



# BEA WebLogic Commerce Server

## Tour of the JSP Templates

BEA WebLogic Commerce Server 3.2  
Document Edition 3.2.1  
January 2001

# Copyright

Copyright © 2001 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

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

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

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

## Trademarks or Service Marks

BEA, WebLogic, Tuxedo, and Jolt are registered trademarks of BEA Systems, Inc. How Business Becomes E-Business, BEA WebLogic E-Business Platform, BEA Builder, BEA Manager, BEA eLink, BEA WebLogic Commerce Server, BEA WebLogic Personalization Server, BEA WebLogic Process Integrator, BEA WebLogic Collaborate, BEA WebLogic Enterprise, and BEA WebLogic Server are trademarks of BEA Systems, Inc.

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

## Tour of the WebLogic Commerce Server JSP Templates

---

<b>Document Edition</b>	<b>Date</b>	<b>Software Version</b>
3.2.1	January 2001	BEA WebLogic Commerce Server 3.2

---

---

# Contents

## About This Document

What You Need to Know .....	x
e-docs Web Site .....	x
How to Print the Document .....	x
Related Information .....	xi
Contact Us! .....	xi
Documentation Conventions .....	xii

## 1. What Is the JSP Templates Tour?

About JSP .....	1-2
About WebLogic Commerce Server Database Schemas .....	1-2
About the Sample Database .....	1-3
About TAXWARE .....	1-4
About CyberCash .....	1-4

## 2. Starting the Tour

Access the Home Page from UNIX .....	2-1
Access the Home Page from Windows .....	2-2
Overview of the Home Page .....	2-3
Features of main.jsp for Anonymous Users (Part 1 of 2) .....	2-4
Technical Note: Importing Tag Libraries and Java Packages .....	2-6
More Features of main.jsp for Anonymous Users (Part 2 of 2) .....	2-7
Technical Note: The Product Catalog .....	2-8

## 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 .....	3-8
Step 4: Create a New User Profile .....	3-9
Features of login.jsp .....	3-10
Features of newuser.jsp (Part 1 of 2).....	3-11
Technical Note: Input Processors.....	3-12
More Features of newuser.jsp (Part 2 of 2) .....	3-14
Step 5: Log In as a Registered User .....	3-15
Features of login.jsp .....	3-15
Features of successfullogin.jsp.....	3-16
Step 6: Choose Shipping Options .....	3-17
Features of shipping.jsp.....	3-18
Features of selectaddress.jsp .....	3-19
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).....	4-6
More Features of browse.jsp (Part 2 of 2).....	4-8
Features of details.jsp .....	4-10

---

Step 2: Review the Shopping Cart and Check Out.....	4-11
Technical Note: The Security Model .....	4-12
Detour: Review the Home Page for Authenticated Users.....	4-13
Features of main.jsp for Authenticated Users .....	4-14
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 .....	5-2
Step 2: Review Your User Profile.....	5-3
Features of viewprofile.jsp.....	5-4
Technical Note: The Database for WebLogic Commerce Servers .....	5-5
Step 3: Add a Credit Card .....	5-5
Features of profilenewcc.jsp .....	5-6
Step 4: View Your Order History and Status .....	5-8
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 GetTopCategories Input Processor .....	6-5
Step 3: Verify the Modification.....	6-6
What's Next?.....	6-8

## Index



---

# About This Document

WebLogic Commerce Server™ includes a set of templates that you can use to set up your e-commerce Web site quickly. This document and a sample database that is installed with WebLogic Commerce Server provide a tour through the templates to demonstrate their scope and design.

The templates consist of JavaServer Pages (JSP) and image files, and rely on WebLogic Commerce Server to render them viewable in a Web browser.

This document includes the following topics:

- Chapter 1, “What Is the JSP Templates Tour?,” which introduces the goals of the tour and describes some basic JSP and WebLogic Commerce Server concepts.
- Chapter 2, “Starting the Tour,” which describes the JSP templates Home page.
- Chapter 3, “New User Buys a Product,” which presents a scenario and illustrates how the JSP templates guide new users through the site.
- Chapter 4, “Registered User Buys a Product,” which presents a scenario and illustrates how the JSP templates change for a user who has created a user profile (registered) with the site.
- Chapter 5, “Registered User Adds a Credit Card,” which illustrates the JSPs for managing a user profile.
- Chapter 6, “Example of Customizing the Templates,” which provides an example of modifying the JSP templates.

Other WebLogic Commerce Server documents, *Product Catalog Management*, *Using the Commerce Server Order Processing Package*, *Webflow and Pipeline Management*, and *Using the Commerce Server Registration and User Processing Package* describe how to modify and extend these templates to meet your specific business needs.

---

# What You Need to Know

This document is intended for project managers, application developers, and Web site developers who are setting up an e-commerce Web site with WebLogic Commerce Server. It assumes a basic familiarity with BEA WebLogic Server, which serves as the platform for WebLogic Commerce Server and BEA WebLogic Personalization Server.

## e-docs Web Site

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

## How to Print the Document

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

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

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at <http://www.adobe.com/>.



---

# Related Information

The following WebLogic Commerce Server documents contain information that is relevant to JavaServer Pages (JSP).

- *Order Processing Package*
- *Registration and User Processing Package*
- *Product Catalog Management*

For more information in general about JSPs, refer to the following source:

- The Sun Microsystems, Inc. JSP site at <http://java.sun.com/products/jsp/index.html>.

# Contact Us!

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

In your e-mail message, please indicate that you are using the documentation for the WebLogic Commerce Server 3.2 release.

If you have any questions about this version of WebLogic Commerce Server, or if you have problems installing and running WebLogic Commerce Server, contact BEA Customer Support through BEA WebSUPPORT at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

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

- Your name, e-mail address, phone number, and fax number
- Your company name and company address

- 
- Your machine type and authorization codes
  - The name and version of the product you are using
  - A description of the problem and the content of pertinent error messages

## Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
<b>boldface text</b>	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
<i>italics</i>	Indicates emphasis or book titles.
monospace text	Indicates code samples, commands and their options, data structures and their members, data types, directories, and filenames and their extensions. Monospace text also indicates text that you must enter from the keyboard. <i>Examples:</i> <pre>#include &lt;iostream.h&gt; void main ( ) the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
<b>monospace boldface text</b>	Identifies significant words in code. <i>Example:</i> <pre>void <b>commit</b> ( )</pre>
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> <pre>String <i>expr</i></pre>

---

<b>Convention</b>	<b>Item</b>
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> LPT1 SIGNON OR
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
[ ]	Indicates optional items in a syntax line. The brackets themselves should never be typed. <i>Example:</i> buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...
	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
...	Indicates one of the following in a command line: <ul style="list-style-type: none"> <li>■ That an argument can be repeated several times in a command line</li> <li>■ That the statement omits additional optional arguments</li> <li>■ That you can enter additional parameters, values, or other information</li> </ul> The ellipsis itself should never be typed. <i>Example:</i> buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...
.	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

---



# 1 What Is the JSP Templates Tour?

WebLogic Commerce Server includes a set of templates 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. The tour makes use of 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 WebLogic Commerce Server 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 basic WebLogic Commerce Server concepts, we recommend that you read the remaining topics in this section:

- About JSP
- About WebLogic Commerce Server Database Schemas
- About TAXWARE
- About CyberCash

## 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 WebLogic Commerce Server, 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 a description of JSPs and the JSP tags that WebLogic Commerce Server provides, refer to the following documents:

- *Product Catalog Management*
- *Registration and User Processing Package*
- *Order Processing Package*

For more information about JSP, refer to [http://www.weblogic.com/docs51/classdocs/APL\\_jsp.html](http://www.weblogic.com/docs51/classdocs/APL_jsp.html) and <http://java.sun.com/products/jsp/index.html>.

## About WebLogic Commerce Server Database Schemas

WebLogic Commerce Server (WLCS) includes schemas for both Cloudscape and Oracle databases. The schemas support all transactions that are available from the WebLogic Commerce Server JSP templates. For more information, refer to the following topics:

- “[Product Catalog Database Schema](#)” in *Product Catalog Management*
- “[Order Processing Database Schema](#)” in *Order Processing Package*
- “[Overview of the Registration and User Processing Package](#)” in *Registration and User Processing Package*

**Note:** When you install WebLogic Server, it sets up an evaluation copy of the Cloudscape DBMS. If you want to use a Cloudscape database for your production Web site, you must purchase your own copy of the Cloudscape DBMS. If you want to use an Oracle database, you must purchase and install an Oracle platform that WebLogic Commerce Server supports. For more information, see [Installing Commerce Servers](#).

## About the Sample Database

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

If you want to configure WebLogic Commerce Server to use an Oracle platform and create and populate a sample Oracle database, do the following:

1. Stop WebLogic Server if it is running.
2. Change the `weblogic.properties`, `weblogiccommerce.properties`, and `set-environment.bat` files to use Oracle databases. For more information, see [Installing Commerce Servers](#).
3. Log in to SQL Plus, and create the WebLogic Commerce Server schema by entering the following command:

```
%WL_COMMERCE_HOME%\db\oracle\create_all_oracle.sql (Windows)
$WL_COMMERCE_HOME/db/oracle/create_all_oracle.sql (UNIX)
```

where `WL_COMMERCE_HOME` is the directory into which you installed WebLogic Commerce Server.

4. Start WebLogic Commerce Server (which implicitly starts WebLogic Server).

## About TAXWARE

To calculate the amount of sales tax due, WebLogic Commerce Server integrates with the TAXWARE International, Inc. tax calculation service. The JSP templates tour includes a set of TAXWARE data files for demonstration purposes only.

If you want to use this service for your production Web site, you must set up your own account with TAXWARE. With the account you can download updated data files, which contain information about tax requirements for all tax jurisdictions in the United States. For more information, see "[Taxation Services](#)" in *Order Processing Package*.

## About CyberCash

To verify and post credit card transactions, WebLogic Commerce Server integrates with the CyberCash, Inc. payment service. If you want to use this service, you must set up an account with CyberCash.

By default, the integration with CyberCash is disabled. You can complete purchases with the integration disabled, but WebLogic Commerce Server does not send data to CyberCash.

For more information about the CyberCash integration, see "[Payment Services](#)" in *Order Processing Package*.



# 2 Starting the Tour

**Note:** Before you can start the tour, you or your system administrator must install WebLogic Commerce Server. For installation information, see [Installing Commerce Servers](#).

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:

- Access the Home Page from UNIX
- Access the Home Page from Windows

Then go to the last topic in this section:

- Overview of the Home Page

## Access the Home Page from UNIX

To access the Home page from UNIX, do the following:

1. If WebLogic Commerce Server is not already running, enter the following command from a WebLogic Commerce Server host:

```
$WL_COMMERCE_HOME/StartCommerce.sh
```

2. From your workstation, open a Web browser and enter the following URL:

```
http://WLCS-host:7501/wlcs
```

For example, if the WebLogic Commerce Server host is a workstation named *spectacle*, enter the following URL in your Web browser:

```
http://spectacle:7501/wlcs
```

**Note:** By default, the WebLogic Commerce Server host name is sufficient for the `WLCS-host:7501` parameter of the URL. However, if your system administrator changes the `weblogic.properties` file to require fully qualified domain names, you must supply one. In this case, the URL in step 2 would change to `http://WLCS-host.domain-name:7501/wlcs`. For example, if the WebLogic Commerce Server host is a workstation named `spectacle` and the domain name is `sprockets.com`, enter the following URL in your Web browser:

```
http://spectacle.sprockets.com:7501/wlcs
```

# Access the Home Page from Windows

To access the Home page from Windows, do the following:

1. If WebLogic Commerce Server is not already running, on a WebLogic Commerce Server host do one of the following:
  - Click Start → Programs → WebLogic Commerce Server 3.2 → Start Commerce Server.
  - From a command prompt, enter the following command:  
`%WL_COMMERCE_HOME%\StartCommerce.bat`
2. From your computer, open a Web browser and enter the following URL:  
`http://WLCS-host:7501/wlcs`

For example, if the WebLogic Commerce Server host is a workstation named `spectacle`, enter the following URL in your Web browser:

```
http://spectacle:7501/wlcs
```

**Note:** By default, the WebLogic Commerce Server host name is sufficient for the `WLCS-host:7501` parameter of the URL. However, if your system administrator changes the `weblogic.properties` file to require fully qualified domain names, you must supply one. In this case, the URL in step 2 would change to `http://WLCS-host.domain-name:7501/wlcs`. For example, if the WebLogic Commerce Server host is a workstation named `spectacle` and the domain name is `sprockets.com`, enter the following URL in your Web browser:

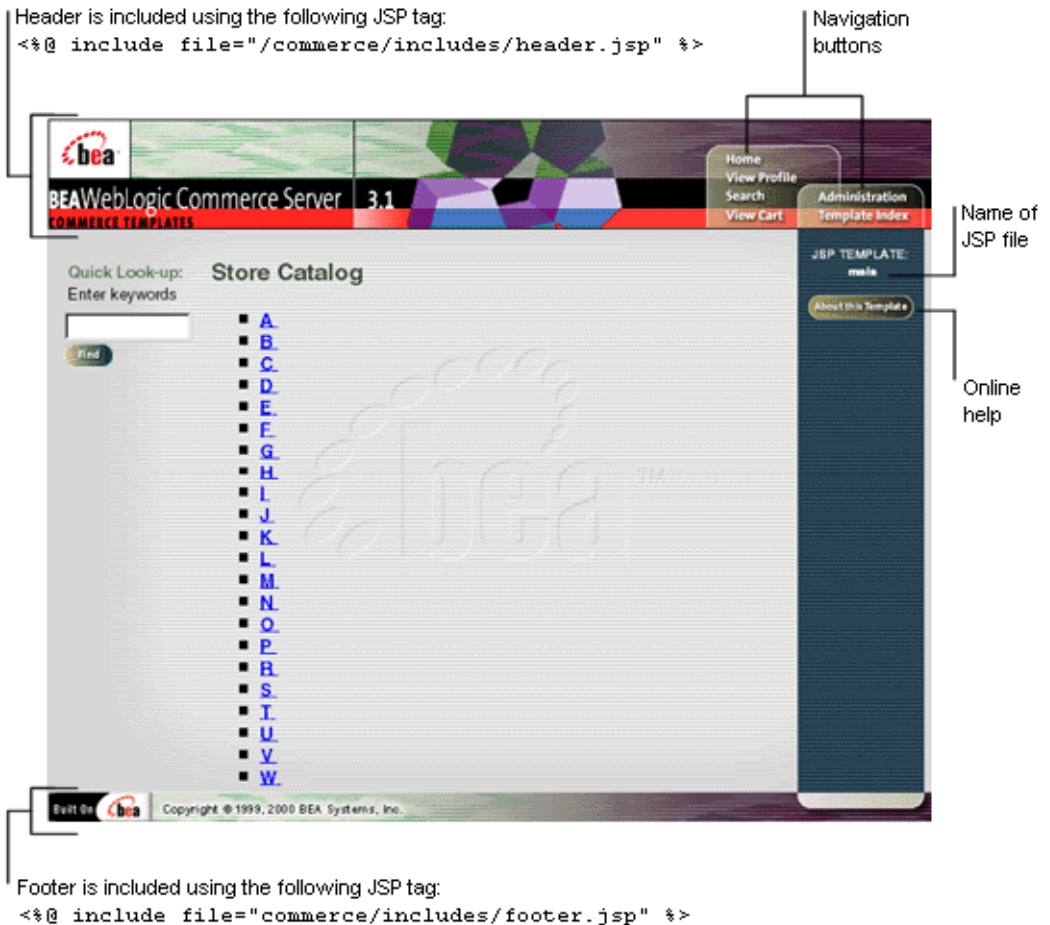
```
http://spectacle.sprockets.com:7501/wlcs
```

# 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:

- On Windows, the file's full pathname is  
`%WL_COMMERCE_HOME%\server\webapps\wlcs\commerce\main.jsp`

- On UNIX, the file's full pathname is

`$WL_COMMERCE_HOME/server/webapps/wlcs/commerce/main.jsp`

**Note:** Hereafter, the tour uses UNIX syntax for pathnames. If you use WebLogic Commerce Server 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).

