

IBM WebSphere Commerce – Express



Easy Start Guide

Version 5.5

Note!

Before using this information and the product it supports, be sure to read the general information in the Notices section.

First Edition first revision (October 2003)

This edition applies to IBM WebSphere Commerce — Express Edition Version 5.5 (product number 5724-A18), and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

© Copyright International Business Machines Corporation 2003. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

About this book

Document description

This guide provides step—by—step information about IBM® WebSphere® Commerce — Express. Use this guide to set up your store using the Express Store sample store. WebSphere Commerce — Express Edition also includes the FashionFlow sample store, for information on setting up the FashionFlow store refer to the *WebSphere Commerce Sample Store Guide*.

IBM welcomes your comments. You can send your comments by using the online IBM WebSphere Commerce documentation feedback form, available at the following URL:

<http://www.ibm.com/software/webservers/commerce/rcf.html>

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Where to find information

WebSphere Commerce has online and hardcopy information describing the complete e-commerce solution. In addition, the software products that are bundled with WebSphere Commerce provide further information, describing the specific features and functions of the software. This section provides a quick overview of where to locate the various types of information.

WebSphere Commerce publications

The following list describes the WebSphere Commerce product library.

Installation, configuration and migration

- *WebSphere Commerce Quick Beginnings*, Version 5.5
This book describes how to install and configure the main components of WebSphere Commerce 5.5 on a single machine. A printed version of this book is included with the WebSphere Commerce product. The *WebSphere Commerce Quick Beginnings* is specific to edition and platform.
- *WebSphere Commerce Installation Guide*, Version 5.5
This book describes how to install and configure WebSphere Commerce on various topologies. It is intended for system administrators or for anyone else responsible for performing installation and configuration tasks. Installation guides are specific to edition and platform.
- *WebSphere Commerce Migration Guide*, Version 5.5
There are two migration guides available for WebSphere Commerce — Express: *WebSphere Commerce - Express Migration Guide*, and *WebSphere Commerce - Express Developer Edition Migration Guide*. These guides explain how to migrate from the WebSphere Commerce Version 5.4 Professional Entry Edition, to WebSphere Commerce — Express Version 5.5.
- *WebSphere Commerce Additional Software Guide*, Version 5.5

This document contains instructions for installing and configuring several optional software packages that can be used with WebSphere Commerce and that are contained in it. A single version of this guide is available for all editions.

Store development

- *WebSphere Commerce Programming Guide and Tutorials*, Version 5.5
This guide provides information about the WebSphere Commerce architecture and programming model. It also provides some programming tutorials. The *WebSphere Commerce Programming Guide and Tutorials* should be used if you need to customize or extend business logic. A single version of this guide is available for all editions.
- *WebSphere Commerce Store Development Guide*, Version 5.5
The *WebSphere Commerce Store Development Guide* provides information about the WebSphere Commerce store architecture and the store development process. In particular, it provides details on the following topics:
 - Store development options
 - The store archive
 - Store development tools
 - Developing your storefront
 - Developing your store data
 - Store data architecture
 - Store data information model
 - Adding access control to your store
 - Packaging your store
 - Publishing your store
 - Adding WebSphere Commerce features to your store

A single version of this guide is available for all editions. For the WebSphere Commerce — Express edition this guide should be used primarily for creating your catalog data.

- *WebSphere Commerce Sample Store Guide*, Version 5.5
This guide provides information about the sample stores included in WebSphere Commerce. In particular, it provides details on the following topics:
 - Business models and the samples
 - Sample store or site conventions
 - Overview, setup, and features information
 - Tutorials and use cases

The *WebSphere Commerce Sample Store Guide* does not describe the Express Store. If you will publish the FashionFlow store, follow the directions in the *WebSphere Commerce Sample Store Guide*; otherwise, follow the directions in the *Easy Start Guide*.

Administration

- *WebSphere Commerce Fundamentals*, Version 5.5
Provides an overview of WebSphere Commerce Business Edition and WebSphere Commerce Professional Edition. This book, the *Easy Start Guide*, replaces the *Fundamentals Guide* for the WebSphere Commerce — Express Edition.

- *WebSphere Commerce Administration Guide, Version 5.5*
This book is intended for administrators and describes how to administer WebSphere Commerce components using the Administration Console, and other administration tools. A single version of this book is provided for all platforms and editions.
- *WebSphere Commerce Security Guide, Version 5.5*
This document describes the security features of WebSphere Commerce and how to configure these features. It details WebSphere Commerce security issues and features such as authentication, authorization, and access control policies.
The objective of this document is to provide the persons responsible for security at your site (which likely includes a system administrator or WebSphere Commerce Site Administrator) with a comprehensive document to enable them to reliably secure a WebSphere Commerce production site.
The intended audience for this document is the chief security officer or the security administrator for a WebSphere Commerce site.

WebSphere Commerce Payments

- *WebSphere Commerce Payments Programming Guide and Reference, Version 5.5*
This book is for programmers who are responsible for developing applications that communicate and interact with the WebSphere Commerce Payments component of WebSphere Commerce. Programmers who develop payment cassettes for use with WebSphere Commerce Payments may find this information useful.
- *WebSphere Commerce Payments Cassette Kit Programming Guide, Version 5.5*
This book explains how to write payment cassettes for the Payments component of IBM WebSphere Commerce, Version 5.5. It describes the WebSphere Commerce Payments cassette programming interface, cassette-specific programming or administrative considerations, and shows how the cassette implements the WebSphere Commerce Payments' various interfaces. This book is for programmers who will develop payment cassettes for WebSphere Commerce Payments. Cassette developers should be experienced Java programmers with a strong background in the field of electronic payment processing.

