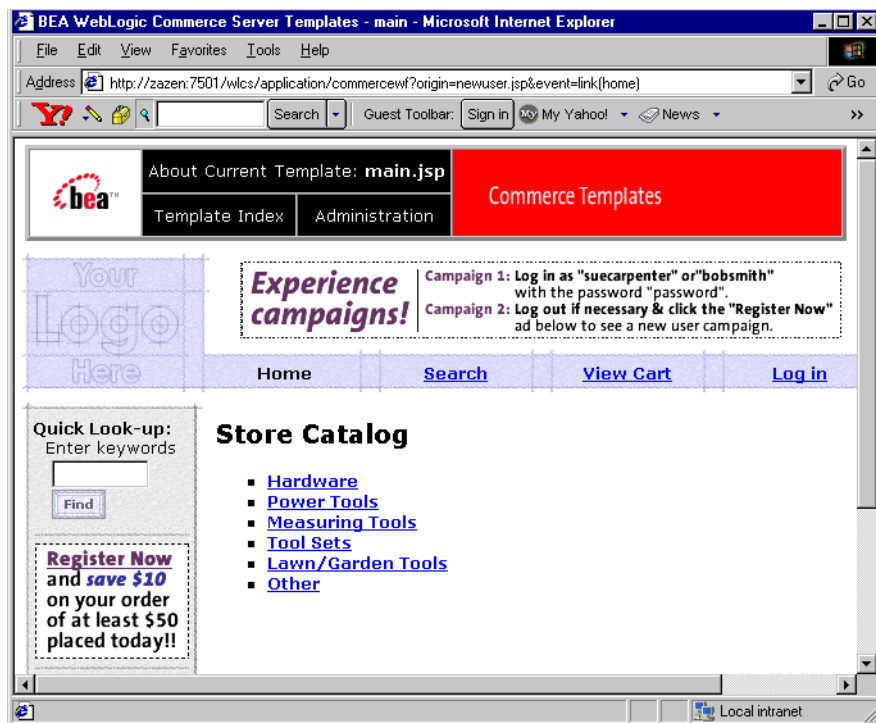
- Sue Carpenter: username is suecarpenter; password is password. When she registered with the Web site, Sue's demographic characteristics included Professional and the salary range off \$125,000 and above. This puts Sue in the "Contractor" customer segment that is defined in the E-Business Control Center.
- Bob Smith: username is bobsmith; password is password. When he registered with the Web site, Bob's demographic characteristics included "Do It Yourselfer" and the salary range of \$100,000 to \$149,999. This puts Bob in the "Affluent Mr/Ms Fixup" customer segment that is defined in the E-Business Control Center.

Notice how on the initial `main.jsp` template for the running Web application, before logging in as any user, the default banner graphics are:

- `tourBanner1.gif` (top banner that includes the text "Experience Campaigns!")
- `tourSide01.gif` (side banner that includes the text "Register Now...")

Figure 7-6 shows the initial display for the sample Web application:

Figure 7-6 Initial Display for the Web Application



The `tourBanner1.gif` and `tourSide01.gif` images are default images. The placeholder JSP tag is simply:

```
<ph:placeholder name="<%= banner %>" />
```

To experience the first sample campaign and see what the Web visitor will see, let's access the Web application in a browser:

Note: If the server is not already running, start it as described in “Starting the Commerce Server and Templates” on page 2-3

- Start → Programs → BEA WebLogic Platform 7.0 → WebLogic Portal 7.0 → Portal Examples → Commerce Templates → Start Commerce Templates

Or:

- `http://<wl-host>:7501/wlcs` (substitute the machine name on which the server is running in the `<wl-host>` parameter). For example:

`http://yourserver:7501/wlcs`

Click the **Log In** link in the horizontal navigation section of `main.jsp`.

When you arrive on the `login.jsp` template, log in as:

Username: `suecarpenter`

Password: `password`

If you entered Sue's username and password correctly, notice the changes when you arrive back on the `main.jsp` template. Figure 7-7 shows the sample display.