- The header and footer are defined in separate JSP template files. JSP tags specify that WebLogic Commerce Server includes these files when your browser requests `main.jsp`.
- It displays the JSP filename in the upper right corner. 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.
- It includes buttons for navigating to different areas of the site:
  - Home links to `main.jsp`. Clicking this link from `main.jsp` reloads the page.
  - 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.
  - View Profile links to `userprofile.jsp`, which displays the address and credit card information that the user has provided. WebLogic Commerce Server prompts anonymous users to log in before it displays `userprofile.jsp`.
  - Administration links to tools for setting up and maintaining WebLogic Commerce Server and WebLogic Personalization Server features. This tour does not describe the administration tools; to access them, you need the WebLogic Commerce Server administrator username and password. For more information, see “[Administration Tasks](#)” in *Product Catalog Management*, *Commerce Server Order Processing Package*, and *Using the BEA WebLogic Personalization Server*.
  - Template Index. Links to `templateindex.html`, which briefly describes all JSP templates included in WebLogic Commerce Server.

### 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="pipeline.tld" prefix="pipeline" %>
<%@ taglib uri="cat.tld" prefix="catalog" %>
<%@ taglib uri="um.tld" prefix="um" %>
```

---

For a description of JSP tags and tag libraries that the JSP templates use, refer to the following sections:

- [“Webflow and Pipeline JSP Tags”](#) in *Webflow and Pipeline Management*
- [“The Product Catalog JSP Templates and Tag Library”](#) in *Product Catalog Management*
- [“JSP Tag Reference”](#) in *WebLogic Personalization Server Developer’s Guide* (for the User Management tag library)

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.*" %>
<%@ page import="com.beasys.commerce.foundation.pipeline.*" %>
<%@ page import="com.beasys.commerce.ebusiness.catalog.*" %>
<%@ page import="com.beasys.commerce.webflow.tags.WebFlowTagConstants" %>
<%@ page import="com.beasys.commerce.axiom.contact.*" %>
<%@ page import="com.beasys.commerce.ebusiness.customer.*" %>
```

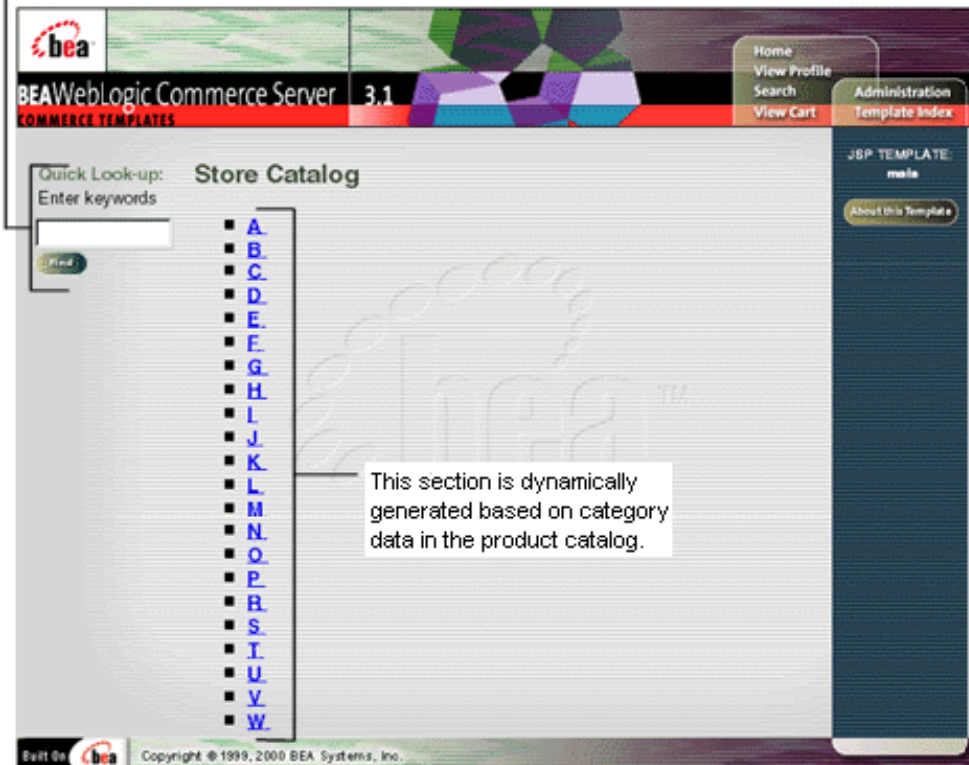
---

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

Figure 2-2 Top Level of the Product Catalog

Quick Look-up is included using the following JSP tag:

```
<%@ include file="/commerce/catalog/includes/quicksearch.jsp" %>
```



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 Commerce Server 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 Commerce Server 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

---

```
<%-- Get the top category from the PipelineSession. --%>

<pipeline:getPipelineProperty propertyName="<%=
PipelineSessionConstants.CATALOG_CATEGORY %>"
returnName="topCategory"
returnType="com.beasys.commerce.ebusiness.catalog.Category"
attributeScope="<%= PipelineConstants.REQUEST_SCOPE %>" />

<%-- Get the subcategories of the top category from the PipelineSession. --%>

<pipeline:getPipelineProperty propertyName="<%=
PipelineSessionConstants.CATALOG_CATEGORIES %>"
returnName="subcategories"
returnType="com.beasys.commerce.ebusiness.catalog.ViewIterator"
attributeScope="<%= PipelineConstants.REQUEST_SCOPE %>" />

<%-- Iterate through all the subcategories, one category at a time, and list the
name of each category --%>
<p class="headl">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 + "=" +
currentCategory.getKey().getIdentifier(); %>
<div class="head2">
<li>
<a href="<%= WebflowJSPHelper.createWebflowURL(pageContext, "main.jsp",
"link(browse)", browseParameters, true) %>">
<catalog:getProperty object="<%= currentCategory %>" propertyName="Name"/>
</a>
<br>
</div>

</catalog:iterateViewIterator>

</ul>
```

---

WebLogic Commerce Server 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 Commerce Server database, you can modify the JSP templates to use your existing catalog system. (However, this low-level modification requires Java programming skills.) For more information about using existing data with WebLogic Commerce Server, see [“Using the Product Catalog Database Loader”](#) in *Product Catalog Management*.

### 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 WebLogic Commerce Server 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 site takes you along the following path:

- 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: Log In as a Registered User
- 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 site; the darker arrows show the path that this section of the tour describes.

**Figure 3-1 JSPs for Finding and Selecting an Item**

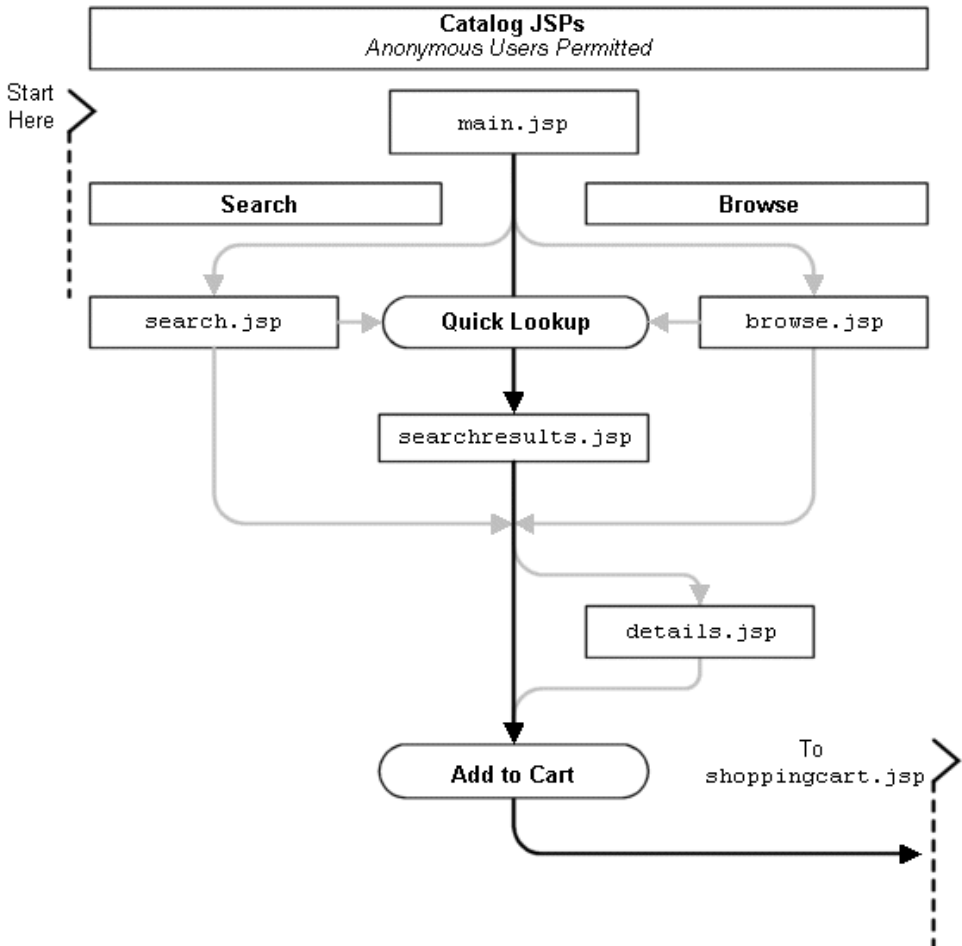


Figure 3-2 JSPs for the Shopping Cart and New User Registration

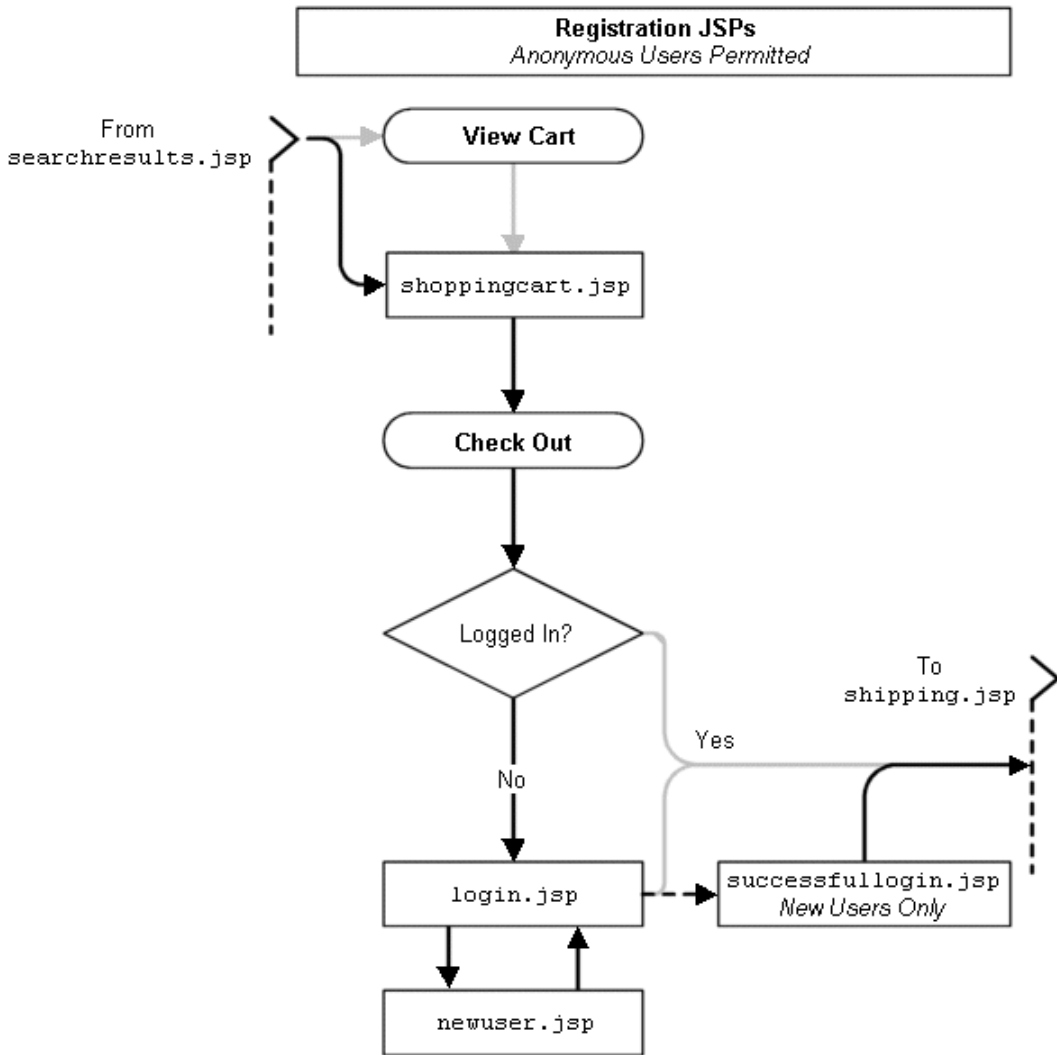
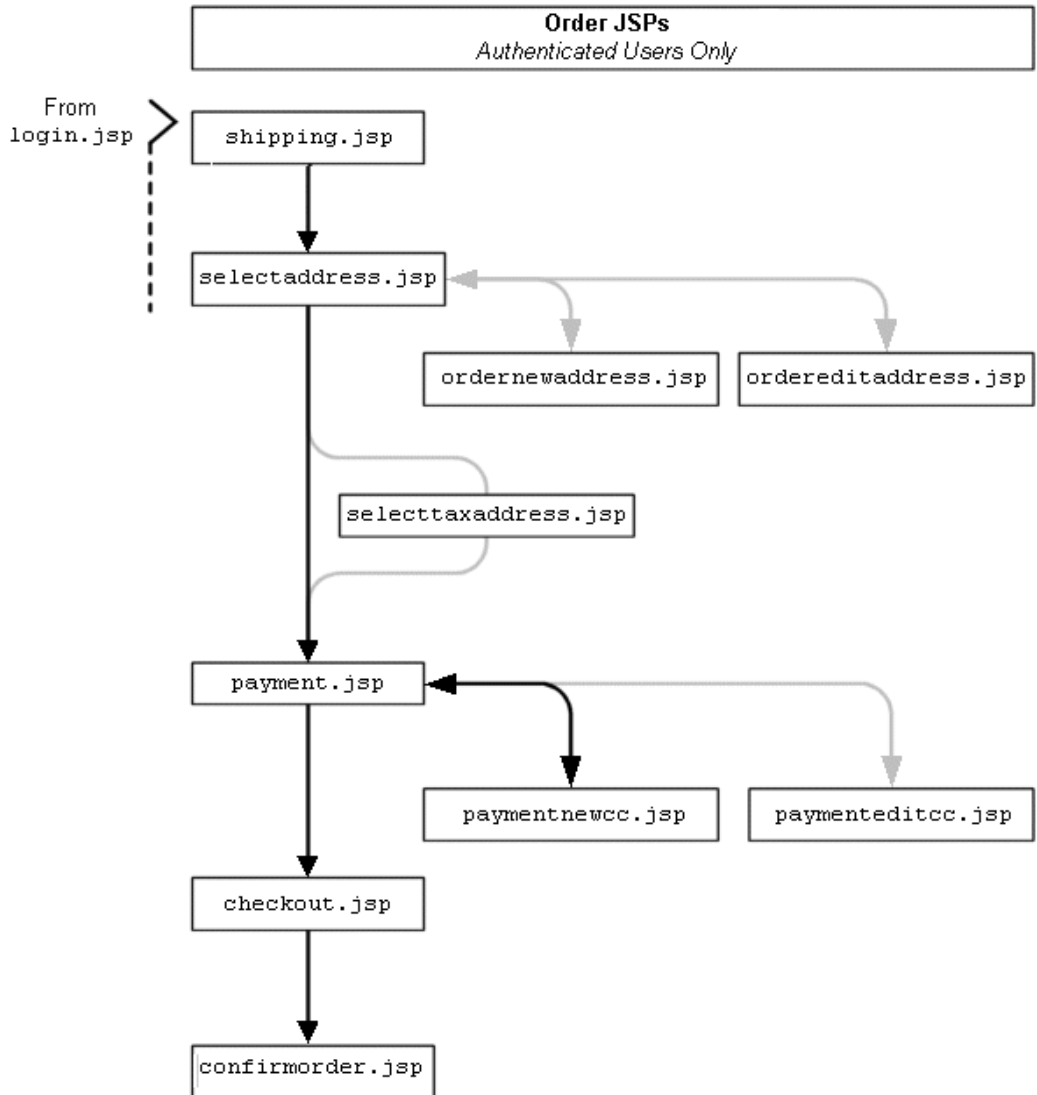