WebSphere Commerce Analyzer

- *IBM WebSphere Commerce Analyzer Installation and Configuration Guide*
This book provides information about installing and configuring *IBM WebSphere Commerce Analyzer Datamart Reference, Version 5.5*.
- *IBM WebSphere Commerce Analyzer Datamart Reference*
This book provides reference information about the *IBM WebSphere Commerce Analyzer Datamart Reference, Version 5.5* (also called WebSphere Commerce Analyzer) datamart.
 - Business questions and the datamart
 - Removing the datamart
 - Recreating the datamart
 - Customizing the datamart
- *IBM WebSphere Commerce Analyzer Technical Reference*

This book provides information for system administrators and marketing analysts about using IBM WebSphere Commerce Analyzer, Version 5.5. After you install, configure, and perform the post-configuration setup of the WebSphere Commerce Analyzer server, use this book for information about using WebSphere Commerce Analyzer, which includes topics such as:

- How to perform regular maintenance activities on the WebSphere Commerce Analyzerserver
- How to improve the performance of the WebSphere Commerce Analyzer server
- How to customize WebSphere Commerce Analyzer

For updates to these publications, refer to the WebSphere Commerce Technical Library page:


Technical Library (<http://www.ibm.com/software/commerce/library>)

WebSphere Commerce - Express development environment online help

The online help applicable to developers, such as API and command reference information, is fully integrated with WebSphere Commerce - Express development environment. The online help system included in the WebSphere Commerce - Express development environment includes several enhancements, including the ability to bookmark topics and advanced search capabilities. Optionally, you may run this online help system as an Infocenter, that is you can serve the help without running the WebSphere Commerce - Express development environment. This help system also includes the help included in the WebSphere Commerce help system.

To view the WebSphere Commerce - Express development environment online help, do the following:


1. Start WebSphere Commerce - Express development environment by selecting **Start > Programs > WebSphere Commerce - Express Developer Edition > WebSphere Commerce development environment**.
2. From the **Help** menu, select **Help Contents**.

Note: If you refer to instructions for multiple platforms in the WebSphere Commerce - Express development environment, ensure you follow the instructions for WebSphere Commerce - Express development environment. When a help page contains information for multiple platforms, information specific to WebSphere Commerce - Express development environment is indicated with the following icon:  .

WebSphere Commerce online help

The WebSphere Commerce help system has been tailored for Site Administrators and business users. This help system describes how to use the various browser based tools provided with WebSphere Commerce. The WebSphere Commerce online help consists of online information that can be viewed using a Web browser. The online help can be accessed from a Web browser that runs on Internet Explorer, Version 5.5, or higher using the following address:

   https://host_name:8000/wchelp/

 http://host_name/wcsdoc/locale/index.htm

where:

host_name

The name of your WebSphere Commerce machine.

locale The language in which you want the help to display.

de_DE	German
en_US	US English
es_ES	Spanish
fr_FR	French
it_IT	Italian
ja_JP	Japanese
ko_KR	Korean
pt_BR	Brazilian Portuguese
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Further information on the Web

WebSphere Commerce Support

To find support information, including newsgroups, FAQs, technical notes, troubleshooting information and downloads, visit the following Web addresses:

WebSphere Commerce Support Web page
(<http://www.ibm.com/software/commerce/support/>)

WebSphere Commerce Technical Library Web page
(<http://www.ibm.com/software/commerce/library/>)

Software partners

There are many software partners that offer products and services to enhance WebSphere Commerce. For information about these partners, visit the Commerce Zone (<http://www.ibm.com/software/wsdd/zones/commerce/>) and click the Software Developers link.

Redbooks™

To find more advanced technical information, visit the Redbooks Web site (<http://www.ibm.com/redbooks>) and search for WebSphere Commerce.

Updates to this book

To learn about last-minute changes to the product, refer to the README file in the root directory of WebSphere Commerce CD 1. In addition, a copy of this book, and any updated versions of this book, are available as PDF files from the WebSphere Commerce Technical Library:

<http://www.ibm.com/software/commerce/library/>

Updated versions of this book are also available from the WebSphere Commerce Zone at WebSphere Developer Domain which is at the following Web site:

<http://www.ibm.com/software/wsdd/zones/commerce/>

Updates from the last version of this book are identified by revision characters contained in the margin. This book uses the following conventions for revision characters:

- The "+" (plus) character identifies updates that have been made in the current version of this book.

The following table shows the main changes that have been made to this book.

Additional information about VisaNet, BankServ and Paymentech cassettes.	See the section "Installing and enabling WebSphere Commerce Payments" on page 84.
Deploying the Express Store	See the section Appendix D, "Deploying the Express Store," on page 185.

Conventions used in this book

This book uses the following highlighting conventions:

Boldface type

Indicates commands or graphical user interface (GUI) controls such as names of fields, icons, or menu choices.

Monospace type

Indicates examples of text you enter exactly as shown.

Italic type

Used to emphasize words. Italics also indicate names for which you must substitute the appropriate values for your system.

host_name

The fully qualified host name of the WebSphere Commerce - Express development environment server (for example, server1.torolab.ibm.com is fully qualified).

instance_name

The name of the WebSphere Commerce - Express development environment instance with which you are working.

▶ Windows

Indicates information specific to programs running on Windows®.

▶ Linux

Indicates information specific to programs running on Linux.

▶ 400

Indicates information specific to programs running on OS/400®.

▶ Developer

Indicates information specific to running WebSphere Commerce within the WebSphere Commerce - Express development environment.

Path variables

This guide uses the following variables to represent directory paths:

DB2_installdir

This is the installation directory for DB2 Universal Database™. The default installation directory for DB2 Universal Database installed by the WebSphere Commerce - Express development environment installation wizard is C:\WebSphere\SQLLIB.