Figure 7-7 Main.jsp Display for Logged-in User Sue Carpenter

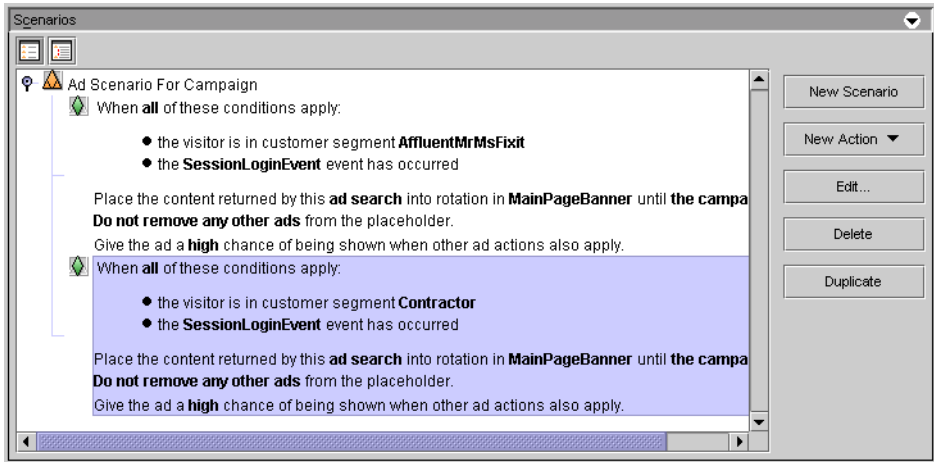


Sue is in the Contractor customer segment. Notice the targeted “Contractor’s Journal” ad is now used in the banner. The image file name is `tourBannerJournal.gif`.

Let’s go back to the E-Business Control Center.

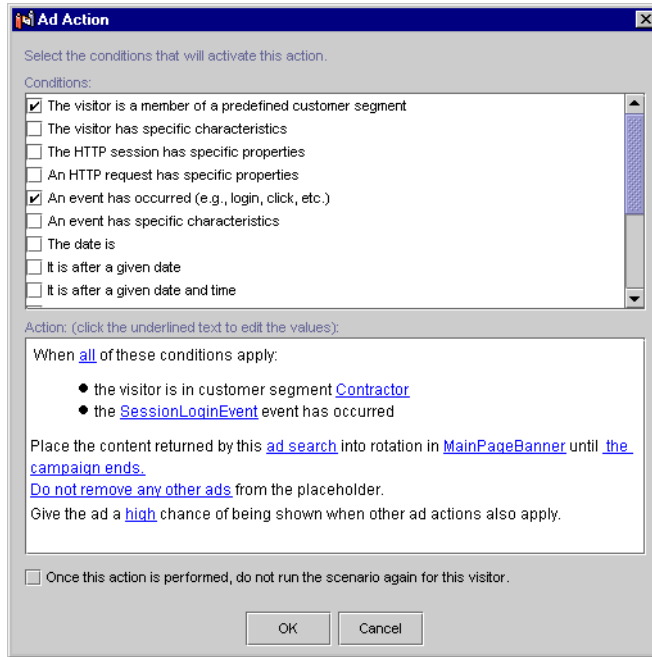
1. On the Editor window for TourCampaign1, in the Scenarios section, click in the second action that references the Contractor customer segment, as shown in [Figure 7-8](#).

Figure 7-8 Action that References Contractor Customer Segment



2. With that action highlighted, click the **Edit** button. On the Ad Action window, click the “ad search” link shown in [Figure 7-9](#).