Figure 3-3 JSPs for Completing the Order



## Step 1: Search for Products

When you first access the 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 Commerce Server 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



This section shows the results of the query.

Notice the following features of `searchresults.jsp`:

- The file's full pathname is `$WL_COMMERCE_HOME/server/webapps/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 Commerce Server lists all items you have selected during the current visit to the site in `shoppingcart.jsp` (see Figure 3-6).

### Features of `shoppingcart.jsp`

Figure 3-6 Shopping Cart

Static HTML

Generated with JSP tags

The screenshot displays the BEA WebLogic Commerce Server interface. At the top, there's a navigation bar with links for Home, View Profile, Search, View Cart, Administration, and Template Index. The main content area is titled 'Shopping Cart' and includes instructions for reviewing items and using the 'Delete' button. Below the instructions, there's a table with columns for Quantity, Item, List Price, Our Price, You Save, and Subtotal. The table contains one row for 'hammer-9-MKHB241Q' with a quantity of 1, a list price of USD311.85, an our price of USD284.85, a you save amount of USD27.00, and a subtotal of USD284.85. A 'Delete' button is next to the item. At the bottom of the table, the total price is shown as USD284.85. There are also buttons for 'Empty cart', 'Check out >', 'Continue shopping', and 'Update'.

Quantity	Item	List Price	Our Price	You Save	Subtotal
1	<a href="#">hammer-9-MKHB241Q</a>	USD311.85	USD284.85	USD27.00	USD284.85
<b>Total</b>					<b>USD284.85</b>

JSP TEMPLATE: shoppingcart

Notice the following features of `shoppingcart.jsp`:

- The file's full pathname is  
`$WL_COMMERCE_HOME/server/webapps/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 (the next topic in this section, [Technical Note: Pipeline Components, Pipeline Sessions, and Webflow](#), describes Pipeline sessions).

For details on the shopping cart, see [“Shopping Cart Management Services”](#) in *Order Processing Package*.

## **Technical Note: Pipeline Components, Pipeline Sessions, and Webflow**

So far this tour has focused on HTML elements and JSP tags, which WebLogic Commerce Server 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 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 `$WL_COMMERCE_HOME/pipeline.properties` file controls the order of Pipeline components in the Pipeline.
- *Pipeline sessions*: Storage locations for information about the current session (such as the current shopping cart) or more transient data (such as error messages about a customer's most recent input).
- *Webflow*: Controls the order of JSPs that WebLogic Commerce Server presents to users based on the current state of the Pipeline session. Instead of using individual HTML hypertext links, WebLogic Commerce Server uses a single properties file, `$WL_COMMERCE_HOME/webflow.properties`, to determine the sequence of pages on your site. Webflow centralizes control of site navigation, making it easy for you to maintain and update the 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. JSP tags that invoke Pipeline components start with the following string:

```
<pipeline:
```

### Listing 3-1 JSP Tags for Retrieving Shopping Cart Content

---

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

---

For more information on Pipelines and Webflow, see [Webflow and Pipeline Management](#).

To continue the tour...

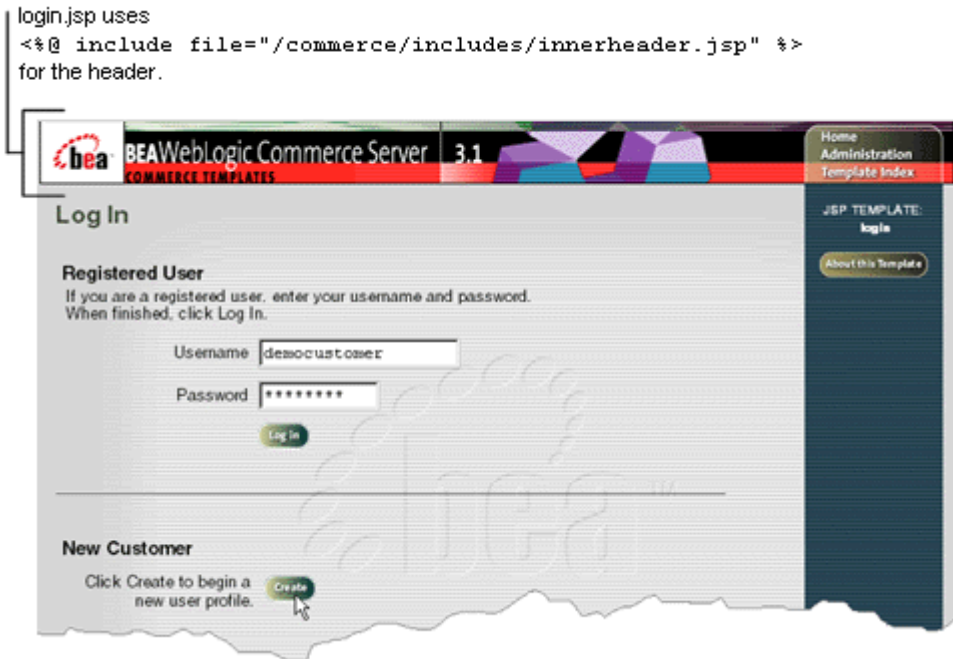
Click Check Out.

## Step 4: Create a New User Profile

When you click Check Out from the shopping cart, WebLogic Commerce Server 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  
\$WL\_COMMERCE\_HOME/server/webapps/wlcs/commerce/login.jsp (UNIX)
- It uses innerheader.jsp for the header and innerfooter.jsp for the footer.  
After a user logs in, most JSPs use innerheader.jsp and innerfooter.jsp.
- 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 Commerce Server prompts you to provide information for your user profile in `newuser.jsp` (see Figure 3-8 and Figure 3-9).

## Features of `newuser.jsp` (Part 1 of 2)

Figure 3-8 Entering Address Information

The screenshot shows a web form titled "Create a New Account" with the following fields and annotations:

- First name:** Input field with "Pat" and an asterisk (\*). Annotation: "\*" Indicates a required field, which is validate by an input processor.
- Middle initial:** Input field with "D".
- Last name:** Input field with "Gomes" and an asterisk (\*).
- Address:** Input field with "1 Yawkey Way" and an asterisk (\*).
- Address 2:** Empty input field.
- City:** Input field with "Boston" and an asterisk (\*).
- State:** Dropdown menu with "MA" selected and an asterisk (\*). Annotation: List of states comes from `states.jsp`.
- Zip/Postal Code:** Input field with "02110" and an asterisk (\*).
- Country:** Dropdown menu with "United States" selected and an asterisk (\*). Annotation: List of countries comes from `countries.jsp`.
- Home phone:** Input field with "867-5309".
- Business phone:** Empty input field.
- Email address:** Input field with "pat@mail.org" and an asterisk (\*).

At the bottom, there is a "Shipping Address" section with a radio button labeled "Same as above" which is selected.

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

- The file's full pathname is `$WL_COMMERCE_HOME/server/webapps/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` (see Figure 4-13) or from `viewprofile.jsp` (see Figure 5-3).
- It includes the following files to provide values for the State and Country lists:  
\$WL\_COMMERCE\_HOME/server/webapps/wlcs/commerce/includes/states.jsp and  
\$WL\_COMMERCE\_HOME/server/webapps/wlcs/commerce/includes/countries.jsp

The State and Country lists are defined in separate JSP files because they are used by several JSP templates.

## 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="<%=WebflowJSPHelper.createWebflowURL(pageContext,
"newuser.jsp", "button(save)", false)%>">

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

<input type=hidden name=event value="button(save)">
```

---

Listing 3-3 shows the excerpt from `webflow.properties` that directs information posted from `newuser.jsp`. In the excerpt, Webflow sends information to the `newuser.CustomerProfile.inputprocessor`; then it directs the results from the `CustomerProfile` input processor to the `CustomerProfile` Pipeline. If the input processor detects an error, it redisplay `newuser.jsp` and populates it with data from the current Pipeline session.

### Listing 3-3 Webflow Sends Data to an Input Processor

---

```
# Process the data from the form
newuser.jsp.button(save)=newuser.CustomerProfile.inputprocessor
# And send it off to create the user
newuser.CustomerProfile.inputprocessor.success=CustomerProfile.pipeline
...
# On an error, bring them back to the newuser.jsp form.
# It will be redisplayed with their old data and error messages added in.
newuser.CustomerProfile.inputprocessor.exception(InvalidInputException)=
commerce/register/newuser.jsp
CustomerProfile.pipeline.exception(PipelineFatalException)=commerce/register/
newuser.jsp
```

---

For a description of input processors that WebLogic Commerce Server provides, refer to the following documents:

- [Product Catalog Management](#)
- [Registration and User Processing Package](#)
- [Order Processing Package](#)

## More Features of newuser.jsp (Part 2 of 2)

Figure 3-9 Entering username and Password

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

**Payment Information (Optional):**

- Credit card type:  \*
- Name on card:
- Card number:
- Expiration date (mm/yyyy):   \*
- Card billing address:
- Address 2:
- City:
- State:  \*
- Zip/Postal Code:
- Country:

**Username and Password:**

- Username:  \*
- Password:  \*
- Repeat password to confirm:  \*

Fields marked with (\*) are required.

Buttons:

Annotations:

- Pointing to the Credit card type and Expiration date fields: "List values are in newcctemplate.jsp"
- Pointing to the State and Country fields: "List files come from Include files."
- Pointing to the Username and Password fields: "The sample JSP templates do not restrict character types in user names and passwords."

Footer: Copyright © 1999, 2000 BEA Systems, Inc.

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

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

The values for the Credit Card Type and Expiration Date lists are defined in newcctemplate.jsp because newcctemplate.jsp 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...

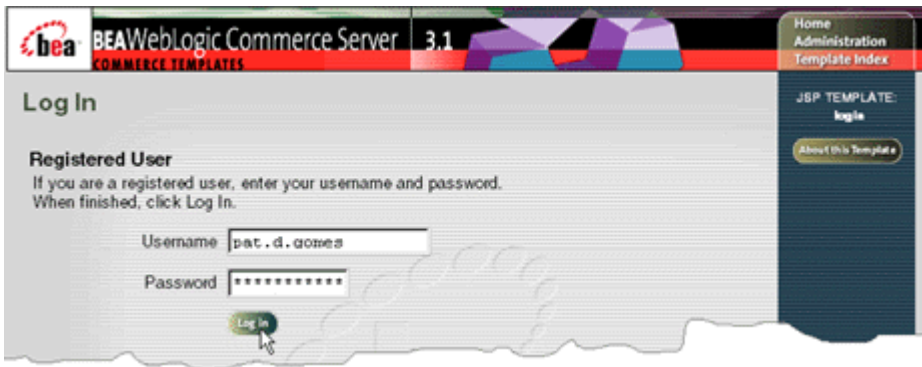
Complete the required fields and click Save.

## Step 5: Log In as a Registered User

When you click Save, WebLogic Commerce Server prompts you to log in from `login.jsp` (see Figure 3-10).

### Features of `login.jsp`

Figure 3-10 Logging In



For a description of `login.jsp`, see “Features of `login.jsp`” on page 3-10.

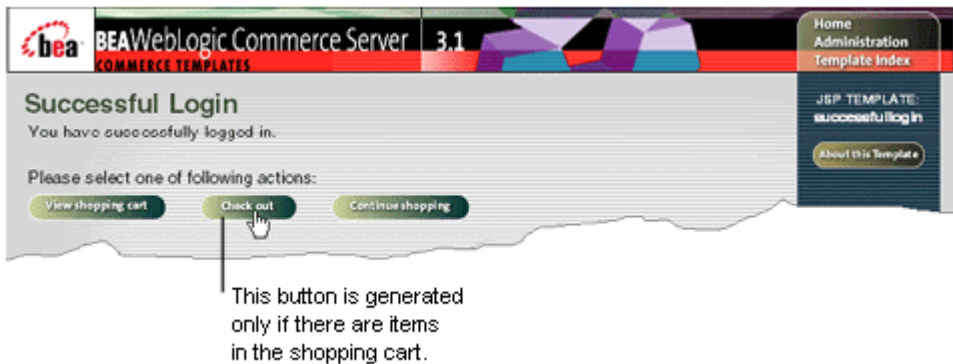
To continue the tour...

Type the username and password you entered with your new user profile. Then click Log In.

When you click Log In, WebLogic Commerce Server displays `successfullogin.jsp` (see Figure 3-11).

## Features of `successfullogin.jsp`

Figure 3-11 `successfullogin.jsp`



Notice the following features of `successfullogin.jsp`:

- The file's full pathname is `$WL_COMMERCE_HOME/server/webapps/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 3-4 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 3-4 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...

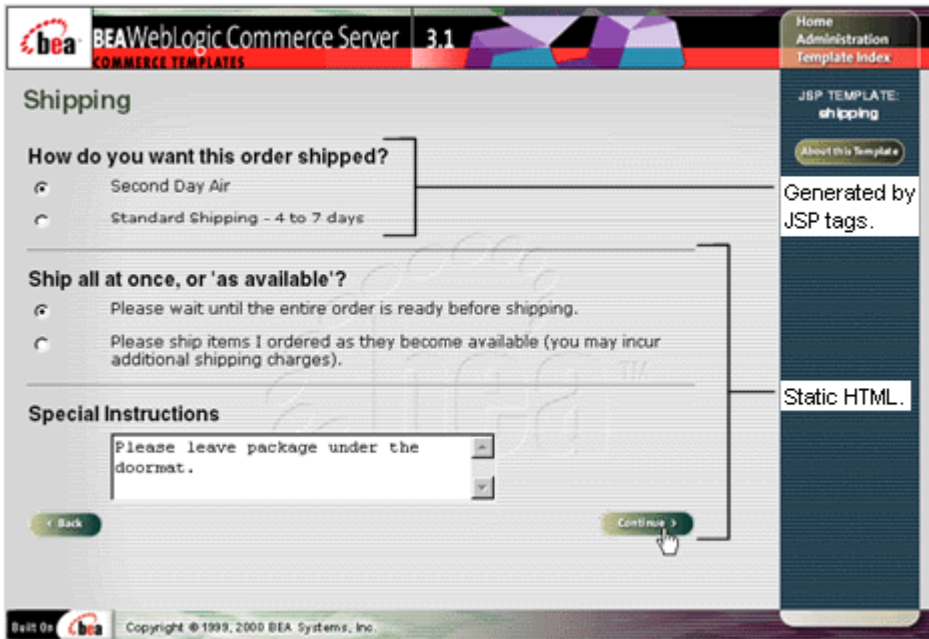
Click the Checkout button.

## Step 6: Choose Shipping Options

When you click Checkout, WebLogic Commerce Server 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



Notice the following features of `shipping.jsp`:

- The file's full pathname is `$WL_COMMERCE_HOME/server/webapps/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 Commerce Server 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 Commerce Server 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 `$WL_COMMERCE_HOME/server/webapps/wlcs/commerce/order/selectaddress.jsp` (UNIX)
- It uses Java scriptlets to iterate through the list of shipping addresses in the user profile (see Listing 3-5).
- It invokes the **Taxation Service**, a WebLogic Commerce Server implementation of business logic and presentation logic that interfaces with TAXWARE to determine the taxes that are due. For information about TAXWARE, see "About TAXWARE" on page 1-4. For more information about the Taxation Service, see "[Taxation Services](#)" in *Order Processing Package*.

#### **Listing 3-5 Java Scriptlets in selectaddress.jsp**

---

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

<table>

    <tr>
    <td><%= shippingAddress.getStreet1() %></td>
    </tr>

    <tr>
    <td><b>Country <%= shippingAddress.getCountry() %></b></td>
    </tr>

</table>

<%
}
%>
```