WCDE_installdir

This is the installation directory for the WebSphere Commerce - Express development environment. The default installation directory for the WebSphere Commerce - Express development environment is C:\WebSphere\CommerceDev55.

WCDE_workspacedir

This is the directory for the WebSphere Commerce - Express development environment workspace. The default WebSphere Commerce - Express development environment workspace directory is C:\WebSphere\workspace_db2.

WSAD_installdir

This is the installation directory for WebSphere Studio Application Developer. The default installation directory for WebSphere Studio Application Developer is C:\WebSphere\Studio5.

WC_installdir

This is the installation directory for WebSphere Commerce. The following are the default installation directories for WebSphere Commerce on various operating systems:

▶ 400 /QIBM/ProdData/CommerceServer55

▶ Linux /opt/WebSphere/CommerceServer55

▶ Windows C:\Program Files\WebSphere\CommerceServer55

▶ 400 *WC_userdir*

This is the directory for all the data that is used by WebSphere Commerce which can be modified or needs to be configured by a user. An example of such data is WebSphere Commerce instance information. This directory is unique to OS/400 .

The *WC_userdir* variable represents the following directory:

/QIBM/UserData/CommerceServer55

WAS_installdir

This is the installation directory for WebSphere Application Server. The following are the default installation directories for WebSphere Application Server on various operating systems:

▶ 400 /QIBM/ProdData/WebAS5

▶ Linux /opt/WebSphere/AppServer

▶ Windows C:\Program Files\WebSphere\AppServer

Variables

storedir

This variable represent the name of the store directory in which your store is located.

Knowledge requirements

This book should be read by system administrators or anyone else responsible for setting up a store using WebSphere Commerce — Express.

People who are setting up a WebSphere Commerce — Express store should have knowledge in the following areas:

- Your operating system
- Basic operating system commands
- DB2 Universal Database Express Edition
- Basic SQL commands

Developers who are developing store-front assets or back-office business logic with WebSphere Commerce - Express Developer Edition should have knowledge in the following areas:

- IBM WebSphere Commerce - Express
- IBM WebSphere Studio Application Developer
- IBM DB2 Universal Database Express Edition
- Java™ programming language
- Enterprise JavaBeans™ component architecture
- JavaServer Pages technology
- HTML
- XML

Contents

About this book	iii
Document description	iii
Where to find information	iii
WebSphere Commerce publications	iii
WebSphere Commerce - Express development environment online help	vi
WebSphere Commerce online help	vi
Further information on the Web	vii
Updates to this book	vii
Conventions used in this book.	viii
Path variables	viii
Variables	ix
Knowledge requirements	x

Part 1. Up and running with the Express Store 1

Chapter 1. Creating your store	3
Determining the business model.	3
Shopping flow in the consumer direct Express Store	4
Shopping flow in the B2B direct Express Store	5
Using the B2B access control policies	5
Prerequisite information	6
Publishing the Express Store	6

Chapter 2. Configuring the Express Store 11

Registration	11
Catalog.	13
Checkout	14
Orders	15
Payments	16
Order status	16
Changing store flows	17

Chapter 3. Replacing the Express Store default information with your store's information 19

Uploading the store's logo	19
Changing store information	20
Changing store style	20
Changing text on store pages and notification messages	21

Chapter 4. Loading the Express Store sample catalog 23

Catalog structure	23
Clothing store sample catalog	24
Hardware store sample catalog.	25
Catalog.	25
Catalog groups	25
Catalog entries	26

Attributes	26
Pricing	27
Marketing and merchandising features in the sample catalogs	27
The Loading process	28
Loading a sample catalog.	29
Loading the sample catalog in a development environment	29
Loading the sample catalog to WebSphere Commerce running on Windows, Linux or iSeries	30
Placing a test order.	31

Chapter 5. Shipping 33

Setting up shipping.	34
------------------------------	----

Chapter 6. Taxes 39

Setting up taxes	40
----------------------------	----

Chapter 7. E-mail notification 43

Setting up e-mail notification	43
--	----

Part 2. Loading store information 47

Chapter 8. Loading the catalog 49

Creating catalog data in XML format	49
Creating catalog data in CSV format	49

Chapter 9. Loading existing customer information 51

Part 3. Adding additional features to the Express Store. 55

Chapter 10. Store front architecture . . . 57

Default commands and views	58
Express Store conventions	58
JavaServer Pages (JSP).	58
JSP templates.	60
Properties files	60
Location structure and directory structure	61

Chapter 11. Changing store pages . . . 63

Commonly used data beans	63
Changing the Express Store Registration page	63
Modifying property files	64

Chapter 12. Adding a page to the store 67

Developing a list of store pages.	67
Determine the store shopping flow	67
Error pages	68
Developing a list of command and view URLs	68
Developing a list of URLs needed	68

Associating JSP filename to views	69
Set access control for a page	70
Caching your store pages	72
Planning your caching strategy	72
Implementing your caching strategy	74

Chapter 13. Changing store data 79

Shipment tracking URL	79
Changing exchange rates	80

Chapter 14. Managing your payments 83

Processing orders with WebSphere Commerce	
Payments enabled	83
Installing and enabling WebSphere Commerce	
Payments	84
Processing orders without enabling WebSphere	
Commerce Payments	88
Encrypting and decrypting credit card	
information	88

Chapter 15. Using e-Marketing Spots 91

Adding an e-Marketing Spot to a store page	91
--	----

Part 4. Running your store front 95

Chapter 16. Managing the catalog 97

Category management	97
Managing categories	98
Procedures for managing categories	98
Product management	98
Characteristics of products, SKUs, bundles and	
prebuilt kits	101
Creating products, attributes, and SKUs	102
Procedures for managing products	103