Figure 7-9 Notice the “ad search” link on the Ad Action Window

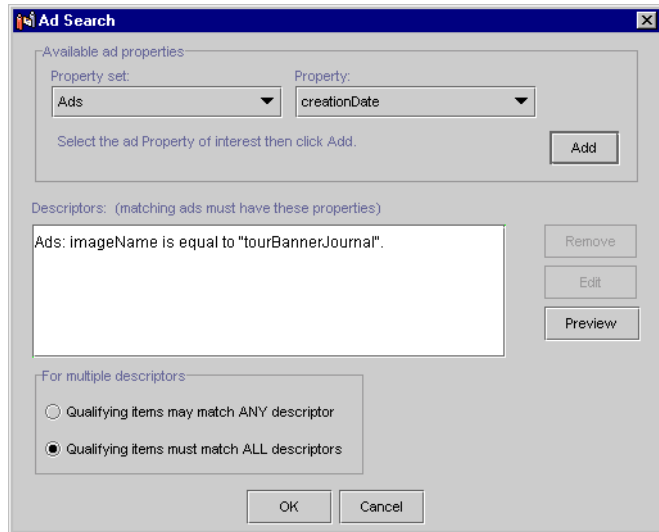


3. Click the “**ad search**” link. Figure 7-10 shows the Ad Search window.
4. If you are prompted to log on, enter the following:

Username: system
Password: weblogic

The password is set when WebLogic Server is installed. Ask your system administrator for the proper password. You can also have your system administrator add you to the Administrators group in the WebLogic Portal Administration Tools. Then you can use your normal username and password.

Figure 7-10 Ad Search Window



Of course, you can now see the `tourBannerJournal.gif` image that is associated with customers in the Contractor customer segment, as defined in the `TourCampaign1`. If you want, you can highlight the item in the Descriptors row to see how the Preview function works.

On the Web application, you can log out as Sue Carpenter and then log in as Bob Smith. (The username is `bobsmith` and the password is `password`.) Bob is in the “Affluent Mr/Ms Fixup” customer segment. [Figure 7-11](#) shows the `main.jsp` display for the logged in Bob Smith.

Figure 7-11 Main.jsp Display for Logged In User Bob Smith



As you can see, the targeted banner ad for Bob Smith is using the `tourBannerFixup.gif` image. You can confirm this in the E-Business Control Center by following the scenario action that references the Affluent Mr/Ms Fixup customer segment.

After you create a new placeholder in the E-Business Control Center, a corporate enterprise developer must use a special tag (the `<ph:placeholder>` JSP tag) in the location that you want to display ads. For example, if you create a placeholder to display ads in the top banner, a corporate enterprise developer must add a `<ph:placeholder>` tag to the top banner.

When a customer accesses a page that contains a placeholder JSP tag, the tag starts the process of choosing a query, running the query, and then generating the HTML tags that the browser needs to display the ad.

Once a corporate enterprise developer uses this tag on a page on your Web site, you use the E-Business Control Center to change the queries that the tag can run; you do not need a corporate enterprise developer to modify the tag.

Index

A

- Add to Cart link
 - anchor 6-2
 - changing 6-1
 - verifying the modification 6-6

B

- beginning the tour 2-3
- browse.jsp
 - features 4-5, 4-7
 - UNIX syntax 4-6
- buttons
 - Add Credit Card 3-21
 - Add New Address 4-16
 - Add to Cart 3-7, 4-9, 6-1
 - Check Out 3-9, 4-10
 - Checkout 3-17
 - Create 3-11
 - Find 3-5
 - Purchase 3-28
 - Use 3-20, 4-18

C

- campaigns
 - sample tour 7-1
- checkout link
 - displaying 3-16
- checkout.jsp
 - features 3-27

- UNIX syntax 3-27

- Cloudscape database
 - installing WLS 1-2
- confirmorder.jsp
 - features 3-29
 - UNIX syntax 3-29

D

- details.jsp
 - Add to Cart link 6-6
 - features 4-8
 - UNIX syntax 4-9

H

- home page
 - for authenticated users 4-13
 - main.jsp features 2-5
- how to
 - access your user profile 5-3
 - add a credit card 5-5
 - add a new user 3-11
 - browse the catalog 4-5
 - check out 4-9
 - choose shipping options 3-17, 4-15
 - create a new user profile 3-9
 - customize JSP templates 6-1
 - enter a new shipping address 4-16
 - enter payment information 3-20
 - log out 5-11
 - purchase an item 3-26, 4-21