---

To continue the tour...

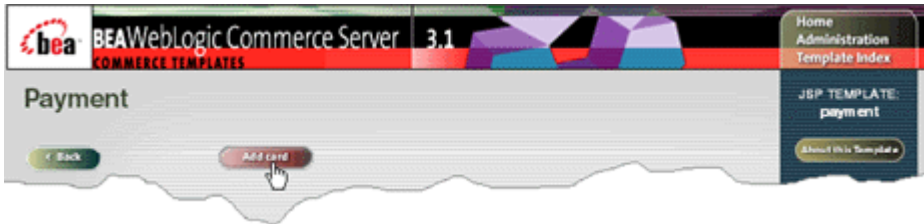
Click Use.

## **Step 7: Enter Payment Information**

When you click Use, WebLogic Commerce Server 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  
`$WL_COMMERCE_HOME/server/webapps/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 Commerce Server prompts you to enter payment information in `paymentnewcc.jsp` (see Figure 3-15).

Figure 3-15 Enter Credit Card Information

BEA WebLogic Commerce Server 3.1

Home Administration Template Index

### New Credit Card

JSP TEMPLATE: paymentnewcc  
[About this Template](#)

Credit card type	VISA *
Name on card	Pat D. Gomes *
Card number	XXXXXXXXXXXX *
Expiration date (mm/yyyy)	6/2006 *
Card billing address	1 Yawkey Way *
Address 2	
City	Boston *
State	MA *
Zip/Postal Code	02110 *
Country	United States *

Fields marked with (\*) are required.

[Save](#)

BEA Copyright © 1999, 2000 BEA Systems, Inc.

Adds to user profile.

This section comes from the following include file:

```
WL_Commerce_Home/webapps/commerce/includes/newcctemplate.jsp
```

Notice the following features of `paymentnewcc.jsp`:

- The file's full pathname is  
`WL_COMMERCE_HOME/server/webapps/wlcs/commerce/order/paymentnewcc.jsp` (UNIX)
- This page is similar to  
`WL_COMMERCE_HOME/server/webapps/wlcs/commerce/user/profilenewcc.jsp` and  
`WL_COMMERCE_HOME/server/webapps/wlcs/commerce/register/newuser.jsp`.

All three JSP files use an include file,  
`WL_COMMERCE_HOME/server/webapps/wlcs/commerce/includes/newcctemplate.jsp`, 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.”

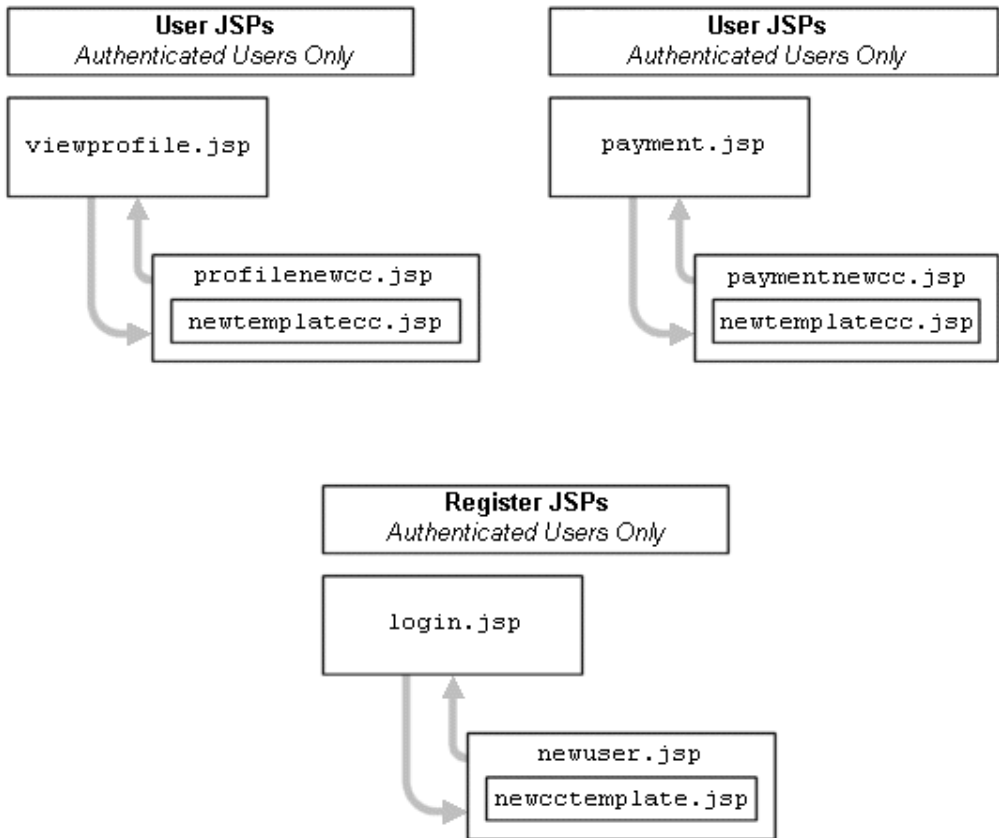
- Clicking Save updates the user profile.

### Technical Note: Templates Included Within Templates

Rather than repeating input fields and other information across several files, the WebLogic Commerce Server 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 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.

Why doesn't Webflow specify `newcctemplate.jsp` in all locations that request credit card information instead of using three files, `paymentnewcc.jsp`, `profilenewcc.jsp`, and `newuser.jsp`? 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



For a list of include files and the JSP templates that use them, see [Product Catalog Management](#) and [Order Processing Package](#).

To continue the tour...

1. Enter information for a credit card.

If you have enabled the CyberCash integration, you can use any of the following credit card types and numbers to test or demonstrate the integration:

- VISA: 4111111111111111 (one four and 15 ones)
- MasterCard: 5555555555554444 (12 fives and four fours)

- Amex: 378282246310005
- Discover: 6011111111111117

For more information about the CyberCash integration, see “About CyberCash” on page 1-4.

If you have not enabled the CyberCash integration, you can use the numbers and card types in the previous list or enter 16-18 characters for any card type. For the Card Number box, the sample JSP templates validate only the number of characters that you enter.

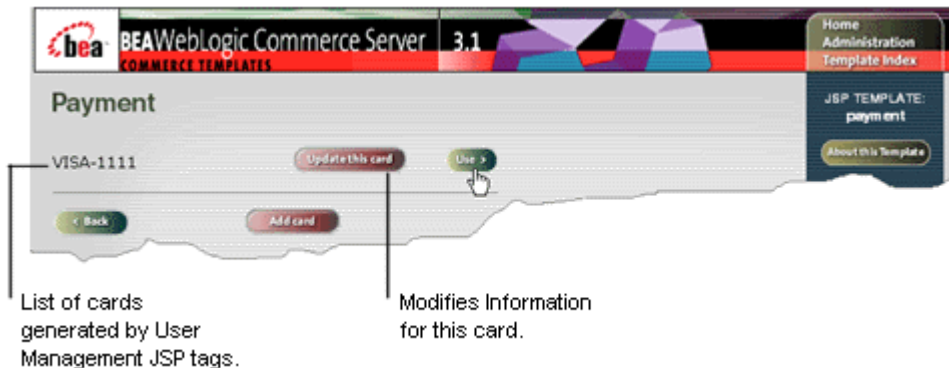
2. Complete the remaining required fields and click Save.

## Step 8: Select a Credit Card

When you click Save in `paymentnewcc.jsp`, WebLogic Commerce Server 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`:

### 3 *New User Buys a Product*

---

- 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 Commerce Server displays `checkout.jsp`, from which you review the tax information and complete the purchase.

## Features of checkout.jsp

Figure 3-18 Purchase the Item

The screenshot displays the 'Final Checkout Review' page. At the top, there is a navigation bar with links for Home, View Profile, Search, View Cart, Administration, and Template Index. The main content area is divided into several sections:

- Shipping Destination:** Pat Gomes, 1 Yawkey Way, BOSTON, MA-02110.
- Shipment Splitting Preferences:** Ship all at once.
- Special Instructions:** Please leave package under the doormat.
- Ship Via:** Second Day Air.

Below these sections is an order summary table:

ID	Description	Quantity	Our Price	SubTotal
9-MKHR2410	hammer; makita; rotary; portable electric tools, rotary hammers	1	USD 294.85	USD 294.85
			Shipping & handling	USD 4.95
			Total tax	USD 14.49
			<b>Total due</b>	<b>USD 304.29</b>

At the bottom of the page, there is a 'Complete purchase' button with a mouse cursor hovering over it. A callout box points to this button with the text: 'Invokes the Payment Service and Taxation Service.'

Invokes the Payment Service and Taxation Service.

Notice the following features of checkout . jsp:

- The file's full pathname is  
`$WL_COMMERCE_HOME/server/webapps/wlcs/commerce/order/checkout.jsp`  
 sp (UNIX)

- It uses JSP tags to gather and display information from the Pipeline, including the tax information that the **Taxation Service** calculates.
- Clicking Complete Purchase invokes the **Payment Service**, a WebLogic Commerce Server implementation of business logic that processes payments. If you have enabled the CyberCase integration, the Payment Service sends your credit card information and transaction amount to CyberCash for verification. If CyberCash verifies the card and purchase amount, WebLogic Commerce Server invokes the **Taxation Service**, which appends to your TAXWARE log file for record keeping. For more information about the Payment Service, see "[Payment Services](#)" in *Order Processing Package*.

To continue the tour...

Click Complete Purchase.

## Step 10: Review the Order Confirmation

When you click Complete Purchase, WebLogic Commerce Server 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

BEA WebLogic Commerce Server 3.1

Home  
View Profile  
Search  
View Cart

Administration  
Template Index

JSP TEMPLATE:  
confirm order  
About this Template

### Confirm Order

Thank you for shopping here with BEA WebLogic Commerce Server 3.1. Don't forget to stop back for more great deals, contests, new store openings and specials on jsp templates.

**Order Confirmation #4002**

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

**Preferences:**  
Ship all at once

Please leave package under the doormat.

The Pipeline generates the confirmation number.

ID	Description	Quantity	Unit Price	Total
9-MKHR2410	hammer-9-MKHR2410	1	USD 284.85	USD 284.85
<b>Shipping</b>				USD 4.95
<b>Total Tax</b>				USD 14.49
<b>Total Due</b>				USD 304.29

Print this page for your records.

Continue shopping

Bea © 1999, 2000 BEA Systems, Inc.

Notice the following features of `confirmorder.jsp`:

- The file's full pathname is  
`$WL_COMMERCE_HOME/server/webapps/wlcs/commerce/order/confirmorder.jsp`  
 (UNIX)
- The Pipeline mechanism generates the confirmation number.
- Pipeline JSP tags gather and display the information on this page.