Chapter 17. Managing customer information 105

Customer registration information	105
Changing customer registration information	105
Procedures for customer registration	106

Chapter 18. Managing orders 107

Working with customer orders	107
Procedures for managing orders	109

Part 5. Marketing and merchandising 111

Chapter 19. Marketing 113

Discounts	113
Discount types	114
Procedures for managing discounts	115
e-Marketing Spots	117
Customer profiles	118
Customer profile management	118
Campaigns	119
Suggestive selling initiative	120
Suggestive selling initiative with conditions	120

Awareness advertisement	121
Campaign management	122

Chapter 20. Setting up an e-mail campaign 123

E-mail activity templates	123
Creating an e-mail activity template	123
Registering an e-mail activity template	125
Configuring e-mail activity accounts	126
Creating a new e-mail activity	128

Chapter 21. Reporting and business intelligence 129

Analytical reports and business intelligence	129
Closed-loops analytics	129
WebSphere Commerce operational reports	130
Expected inventory management	130
Inventory reports	130
Marketing reports	131
Product management reports	131
Return management reports	131
Shipping exceptions	132

Appendix A. Use cases 133

Adding a new address	134
Adding an item to a requisition list	135
Adding an item to the wish list	136
Adding items to an order	137
Changing personal information	138
Checking out the shopping cart	139
Creating a quick checkout profile	141
Displaying cross-sells in a category page	142
Displaying the bundle page	143
Displaying the package page	144
Displaying the product page	145
Displaying the shopping cart	146
Editing an address book	147
Logging on by a Registered Customer	148
Logon	149
Managing an account	150
Placing a quick order	151
Registering	152
Searching for a specific product	153
Setting up an e-mail address to receive e-mail	
option	154
Setting up cross-sells, up-sells, and accessories	155
Submitting your order	156
Updating personal information	158
Updating receive promotional e-mail option	159
Using a private requisition list	160
Using quick checkout	162
Using the Home page	164
Using the shopping total	165
Viewing a product category	166
Viewing a wish list	167
Viewing orders	169

Appendix B. Data beans, access beans and commands used in the Express Store	171
Data beans	171
Access beans	174
Commands	175
Appendix C. Publishing a store to a production environment.	179

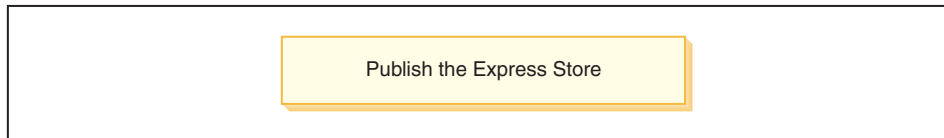
Publishing the Express Store to a WebSphere Commerce Server	179
+ Appendix D. Deploying the Express Store	185
Notices	187
Trademarks	188

Part 1. Up and running with the Express Store

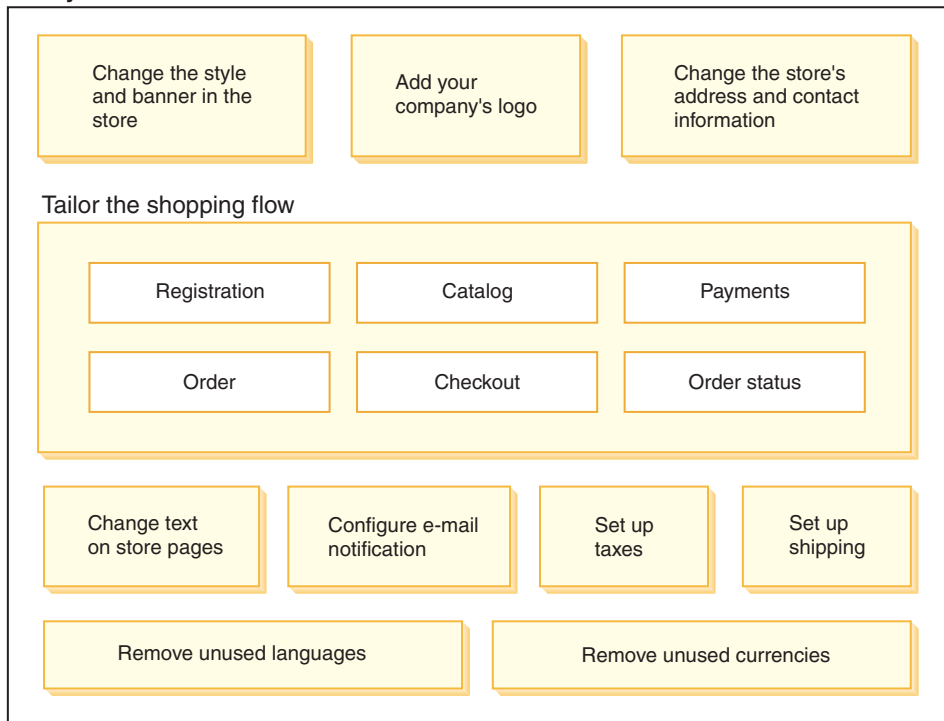
WebSphere Commerce — Express Version 5.5 is a comprehensive set of integrated software components that helps you build, maintain, and manage stores on the Web. This part of the book provides step-by-step instructions to publish and configure the Express Store. Once the store has been set up you may want to customize the store. Customizing WebSphere Commerce is described in Part 3, “Adding additional features to the Express Store,” on page 55.