- review the order confirmation 3-28, 4-22
- review the shopping cart 4-9
- review your user profile 5-3
- search for products 3-5
- select a credit card 3-25, 4-20
- select an item 3-5, 4-5
- view the shopping cart 3-7
- view your order history 5-7
- view your order status 5-7

I

- input processor
 - defined 3-12

J

- JavaServer Pages (JSP)
 - about 1-2
 - finding and selecting an item map 3-2
 - new user roadmap 3-2
 - registered user roadmap 4-2
- JSP templates
 - addaddress.jsp 4-16
 - browse.jsp 4-5
 - checkout.jsp 3-26, 4-21
 - confirmorder.jsp 3-28, 4-22
 - customizing 6-1
 - include files 3-23
 - login.jsp 3-9
 - main.jsp 3-5
 - new user roadmap 3-2
 - newuser.jsp 3-11
 - orderhistory.jsp 5-8
 - orderstatus.jsp 5-9
 - payment.jsp 3-20, 3-25, 4-20
 - paymentnewcc.jsp 3-21
 - profilenewcc.jsp 5-5
 - registered user roadmap 4-2
 - restrictions 3-15
 - searchresults.jsp 3-5, 3-6

- selectaddress.jsp 4-17
- selecttaxaddress.jsp 4-18
- shipping.jsp 4-15
- shoppingcart.jsp 3-7, 4-9
- templates included within 3-22, 3-23
- user profile roadmap 5-1
- viewprofile.jsp 5-3

L

- login.jsp
 - features 3-10
 - UNIX syntax 3-10

M

- main.jsp
 - features 2-5, 2-8
 - for authenticated users 4-13
 - importing Java packages 2-7
 - importing tag libraries 2-7
 - modifying Add to Cart link 6-5
 - searching for products 3-5
 - UNIX syntax 2-5

N

- newuser.jsp
 - features 3-11, 3-14
 - UNIX syntax 3-11

O

- Oracle database
 - installing WLS 1-2
- orderhistory.jsp
 - features 5-9
 - UNIX syntax 5-9
- orderstatus.jsp
 - features 5-10
 - UNIX syntax 5-10

P

- payment.jsp
 - features 3-21, 3-25
 - UNIX syntax 3-21
- paymentnewcc.jsp
 - features 3-21
 - UNIX syntax 3-22
- Pipeline
 - components 3-8
 - components defined 3-8
 - ending the session 5-11
 - input processor 3-12
 - sessions 3-8
 - sessions defined 3-8
- profilenewcc.jsp
 - features 5-6
 - UNIX syntax 5-7

R

- registered user 4-10

S

- searchresults.jsp
 - features 3-6
 - UNIX syntax 3-6
- security model 4-11
- selectaddress.jsp
 - features 3-18
 - Java scriptlets 3-19
 - UNIX syntax 3-18
- selecttaxaddress.jsp
 - features 4-19
 - UNIX syntax 4-19
- shipping.jsp
 - features 3-17
 - UNIX syntax 3-17
- shoppingcart.jsp
 - features 3-7
 - UNIX syntax 3-7

- starting the tour 2-3
- successfullogin.jsp
 - features 3-15
 - UNIX syntax 3-15
- support, technical iv-vii

T

- template include files 3-23
- tour, starting 2-3

U

- User Management 3-25

V

- viewprofile.jsp
 - features 5-4
 - UNIX syntax 5-5

W

- Webflow 3-8
 - defined 3-8
- Webflow and Pipeline Editor 6-5
- WebLogic Portal
 - product catalog 2-9
- WL_COMMERCE_HOME, description of 1-2
- WLCS_ORDER database table 5-9