### **3** *New User Buys a Product*

---

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 (not to mention the ease of use!) 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
- 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

# 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**

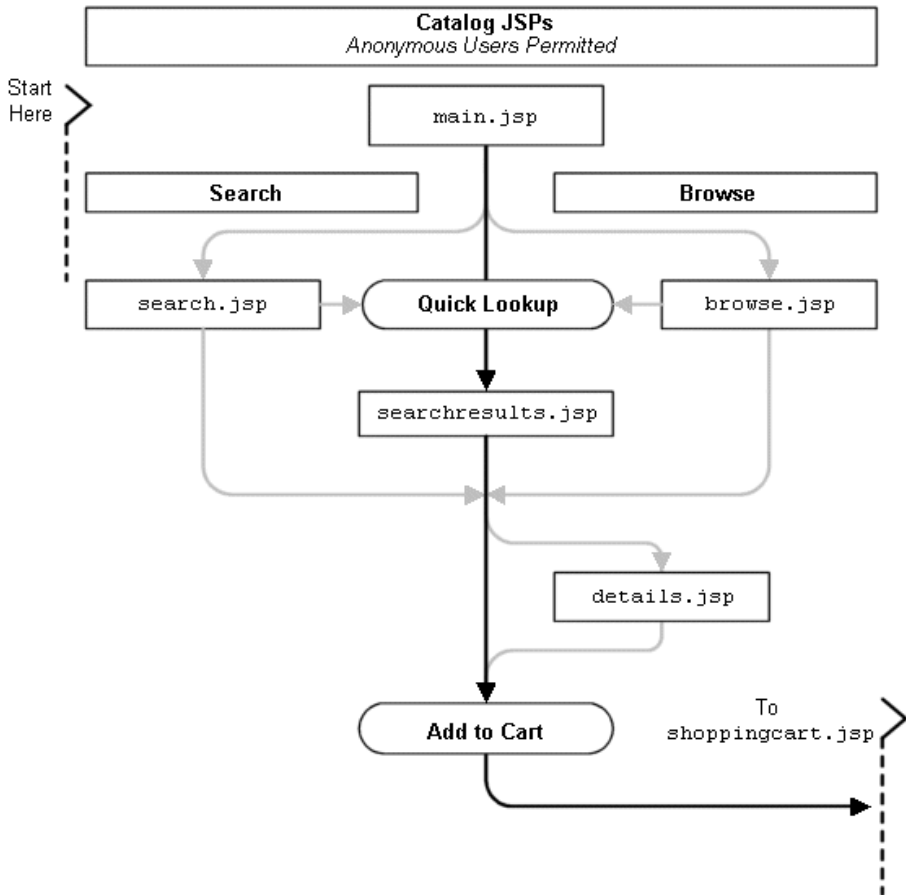


Figure 4-2 JSPs for the Shopping Cart and User Login

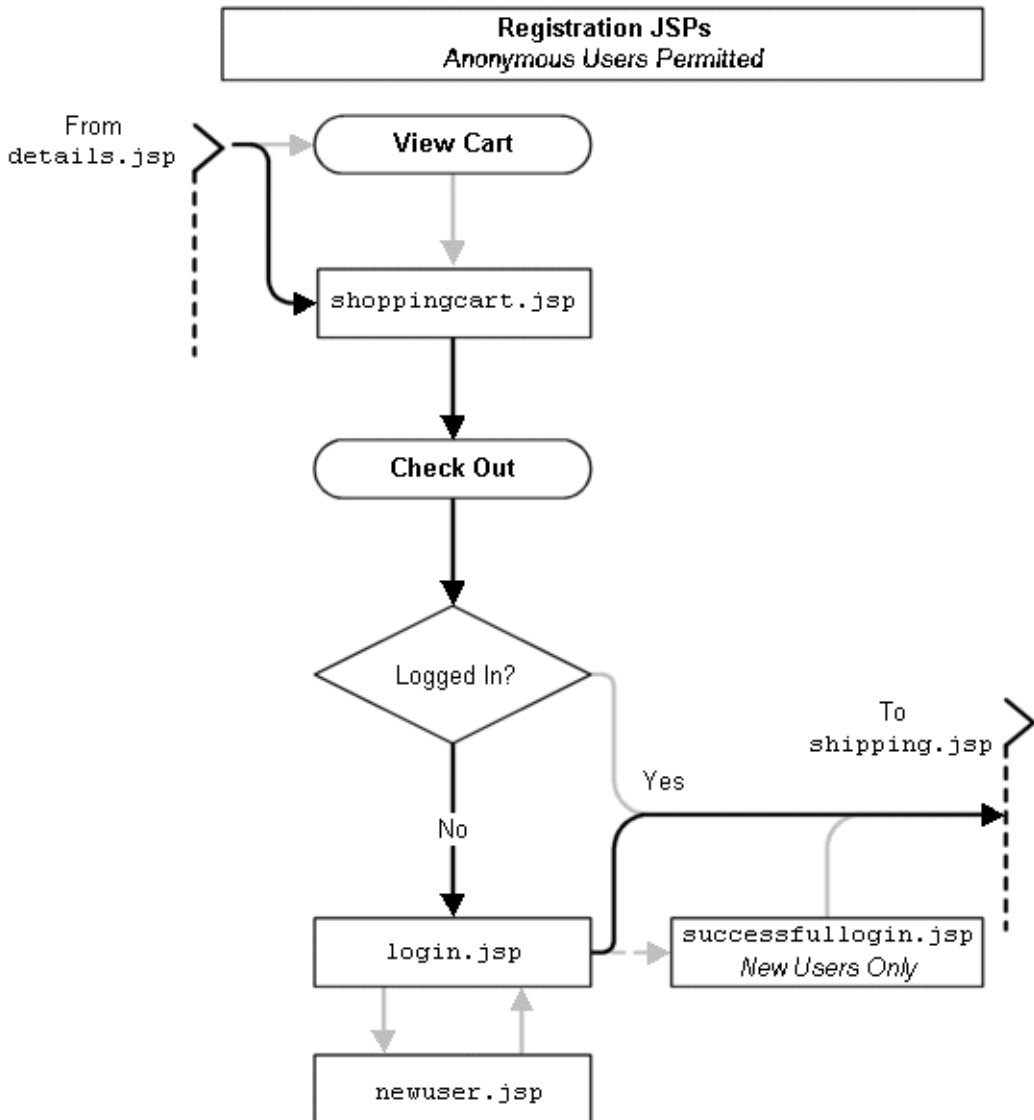
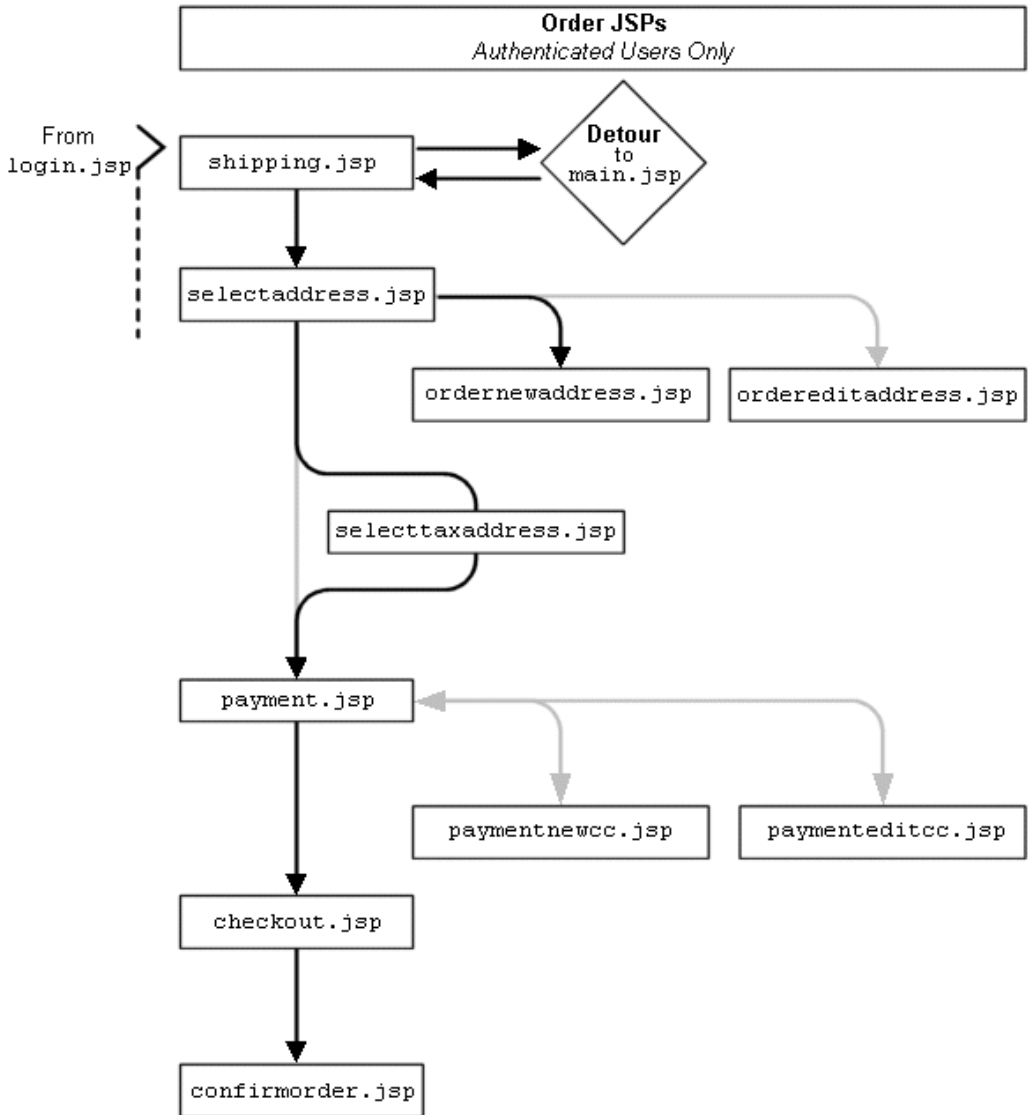


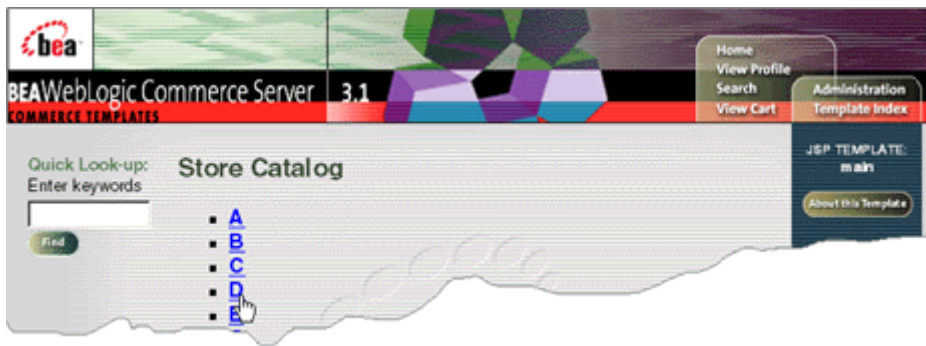
Figure 4-3 JSPs for Completing the Order



# Step 1: Browse the Catalog and Select an Item

When you arrive at the Home page for your return shopping trip, you decide to browse for drills in the catalog.

Figure 4-4 Home Page: Top Level of the Catalog



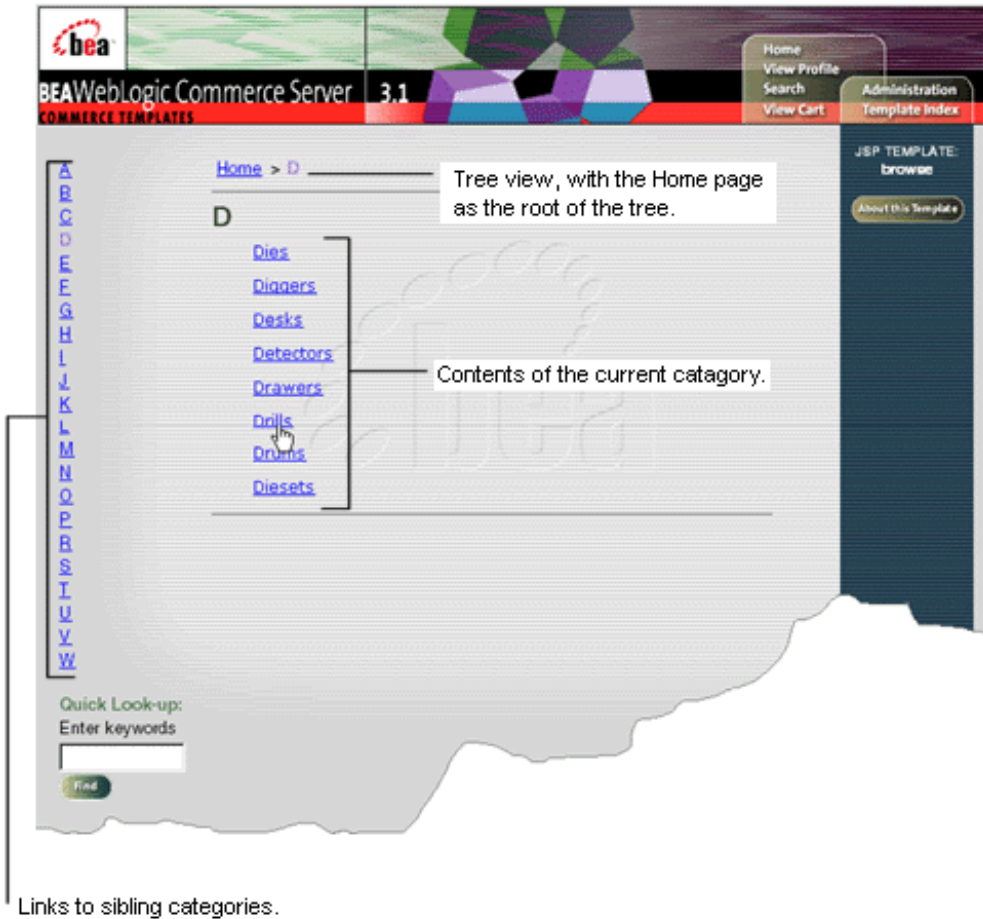
To continue the tour...

Click D.

When you click D, WebLogic Commerce Server displays the contents of the D category in `browse.jsp` (see Figure 4-5).

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

Figure 4-5 Second Level of the Catalog



Notice the following features of `browse.jsp`:

- The file's full pathname is  
`$WL_COMMERCE_HOME/server/webapps/wlcs/commerce/catalog/browse.jsp` (UNIX)

- Pipeline JSP tags in the left column query the product catalog for a list of sibling categories. The left column then displays the name of the current category and links to its siblings.
- 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

---

```
<!-- 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"/>
```

---

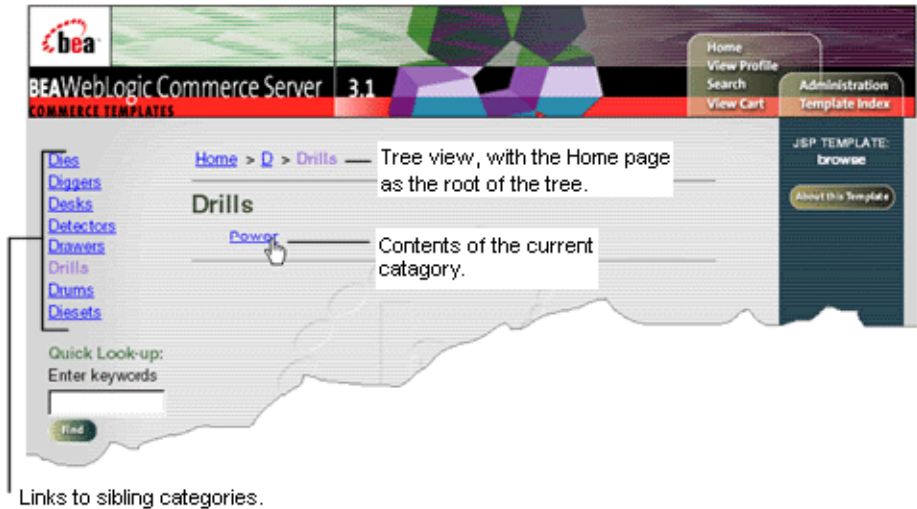
To continue the tour...

Click Drills.

When you click Drills, WebLogic Commerce Server displays the contents of the Drills category in `browse.jsp` (see Figure 4-6).

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

Figure 4-6 Third Level of the Catalog



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

- It is the same source file that Figure 4-5 illustrates, but the content is different because the user's context is different.
- In the left column, 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 4-5, the list changes.
- 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 4-5, hence the contents and layout in Figure 4-6 changes.

To continue the tour...

Click Power.



When you click Power, WebLogic Commerce Server displays a selection of power drills in `browse.jsp`. (see Figure 4-7). Each drill includes a link to detailed product information.

Figure 4-7 Contents of the Power Drills Category



To continue the tour...

Click the name of a drill.

When you click the name of a drill, WebLogic Commerce Server displays a product description in `details.jsp` (see Figure 4-8).