The following diagram shows the steps required to set up your store:

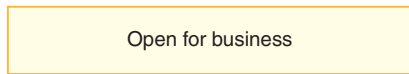
Create a store



Add your information



Complete the store



Depending on how the Express Store will be deployed, you may need to perform the steps shown in the diagram on one or more machines. For information about deploying the Express Store refer to the section Appendix D, "Deploying the Express Store," on page 185.

You will use several tools to publish and set up your store, all these tools can be accessed from a browser. When you use these tools; the changes that you make will be immediately reflected on your site.

Administration Console

The Administration Console allows you to control your site or store by completing administrative operations and configuration tasks. If you are a Site Administrator, you select the store and language with which you want to work when you log on to the Administration Console. The tasks that you are authorized to perform display on the Administration Console home page through various menus. These tasks are based on the user group names (roles) and authority levels.

WebSphere Commerce Accelerator

The WebSphere Commerce Accelerator allows you to maintain online stores and catalogs by completing various store operations, from managing the look and feel of your store to creating and maintaining orders to tracking store activities.

If you are authorized to work with multiple stores, when you log on to the WebSphere Commerce Accelerator, you select the store and language with which you want to work. If you are authorized to work with a single store, the store name is pre-selected during logon. Additionally, if the store supports more than one language, you can select the language with which you want to work.

Organization Administration Console

The Organization Administration Console allows you to set up users at your site. This tool is used both to set up administrators, and in the case of a B2B direct site to register customers to the site.

Chapter 1. Creating your store

This chapter describes how to select a business model; and how to publish the Express Store SAR file to a WebSphere Commerce Server.

Publishing refers to transferring files to a location on the WebSphere Commerce Server, and loading data assets to the WebSphere Commerce database. You use the Administration Console to publish the store archive to a WebSphere Commerce Server, thus creating an operational store.

Determining the business model

The Express Store is available in two variations:

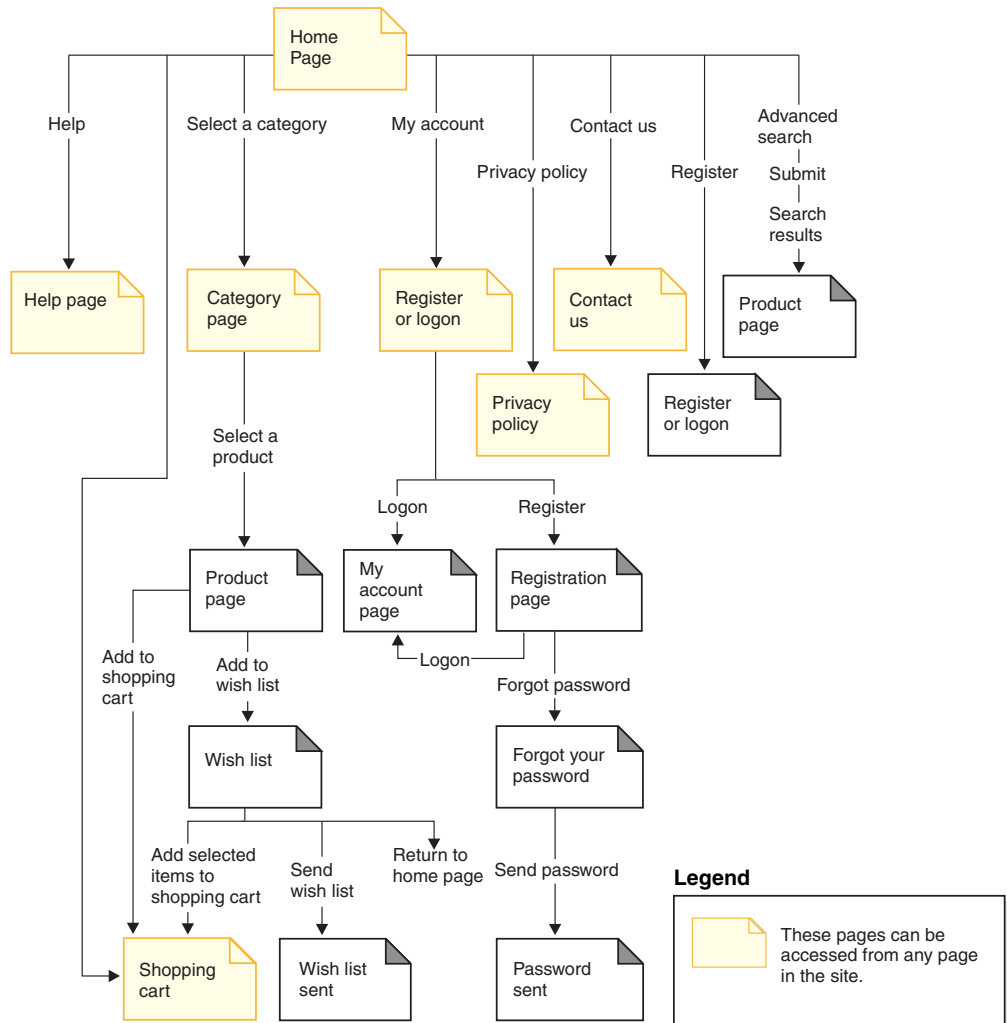
Consumer direct

The consumer direct business model supports commerce transactions that involve products, services, or information between businesses and consumers. Consumers typically purchase goods or services directly from a business in a consumer direct scenario. The Express Store sample store is an example of a consumer direct business. The shopping flow for the consumer direct Express Store is shown in the section “Shopping flow in the consumer direct Express Store” on page 4.

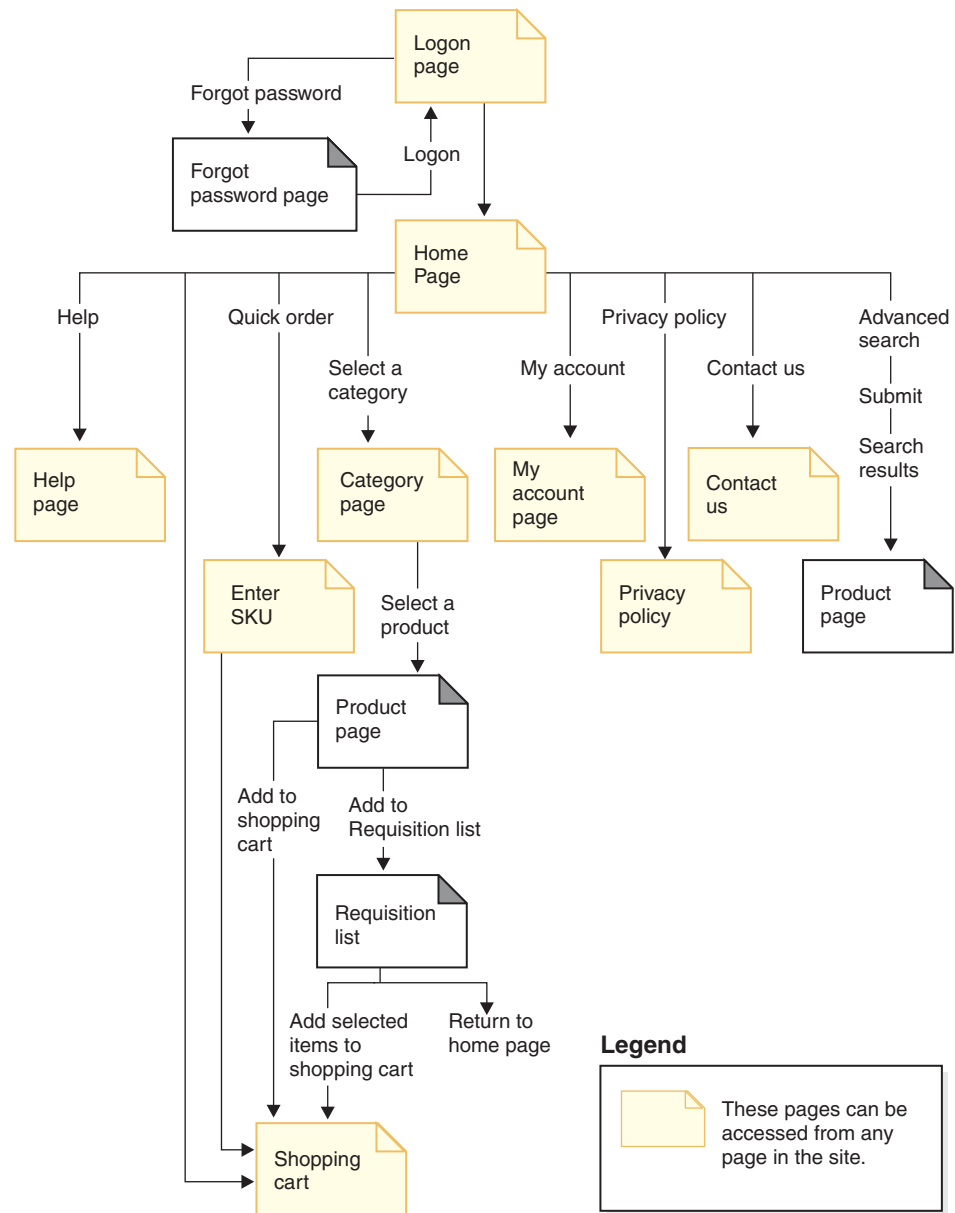
B2B direct

The B2B direct business model supports commerce transactions that involve products, services, or information between two businesses or parties. Typical B2B direct transactions occur between buyers, suppliers, manufacturers, resellers, distributors, and trading partners. If you use the B2B access control policies, the Express Store sample store becomes an example of a B2B direct business. The shopping flow for the B2B direct Express Store is shown in the section “Shopping flow in the B2B direct Express Store” on page 5.

Shopping flow in the consumer direct Express Store



Shopping flow in the B2B direct Express Store



Using the B2B access control policies

Important

A B2B store and a consumer direct store can not coexist in the same instance. The B2B direct access control policies restrict some access, while the consumer direct access control policies grant all access. For example, if you publish a B2B store, then publish a consumer direct store, the B2B access control policies will be overwritten by the consumer direct access control policies. The result will be two consumer direct stores.

To create a B2B direct variant of the Express Store replace the consumer direct access control policy file in the Express Store SAR file with a B2B direct access control policy file.

1. Navigate to the directory:

▶ 400 ▶ Linux ▶ Windows

WC_installdir/samplestores/ExpressStore

▶ Developer

WCDE_installdir/samplestores\ExpressStore

2. Extract the *ExpressStore.sar* to a temporary directory, for example *c:\temp*.
3. Rename the file *ExpressStore.sar* to *ExpressStore.sar.bak*.
4. Replace the file *accesscontrol.xml* in the directory:

temporary_directory/WEB-INF/stores/StoreAssetsDir/data/accesscontrol.xml

with the file *accesscontrol.xml* in the directory:

WC_installdir/samplestores/ExpressStore/B2BAccessControl/accesscontrol.xml

5. Compress the *temporary_directory*, preserving the subdirectory structure, to *ExpressStore.sar*.

You may use any up-to-date compression utility, such as Winzip or PKZIP.