## Features of details.jsp

Figure 4-8 Item Details



Notice the following features of `details.jsp`:

- The file's full pathname is `$WL_COMMERCE_HOME/server/webapps/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.

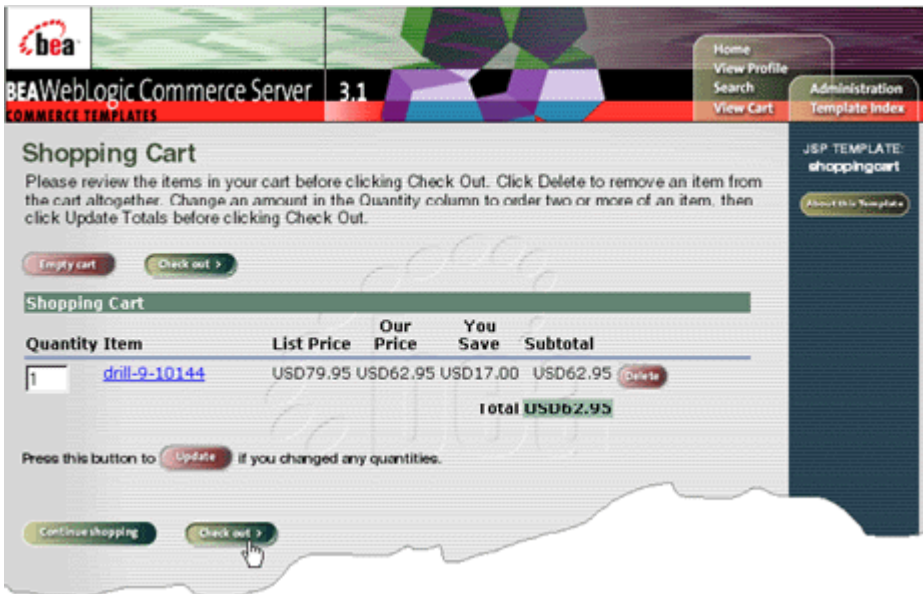
To continue the tour...

Click Add to Cart.

## Step 2: Review the Shopping Cart and Check Out

When you click Add to Cart, WebLogic Commerce Server displays your selections in `shoppingcart.jsp` (see Figure 4-9).

Figure 4-9 Shopping Cart



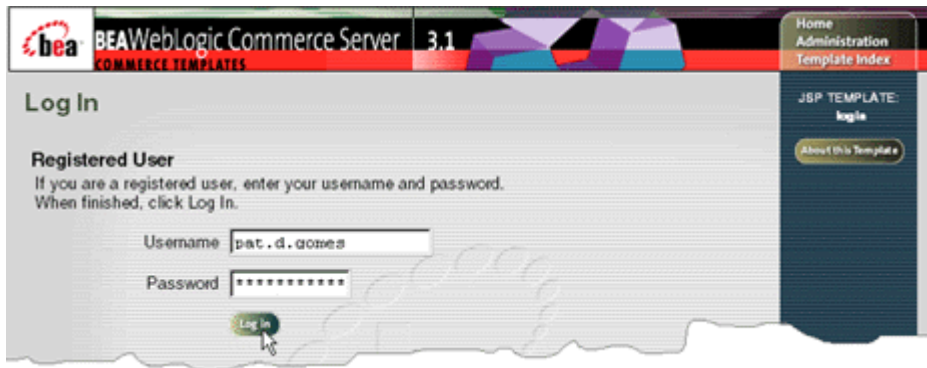
For a description of `shoppingcart.jsp`, see “Features of `shoppingcart.jsp`” on page 3-7.

To continue the tour...

Click Check Out.

When you click Check Out, WebLogic Commerce Server prompts you to log in (see Figure 4-10). Because you created a user profile in Chapter 3, “New User Buys a Product,” you are now a registered user.

**Figure 4-10 Log In**



For a description of `login.jsp`, see “Features of `login.jsp`” on page 3-10.

### Technical Note: The Security Model

How does WebLogic Commerce Server determine if users must log in when they click a link? Instead of using Webflow to implement security, WebLogic Commerce Server follows the J2EE specification for security. With this specification, the 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 Commerce Server 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 Commerce Server displays shipping options in `shipping.jsp` (see Figure 4-11).

**Figure 4-11 Shipping Page Detour**



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

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.

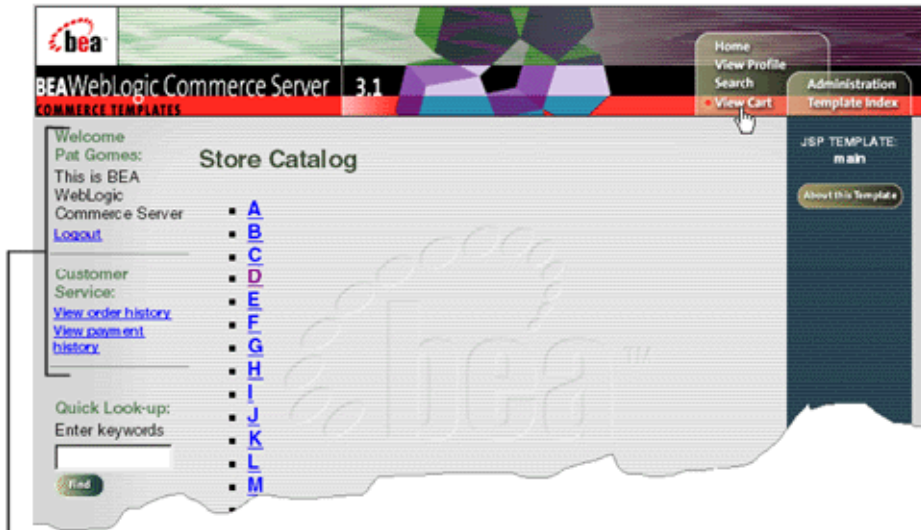
To take the detour, click Home (see Figure 4-11).

## **Detour: Review the Home Page for Authenticated Users**

When you click Home, WebLogic Commerce Server displays `main.jsp` (see Figure 4-12).

## Features of main.jsp for Authenticated Users

Figure 4-12 Home Page for Authenticated Users



Displayed only 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. Listing 4-2 shows that a Pipeline tag and a Java scriptlet determines whether a username is associated with the current Pipeline session. If a name is associated with the Pipeline session, User Management JSP tags query the database for your username and display the results.