6. Copy the file *ExpressStore.sar* to the directory:

▶ 400 ▶ Linux ▶ Windows

WC_installdir/samplestores/ExpressStore

▶ Developer

WCDE_installdir\samples\stores\ExpressStore

You have now replaced the consumer direct access control policy file with the B2B direct access control policy file. Continue with the steps in the next section. After you publish the store, ensure that you Restrict the site to Registered Customers, for more information see the section on page 13.

Prerequisite information

Ensure that WebSphere Commerce has been correctly installed and configured. If you are publishing the Express Store to a development environment, ensure that WebSphere Commerce - Express development environment has been correctly installed and configured.




Publishing the Express Store

The instructions in this section describe how to publish the Express Store to the WebSphere Commerce - Express development environment or to a WebSphere Commerce Server running on Windows, iSeries™, or Linux on Intel™.


Important

If you are publishing a store to a production environment, as opposed to a demonstration, development or learning environment, follow the steps in Appendix C, "Publishing a store to a production environment," on page 179 instead of the following directions.

Follow the directions applicable to your platform:

-  iSeries
-  Linux on Intel
-  Windows
-  WebSphere Commerce - Express development environment

Steps that are common to all platforms and environments are not marked with icons.

1. Ensure that you have selected a business model, either B2B direct or consumer direct. If the store will use the B2B direct model, ensure that the access control policies have been replaced in the SAR file as described in the section "Using the B2B access control policies" on page 5. If the store will use the consumer direct model, proceed with the steps in this section.
2.  Ensure that you have started WebSphere Commerce within the development environment.
 - Select **Start > Programs > IBM WebSphere Commerce Studio > WebSphere Commerce development environment**.
 - In the **Servers** view, right-click **WebSphereCommerceServer** and select **Start** from the pop-up menu.

The server started successfully if there are no errors in the Console view and the following message appears:

```
Server server1 open for e-business
```
3. If you will use WebSphere Commerce Payments, and you would like to set up merchant information, accounts and brands for the Offline Card; start the Payment Server. Follow the directions in the section 2 on page 84.
4. Open the Administration Console:



`https://host_name/webapp/wcs/admin/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogon`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

`https://host_name:8002/adminconsole`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server.

5. From the Logon page, type your WebSphere Commerce user name and password and click **Log On**; then select the **site** radio button and click **OK**.
6. From the Store Archives menu, select **Publish**. The Store Archives page displays.

7. Select the ExpressStore; then click **Next**. The Parameters page displays.
8. If this is the first time that you are publishing the store, accept the default values; otherwise choose new values. The store identifier and store name must be unique. Do not publish a store using the same store directory or store identifier as a previously published store.

store directory

ExpressStore

store identifier

ExpressStore

You will need to know these values to load the sample catalog.

Important



The values that you use for the store directory and store identifier must be unique. If you publish the ExpressStore multiple times, make sure that you always use a different value for the store directory and the store identifier. Using the same value more than once will cause incorrect data to be displayed in the store.

9. Review the summary information; then click **Finish**. A confirmation dialog displays, telling you the number of the publishing job. Note the job number for future reference. Click **Ok**. The Publish Job Status page displays. Click **Refresh** to update the status.

The text Successful in the Publish Status column indicates that the publish job completed successfully.

10. Review the publishing details.
11. If the store archive published successfully, you can launch the store. To view and test your store, Click **Launch Store**. A dialog box displays, asking you for the Web Application Web path. Accept the default. Click **Ok**.

The store you have published contains all the pages and assets necessary for a functioning store, except for a catalog. You will need to add a catalog to the store before you can complete a shopping flow.

12.    Compile WebSphere Commerce JSP pages. This will significantly reduce the time needed to load each store page.

a. Start a command line session.

b. Change to the following directory:

```
WAS_installdir/bin
```

c. Run the JspBatchCompiler command, note that the command is shown on multiple lines for display purposes only. Enter the command on one line.

```
JspBatchCompiler.bat -enterpriseapp.name "WC_instance_name"
-cell.name "hostname" -node.name "hostname"
-server.name "WC_instance_name"
```

where:

instance_name

The name of the WebSphere Commerce instance for which you are compiling the JSP files.

hostname

The short host name of the machine on which you are compiling the JSP files.

Compiling the pages will take some time. Several errors may be logged when you perform these compiles. These errors may be safely ignored.

13. Enable caching.

a. Copy the file

```
WC_installdir/samples/dynacache/ExpressStore/cachespec.xml
```

to the directory:

```
WAS_installdir/installedApps/host_name/WC_instanceName.ear/  
Stores.war/WEB-INF
```

This file contains the caching rules for shopping events at the Express Store, for example, the file specifies that a requisition list is deleted from the cache when a customer creates a new requisition list.

b. Open the file

```
WC_installdir/samples/dynacache/invalidation/catalog/cachespec.xml
```

Copy the invalidation policies from this file to the cachespec.xml file for your store. You can place the invalidation policies at the end of your store's cachespec.xml file before the last element. This file describes the non-shopping events that invalidate cached data, for example, when the product description is updated in the WebSphere Commerce Accelerator the cached pages containing that product are invalidated.

c. If necessary, change the name and directory of JSP files in the sections copied from the sample invalidation files to match your store's information. You will need to do this if the name of the store is not ExpressStore, or if you did not publish the store to the ExpressStore directory.

d. Save and close the file,

```
WAS_installdir/installedApps/host_name/WC_instanceName.ear/Stores.war/WEB-INF
```

e. Restart the WebSphere Commerce Server. Directions for restarting the WebSphere Commerce Server are described in the *WebSphere Commerce Installation Guide* for your platform and edition.