### Listing 4-2 Conditional Display of Information

```
<%-- Get the username --%>  
  
<pipeline:getPipelineProperty propertyName="<%=  
PipelineSessionConstants.USER_NAME %%"
```

```
returnName="userName"
returnType="String"
/>

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

<!-- Tag to get customer profile -->

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

<p><font color="#567856"><b>Welcome<br>

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

<p>This is BEA WebLogic Commerce Server</p>

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

---

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 Commerce Server displays shipping options in `shipping.jsp` (see Figure 4-11).

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

To continue the tour...

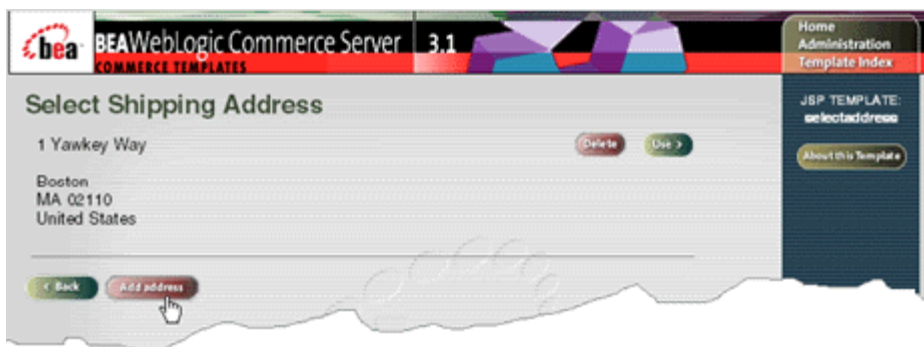
From `shipping.jsp`, click Continue.

## Step 4: Enter and Select a New Shipping Address

When you click Continue in `shipping.jsp`, WebLogic Commerce Server displays shipping addresses in `selectaddress.jsp` (see Figure 4-13).

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

**Figure 4-13** Select Add Address



For a description of `selectaddress.jsp`, see “Features of `selectaddress.jsp`” on page 3-19.

To continue the tour...

Click Add Address.

When you click Add Address, WebLogic Commerce Server prompts you to enter a new shipping address in `addaddress.jsp` (see Figure 4-14).



Figure 4-14 Enter the New Address

The screenshot shows a web browser window displaying the 'Add Shipping Address' form. The page header includes the BEA logo and 'BEA WebLogic Commerce Server 3.1'. The form fields are as follows:

Street address	<input type="text" value="23 Bronco Blvd"/>	*
Address 2	<input type="text"/>	
City	<input type="text" value="Denver"/>	*
State	<input type="text" value="CO"/>	*
Zip/Postal Code	<input type="text" value="80212"/>	*
Country	<input type="text" value="United States"/>	*

Fields marked with (\*) are required.

Navigation buttons: Back, Save (highlighted with a mouse cursor).

Right sidebar: Home, Administration, Template Index, JSP TEMPLATE: addaddress, About this Template.

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 Commerce Server displays the addresses from your user profile in `selectaddress.jsp` (see Figure 4-15).

Figure 4-15 Select the New Address



For a description of `selectaddress.jsp`, see “Features of `selectaddress.jsp`” on page 3-19.

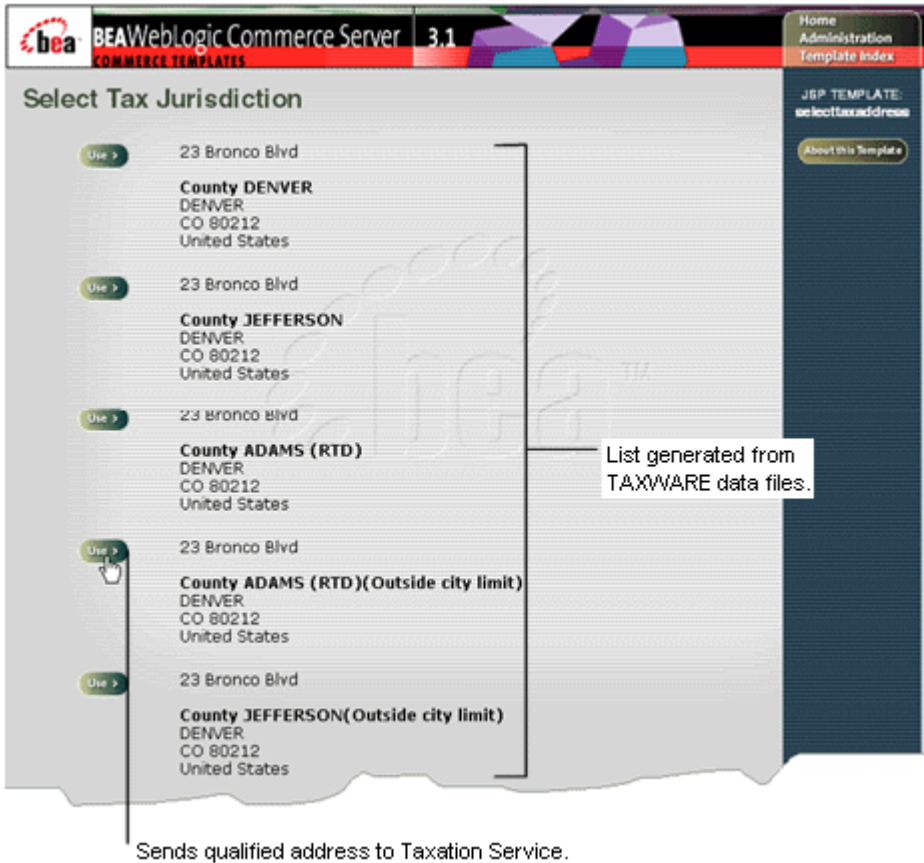
To continue the tour...

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

When you click Use, the Pipeline sends data to the **Taxation Service**, which uses TAXWARE to determine the tax jurisdiction and amount, based on the shipping address. Because the zip code 80212 covers different tax jurisdictions, WebLogic Commerce Server prompts you to choose a specific area within the zip code from `selecttaxaddress.jsp` (see Figure 4-16).

## Features of selecttaxaddress.jsp

Figure 4-16 Select Tax Jurisdiction



Note the following features of `selecttaxaddress.jsp`:

- The file's full pathname is  
`$WL_COMMERCE_HOME/server/webapps/wlcs/commerce/order/selecttaxaddress.jsp` (UNIX)
- It uses a Pipeline JSP tag to collect the tax jurisdiction data from the Pipeline session (which contains information from TAXWARE data files).

## 4 Registered User Buys a Product

---

- It uses Java scriptlets and HTML to display and format the data.
- Clicking Use sends data to the WebLogic Commerce Server **Taxation Service** for tax calculation.

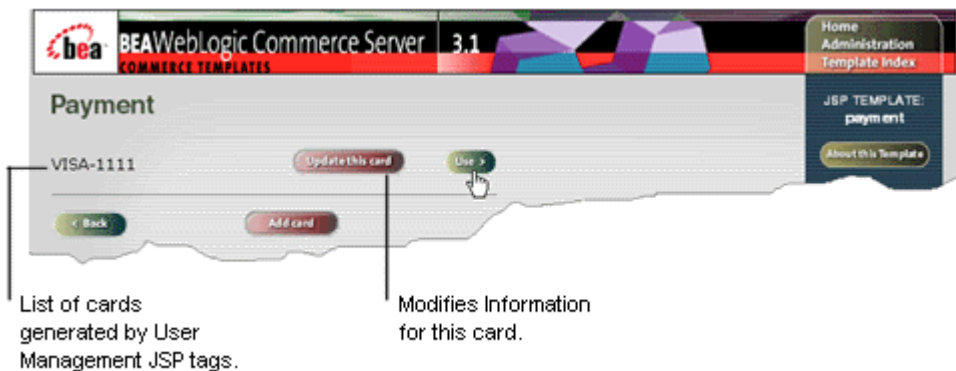
To continue the tour...

Click Use next to an address.

## Step 5: Select a Credit Card

When you click Use, WebLogic Commerce Server displays `payment.jsp`, which lists the credit cards in your user profile (see Figure 4-17).

**Figure 4-17** 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 Commerce Server displays the final purchase information in `checkout.jsp` (see Figure 4-18).

**Figure 4-18 Purchase the Items**

The screenshot displays the 'Final Checkout Review' page. At the top, there is a navigation bar with links for Home, View Profile, Search, View Cart, Administration, and Template Index. The main content area is divided into sections: Shipping Destination, Shipment Splitting Preferences, Special Instructions, and Ship Via. Below these is an 'Order' table with columns for ID, Description, Quantity, Our Price, and SubTotal. A summary section at the bottom right shows Shipping & handling, Total tax, and Total due. A 'Complete purchase' button is located at the bottom center, with a mouse cursor pointing to it. A text box on the left side of the summary section contains the message: 'Slows tax information received from the Taxation Service.'

ID	Description	Quantity	Our Price	SubTotal
9-10144	drill; power; craftsman; variable rpm; 3/8 drive; portable electric tools, drill	1	USD 62.95	USD 62.95

Shipping & handling USD 4.95  
 Total tax USD 2.58  
**Total due USD 70.48**

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 Commerce Server completes the purchase and displays the confirmation number in `confirmorder.jsp` (see Figure 4-19).

**Figure 4-19 Review the Order Confirmation**

**Confirm Order**

Thank you for shopping here with BEA WebLogic Commerce Server 3.1. Don't forget to stop back for more great deals, contests, new store openings and specials on jsp templates.

**Order Confirmation #4012**

**Will be shipped to:**  
Demo Customer  
One Winthrop Square  
BOSTON  
MA-02110

**Preferences:**  
Ship all at once

ID	Description	Quantity	Unit Price	Total
9-NF37761	file set-9-NF37761	1	USD 111.95	USD 111.95
<b>Shipping</b>				USD 4.95
<b>Total Tax</b>				USD 5.85
<b>Total Due</b>				USD 122.75

[Continue shopping](#)

BEA WebLogic Commerce Server 3.1  
Copyright © 1999, 2000 BEA Systems, Inc.

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 the next section, Chapter 5, “Registered User Adds a Credit Card.”





# 5 Registered User Adds a Credit Card

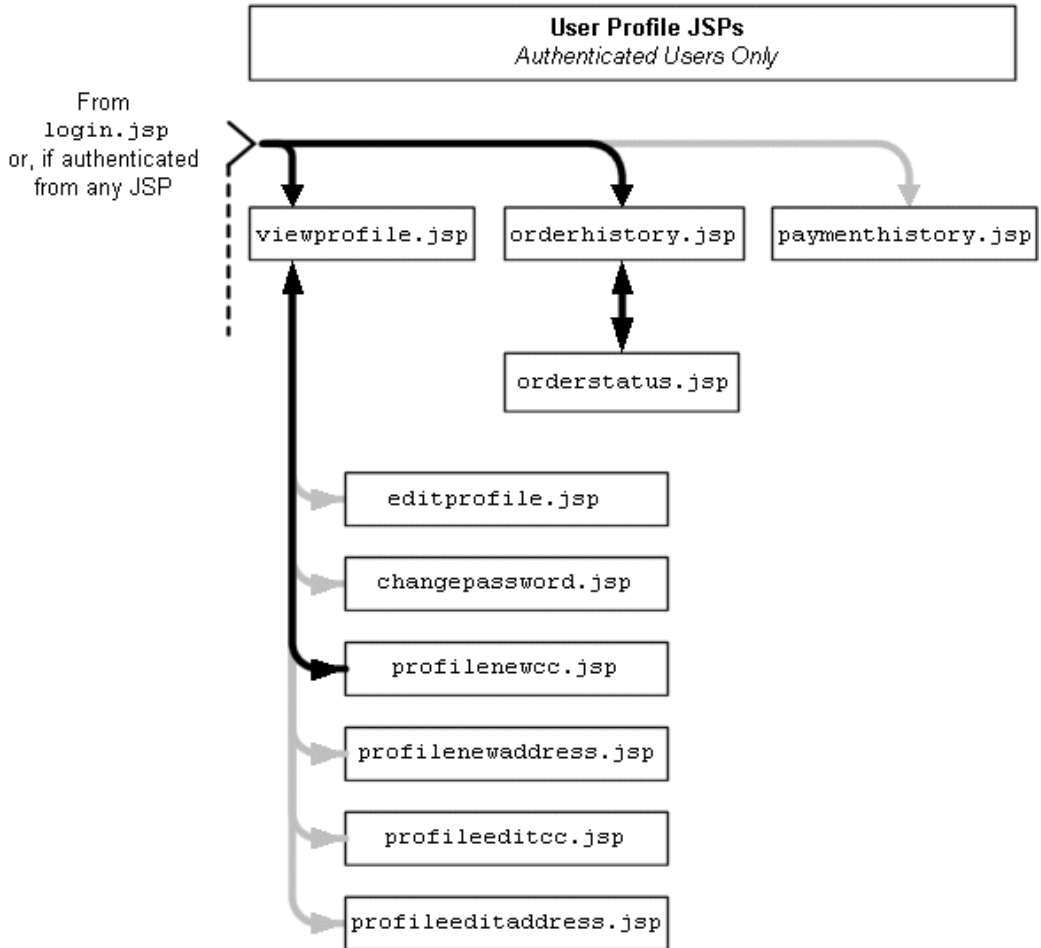
In this section of the tour, assume that you want to access the Web site to add credit card information to your user profile and to see the status of your last order. (This probably is not a realistic scenario for your customers, but it highlights additional features that are available in the JSP templates.) 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 header (see Figure 5-1). If you are accessing the site anonymously WebLogic Commerce Server prompts you to log in before it displays your profile.

Figure 5-2 Click View Profile



To continue the tour...

From any page on the site, click View Profile in the header. Log in if WebLogic Commerce Server prompts you.

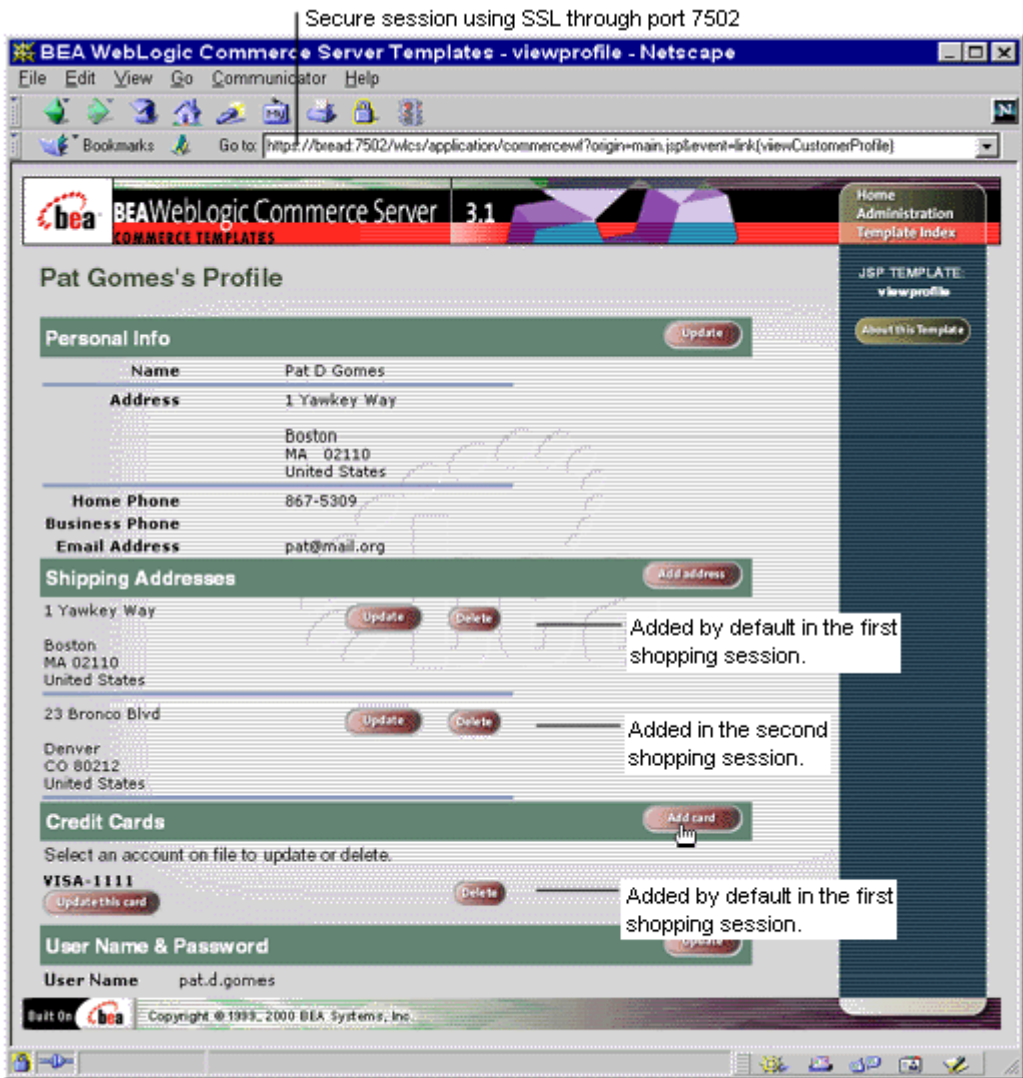
For information about WebLogic Commerce Server security, see “Technical Note: The Security Model” on page 4-12.

## 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 Commerce Server displays `viewprofile.jsp` (see Figure 5-3).

## Features of viewprofile.jsp

Figure 5-3 Your User Profile



Notice the following features of `viewprofile.jsp`:

- The file's full pathname is  
`$WL_COMMERCE_HOME/server/webapps/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.

### Technical Note: The Database for WebLogic Commerce Servers

WebLogic Commerce Server and WebLogic Personalization Server store their data in a single Cloudscape or Oracle database. For a description of the database schema and tables, refer to the following sections:

- “[The Product Catalog Schema](#)” in *Product Catalog Management*
- “[Overview of the Registration and User Processing Package](#)” in *Registration and User Processing Package*
- “[WebLogic Personalization Server Schema](#)” in *WebLogic Personalization Server Developer's Guide*

To continue the tour...

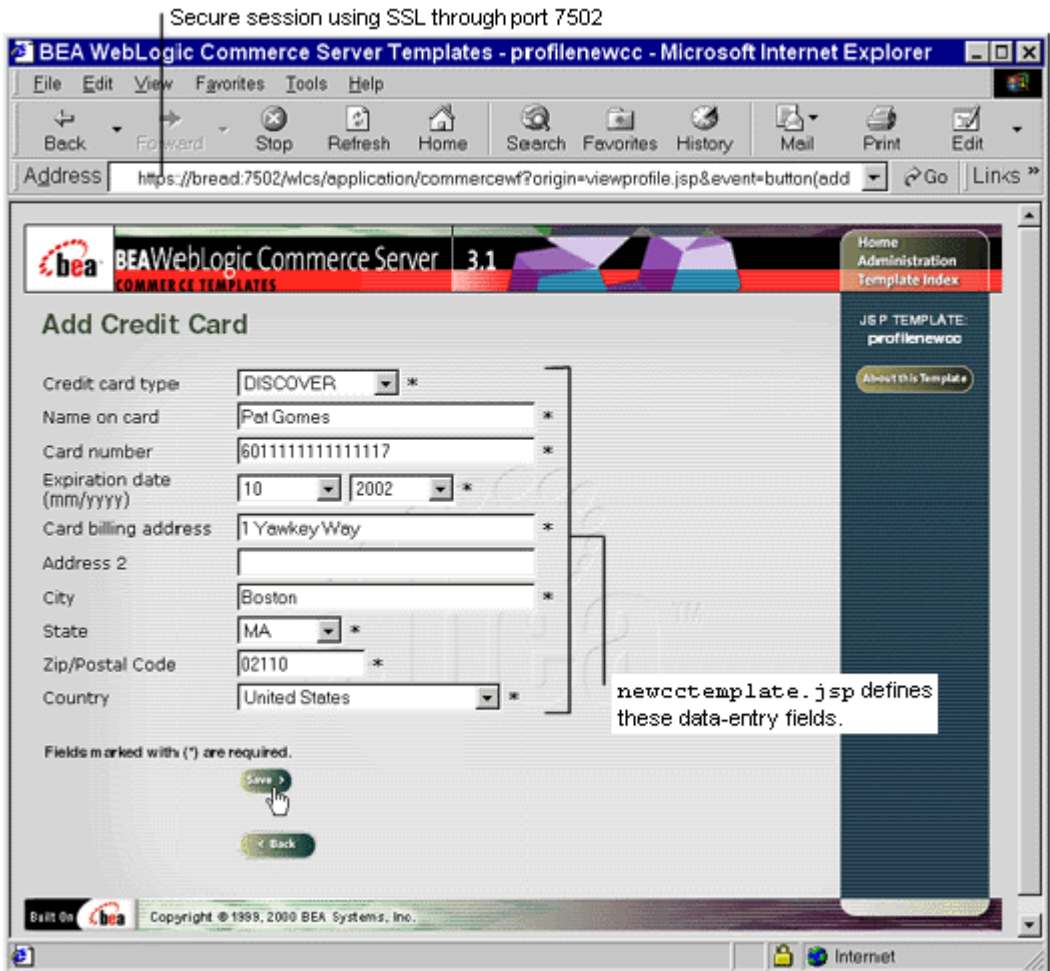
Click Add Card.

## Step 3: Add a Credit Card

When you click Add Card, WebLogic Commerce Server prompts you to enter credit card information in `profilenewcc.jsp` (see Figure 5-4).

## Features of profilnewcc.jsp

Figure 5-4 Adding a Credit Card



Notice the following features of `profilenewcc.jsp`:

- The file's full pathname is  
`$WL_COMMERCE_HOME/server/webapps/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.

If you have enabled the CyberCash integration, you can use any of the following credit card types and numbers to test or demonstrate the integration:

- VISA: 4111111111111111 (one four and 15 ones)
- MasterCard: 5555555555554444 (12 fives and four fours)
- Amex: 378282246310005
- Discover: 601111111111117

For more information about the CyberCash integration, see “About CyberCash” on page 1-4.

If you have not enabled the CyberCash integration, you can use the numbers and card types in the previous list or enter 16-18 characters for any card type. For the Card Number box, the sample JSP templates validate only the number of characters that you enter.

2. Complete the remaining required fields. Then click Save.

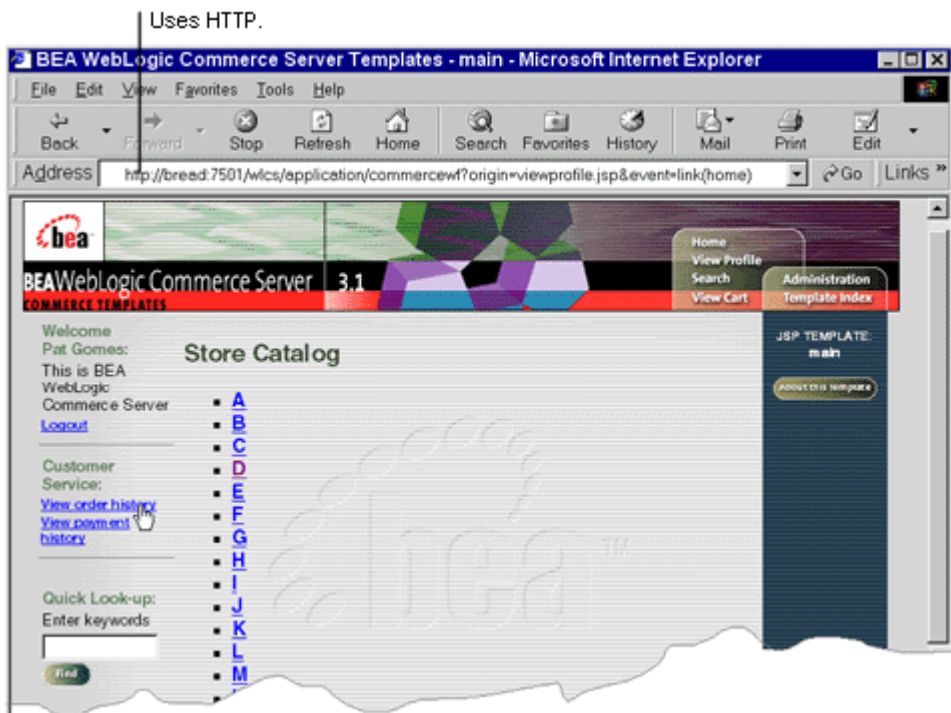
The next part of the tour shows your order history and status, which you access from the Home page for authenticated users.

3. Click Home in the header.

## Step 4: View Your Order History and Status

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

Figure 5-5 Click View Order History



For details about `main.jsp` for authenticated users, see “Features of `main.jsp` for Authenticated Users” on page 4-14.

To continue the tour...

Click View Order History, WebLogic Commerce Server displays `orderhistory.jsp` (see Figure 5-6).



## Features of orderhistory.jsp

Figure 5-6 Order History

Date	Order Number	Amount
2000-09-07	1	USD 335.90
2000-09-08	4002	USD 149.05

Order confirmation number, which is stored in the WLCS\_ORDER database table.

Notice the following features of `orderhistory.jsp`:

- The file's full pathname is  
`$WL_COMMERCE_HOME/server/webapps/wlcs/commerce/order/orderhistory.jsp` (UNIX)
- A Pipeline JSP tag retrieves the history:
 

```
<pipeline:getPipelineProperty
  propertyName="<%=PipelineSessionConstants.ORDER_HISTORY%>"
  returnName="orderHistory" returnType="java.util.List"/>
```

WebLogic Commerce Server stores this history in the WLCS\_ORDER database table until the database administrator archives the data. For more information about the WLCS\_ORDER table, see [“The Order Processing Database Schema”](#) in *Order Processing Package*.

To continue the tour...

Click View to see the order status.

When you click View, WebLogic Commerce Server displays `orderstatus.jsp` (see Figure 5-7).

## Features of orderstatus.jsp

Figure 5-7 Order Status

Order Status

Confirmation number 4002

Order status Submitted

Date ordered 2000-09-11

Splitting preferences Ship all at once

Special instructions Please leave package under the doormat.

Shipping address 1 Yawkey Way  
BOSTON  
MA-02110

ID	Description	Quantity	Unit Price
9-MKHR2410	hammer-9-MKHR2410	1	USD284.85
<b>Shipping &amp; handling</b>			USD 4.95
<b>Total tax</b>			USD 14.49
<b>Total due</b>			USD 304.29

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

Notice the following features of `orderhistory.jsp`:

- The file's full pathname is  
`$WL_COMMERCE_HOME/server/webapps/wlcs/commerce/order/orderstatus.jsp` (UNIX)
- A Pipeline JSP tag collects properties of the order:
 

```
<pipeline:getPipelineProperty
propertyName="<%=PipelineSessionConstants.SELECTED_ORDER%>"
returnName="orderValue"
returnType="com.beasys.commerce.ebusiness.order.OrderValue"/>
```

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.

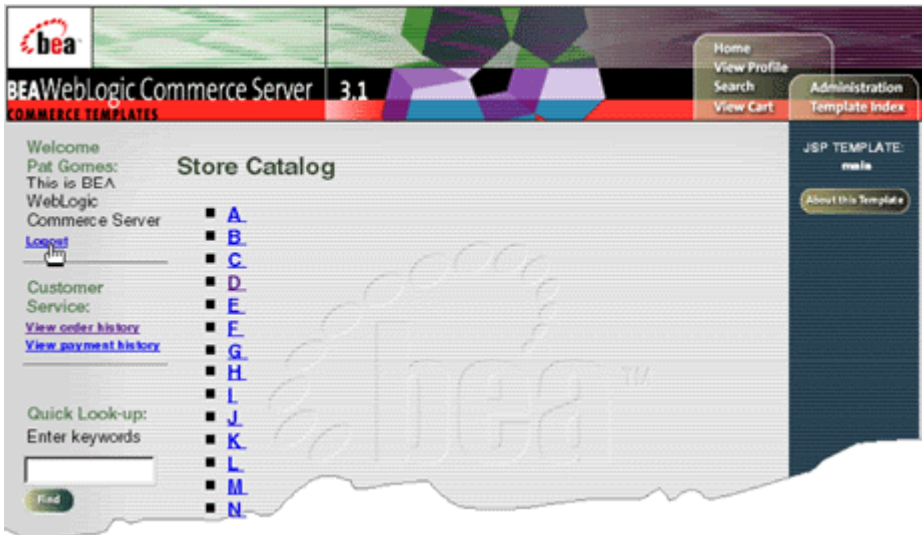
To continue the tour...

Click Home.

## Step 5: Log Out

When you click Home, WebLogic Commerce Server displays the Home page for authenticated users (see Figure 5-8).

**Figure 5-8 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 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, WebLogic Commerce Server displayed `shoppingcart.jsp` to prompt you to check out. Yet, your business model might encourage users 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 changing only a single line in the `webflow.properties` file, but before you change the line, you need background information to understand which line to modify and why. 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 Commerce Server 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 `$WL_COMMERCE_HOME/server/webapps/wlcs/commerce/catalog/includes/itemdetails.jsp`. For more information on `itemdetails.jsp`, see “[The Catalog JSP Templates](#)” in *Product Catalog Management*.

### Listing 6-1 Add to Cart Anchor

---

```
<a href="<%= WebflowJSPHelper.createWebflowURL(pageContext, "itemdetails.jsp",  
"link(add)", "&" + HttpRequestConstants.CATALOG_ITEM_SKU + "=" +  
productItem.getKey().getIdentifier(), true) %>">  
  
" border="0" alt="Add to Shopping  
Cart"></a>
```

---

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 `WebflowJSPHelper.createWebflowURL()` method to pass data to Webflow. Note the following two data items that `WebflowJSPHelper.createWebflowURL()` passes to Webflow:

- The name of the JSP file that contains the link source. For the Add to Cart link, the JSP name is `itemdetails.jsp`.
- The event type and name. For the Add to Cart link, the event type is `link` and the name is `(add)`.

For more information about `WebflowJSPHelper.createWebflowURL()` method, see “Customizing Webflow and Pipelines” in *Webflow and Pipeline Management*.

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

## How Webflow Resolves the Link

When the `WebflowJSPHelper.createWebflowURL()` method sends data to Webflow, Webflow opens the `$/WL_COMMERCE_HOME/webflow.properties` file to find instructions for processing the data.

Listing 6-2 shows that `webflow.properties` directs the data it receives from `link(add)` to the `AddProductItemToShoppingCart.inputprocessor` input processor. (In `webflow.properties`, anything to the left of an equals sign is sending data to an input processor, Pipeline, Pipeline component, or other service, and anything to the right of an equals sign is receiving data.)

### Listing 6-2 The Add to Cart Link in `webflow.properties`

---

```
#####  
Handle add to shopping cart.  
#####  
  
# On the add link, extract parameters for the add to shopping cart pipeline  
*.jsp.link(add)=AddProductItemToShoppingCart.inputprocessor
```

---

**Note:** The `*.jsp.link(add)` construct indicates that Webflow applies this property to any link named `link(add)` on any JSP.

Listing 6-3 (which continues the `webflow.properties` excerpt started in Listing 6-2) shows that the input processor sends the link output into a Pipeline named `AddProductItemToShoppingCart.pipeline`. After this Pipeline processes data from the link output, it sends data to another Pipeline named `RefreshSavedList.pipeline`.

## 6 Example of Customizing the Templates

---

### Listing 6-3 Sending Data from Input Processor to Pipelines

---

```
# Execute the add to shopping cart pipeline
AddProductItemToShoppingCart.inputprocessor.success=AddProductItemToShoppingCart.pipeline

# If add operation succeeded, we want to make sure the saved list is refreshed before
RefreshSavedList.pipeline

# going to shopping cart screen
AddProductItemToShoppingCart.pipeline.success=RefreshSavedList.pipeline
```

---

To find out how Webflow specifies which JSP to display after you click Add to Cart, you must continue following the Webflow trail to see where `RefreshSavedList.pipeline` directs its output.

The next instance of the `RefreshSavedList.pipeline` string, displayed in Listing 6-4, shows the Pipeline **receiving** data from another input processor; this represents another path for sending data into the Pipeline. However, we are interested in seeing where the Pipeline **sends** data, so we must search for an instance of the `RefreshSavedList.pipeline` string that is on the left side of an equals sign.

In this case, the `RefreshSavedList.pipeline=` string is located only three lines later in the file. It specifies that when the `RefreshSavedList.pipeline` pipeline succeeds, WebLogic Commerce Server displays `shoppingcart.jsp`.

### Listing 6-4 Resolving the Link

---

```
#####
# Handle view shopping cart.
#####
# First call InitShoppingCarts to make sure the pipeline session is initialized
*.jsp.link(shoppingcart)=customerhome_InitShoppingCart.inputprocessor
*.jsp.link(viewShoppingCart)=customerhome_InitShoppingCart.inputprocessor

# On success, execute the RefreshSavedList pipeline
customerhome_InitShoppingCart.inputprocessor.success=RefreshSavedList.pipeline

# if the pipeline succeeds go to shopping cart screen
RefreshSavedList.pipeline.success=commerce/shoppingcart.jsp
```



## Step 2: Modify Add to Cart

After locating the Webflow property that determines the user-visible behavior of Add to Cart, you can modify it to direct users 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`. But changing the output of `RefreshSavedList.pipeline.success` from `commerce/shoppingcart.jsp` to `commerce/main.jsp` would not produce desirable results.

`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 `RefreshSavedList.pipeline.success` to display `main.jsp` directly, the Web browser would display an empty page. Instead, you must invoke an input processor named `GetTopCategories.inputprocessor`, which gathers data about categories in the product catalog and displays it in `main.jsp`.

For more information on how WebLogic Commerce Server populates `main.jsp` with data, see “[The Product Catalog JSP Templates and Tag Library](#)” in *Product Catalog Management*.

### Directing the Output of the GetTopCategories Input Processor

You can direct output from `RefreshSavedList.pipeline` to `GetTopCategories.inputprocessor` by either using the Webflow and Pipeline Editor or by modifying the `$WL_COMMERCE_HOME/webflow.properties` file using a text editor such as Notepad. However, if you modify the properties files using a text editor and invalid entries are created, the behavior of the editor may be unpredictable. The Webflow and Pipeline Editor will only work with a valid Webflow or Pipeline properties file. A description of both methods are listed below:

- If you use the Webflow and Pipeline Editor, you must modify the destination of the success event for the input processor’s origin. The input processor origin is `RefreshSavedList.pipeline.success`. The success event’s destination that you need to set is `GetTopCategories.inputprocessor`. For more information

about modifying input processors, see “Using the Webflow and Pipeline Editor” in *Webflow and Pipeline Management*.

- If you use a text editor, open the `$WL_COMMERCE_HOME/webflow.properties` file and change the value on the right side of the equals sign as illustrated in Listing 6-5. You can change the previous property to a comment line, making it easy to return to your previous configuration.

### Listing 6-5 Modifying `webflow.properties`

---

```
# if the pipeline succeeds go to shopping cart screen
# RefreshSavedList.pipeline.success=commerce/shoppingcart.jsp
RefreshSavedList.pipeline.success=GetTopCategories.inputprocessor
```

---

## Step 3: Verify the Modification

After you save your modifications to `webflow.properties`, 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 WebLogic Commerce Server.

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



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

To verify that WebLogic Commerce Server added your item to the shopping cart, do the following from `main.jsp`:

1. Click View Cart in the header.
2. If WebLogic Commerce Server 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 Commerce Server displays `shoppingcart.jsp`, which contains the item you added to the cart (see Figure 6-2).

Figure 6-2 The Shopping Cart



## What's Next?

This concludes the tour. To learn more about the WebLogic Commerce Server JSP templates, Webflow, Pipeline, the product catalog, and the other concepts introduced in the tour, refer to the following documents:

- [Webflow and Pipeline Management](#)
- [Product Catalog Management](#)
- [Registration and User Processing Package](#)
- [Order Processing Package](#)

---

# Index

## A

- Add to Cart link
  - anchor 6-2
  - changing 6-1
  - modifying 6-5
  - verifying the modification 6-6

## B

- browse.jsp
  - features 4-6, 4-8
  - UNIX syntax 4-6
- buttons
  - Add Credit Card 3-21
  - Add New Address 4-16
  - Add to Cart 3-7, 4-11, 6-1
  - Check Out 3-9, 4-12
  - Checkout 3-17
  - Create 3-11
  - Find 3-5
  - Purchase 3-28
  - Use 3-20, 4-18

## C

- checkout link
  - displaying 3-17
- checkout.jsp
  - features 3-27
  - UNIX syntax 3-27
- Cloudscape database 5-5
  - installing WLS 1-3
- confirmorder.jsp
  - features 3-29
  - UNIX syntax 3-29
- customer support contact information xi
- CyberCash
  - about 1-4
  - checkout.jsp 3-28
  - credit card support 3-24, 5-7

## D

- details.jsp
  - Add to Cart link 6-6
  - features 4-10

---

UNIX syntax 4-10

documentation, where to find it x

## H

home page

for authenticated users 4-13

main.jsp features 2-4

UNIX platform 2-1

Windows platform 2-2

how to

access your user profile 5-2

add a credit card 5-5

add a new user 3-11

browse the catalog 4-5

check out 4-11

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 in as a registered user 3-15

log out 5-11

purchase an item 3-26, 4-21

review the order confirmation 3-28, 4-22

review the shopping cart 4-11

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 5-8

view your order status 5-8

## 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 3-22

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, 4-11

templates included within 3-22, 3-23

user profile roadmap 5-1

viewprofile.jsp 5-3

## L

login.jsp

features 3-10, 3-15

UNIX syntax 3-10

---

## M

### main.jsp

- features 2-4, 2-7
- for authenticated users 4-14
- importing Java packages 2-6
- importing tag libraries 2-6
- modifying Add to Cart link 6-5
- searching for products 3-5
- UNIX syntax 2-5
- Windows syntax 2-4

## N

### newuser.jsp

- features 3-11, 3-14
- UNIX syntax 3-11

## O

### Oracle database 5-5

- installing WLS 1-3

### 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

printing product documentation x

profilenewcc.jsp

features 5-6

UNIX syntax 5-7

## R

registered user 4-12

related information xi

## S

### searchresults.jsp

features 3-6

UNIX syntax 3-6

security model 4-12

### selectaddress.jsp

features 3-19

Java scriptlets 3-20

UNIX syntax 3-19

### selecttaxaddress.jsp

features 4-19

UNIX syntax 4-19

### shipping.jsp

features 3-18

UNIX syntax 3-18

### shoppingcart.jsp

features 3-7

UNIX syntax 3-7

### successfullogin.jsp

features 3-16

UNIX syntax 3-16

### support

technical xi

## T

### TAXWARE

about 1-4

---

selectaddress.jsp 3-19  
template include files 3-23

## **U**

User Management 3-26

## **V**

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

## **W**

Webflow 3-8  
    defined 3-8  
Webflow and Pipeline Editor 6-5  
WebLogic Commerce Server  
    product catalog 2-8  
    UNIX home page 2-1  
    Windows home page 2-2  
WL\_COMMERCE\_HOME, description of  
    1-3  
WLCS\_ORDER database table 5-9