Chapter 2. Configuring the Express Store

This chapter describes how to enable or disable optional store features by using the Change flow feature in the WebSphere Commerce Accelerator. When you publish the Express Store many features are enabled, for example there is a field on the store's Registration page that asks customers to provide their gender. You may not want this field to be displayed in your store; by using the Change flow page, you can easily remove this field. The aspects of your store that you can configure by using Change flow are as follows:

- Registration
- Catalog
- Orders
- Checkout
- Payments
- Order status

To configure the Express Store, do the following:

- Review the diagrams showing the shopping flow in the Express Store in the sections "Shopping flow in the consumer direct Express Store" on page 4 and "Shopping flow in the B2B direct Express Store" on page 5.
- Read through the available configuration options in this chapter.
- Configure the options for your store by following the directions in the section "Changing store flows" on page 17.

Registration

Option	Description	When to use	Implication	Enabled by default
Preferred language	Allows customers to select the language in which the store displays.	The store is available in more than one language.	Ensure that product descriptions and names are entered for every language that the store supports.	Yes

Option	Description	When to use	Implication	Enabled by default
Preferred currency	Allows customers to select the currency in which product and item prices display.	Customers can purchase merchandise in more than one currency.	Ensure that currency exchange rates are set for all supported currencies, for more information see "Changing exchange rates" on page 80. Alternately, enter the product prices in all supported currencies.	Yes
Collect age information	Adds a question on age to the registration and change personal information pages.	The store uses marketing campaigns that target customers based on age.	The store creates campaigns.	Yes
Collect gender information	Adds a question on gender to the registration and change personal information pages.	The store uses marketing campaigns that target customers based on gender.	The store creates campaigns.	Yes
Enable customer e-mail	Offer the shopper a choice of whether to receive promotional e-mail.	The store will use promotional e-mail campaigns.	To send e-mail to customers, you must configure your e-mail accounts and activities.	Yes

Option	Description	When to use	Implication	Enabled by default
Restrict the store to registered customers	Customers will not be able to self-register at the store.	The store is a business-to-business store that is restricted to pre-approved, pre-registered customers.	If this option is selected, ensure that all customers who should be able to purchase from the store are already Registered Customers. Customer information must exist in WebSphere Commerce database. For information about using the Loader to mass load Registered Customer information see Chapter 9, "Loading existing customer information," on page 51. Alternately, enter customer information by using the Customer Profile notebook in the WebSphere Commerce Accelerator.	No

Catalog

Option	Description	When to use	Implication	Enabled by default
Include search in the store	Provides customers with the option to search the store catalog.	The store catalog is large enough that searching the catalog based on keywords would be helpful.		Yes

Option	Description	When to use	Implication	Enabled by default
Include quantity selection	Adds a quantity selection box to the product, prebuilt kit, or bundle page.	To enable customers to select the quantity when adding these items to the shopping cart.	If this field is disabled, customers can change the quantity on the Checkout page.	Yes
Include category selection list	Adds a list to the header that displays all categories and subcategories in the store.	Enables customers to quickly go to any section of the store's catalog.		No
Display product only	Select this to display the product; otherwise all items under the product are also displayed.			Yes

Checkout

Option	Description	Implication	Enabled by default
Quick checkout	Allows registered customers to complete an order by using shipping and billing information that has already been defined in the quick checkout profile.	In order to use a quick orders, customers must first create a quick checkout profile, which includes entering billing and shipping addresses, shipping method, and payment information.	Yes
Single or multiple shipping addresses per order	Allows customers to ship all items in an order to a single address, or to ship items within an order to multiple shipping addresses.	If this option is disabled, customers can only ship an order to a single address.	Yes

Option	Description	Implication	Enabled by default
Single or multiple shipping methods per order	Allows customers to ship all items in an order using a single shipping method, or to ship items within an order using multiple shipping methods. For example, a customer could ship one item using Express mail and all other items using Regular mail.	If this option is disabled, customers can only ship an order using a single shipping method.	Yes

Orders

Option	Description	When to use	Enabled by default
Shopping total	Allows customers to see the current value of their order, and the number of items in the shopping cart, without going to the Shopping Cart page. The shopping total is displayed in the sidebar, therefore, it can be seen on all store pages.		Yes
Quick order	Allow customers to enter a SKU to order an item quickly. The item is added to the shopping cart	Use quick order when customers are likely to know the SKU numbers. For example, if the store sends out paper catalogs the customer could simply enter the SKU number, as printed in the catalog, on the Quick Order form.	No
Include wish list	Allows customers to maintain a list of items that they would like to order in the future. Customers can also send their wish list to family and friends using e-mail.	Wish lists are usually used in a Consumer direct store.	Yes

Option	Description	When to use	Enabled by default
Include requisition list	Allows registered customers to create a reusable list of items to create orders at a later date. Customers can create requisition lists of items that they reorder frequently and use the requisition list to periodically re-order the items.	Use requisition lists in a B2B direct store where customers frequently reorder the same items.	No
Disable wish list and private requisition list	Disables both requisition lists and wish lists.	The store will not use either requisition lists or wish lists.	No

Payments

Option	Description	When to use	Enabled by default
Enable WebSphere Commerce Payments	WebSphere Commerce Payments handles all the background details of Internet payments for you. It provides a graphical interface to simplify payment transaction management.	If you select this option, you must install and configure WebSphere Commerce Payments. For more information see Chapter 14, "Managing your payments," on page 83.	No

Order status

Option	Description	When to use	Implication	Enabled by default
Track Order Status	Allows customers to track the status of their order. For example, customers can see if an order is still waiting approval, is already processed, or is scheduled to be processed.	If the store will update the order status during payment authorization, capture and fulfillment. If you use WebSphere Commerce Payments this field will show the current status.	If the store does not update the order status, this field should be disabled since the value that it displays may be misleading to customers.	No

Option	Description	When to use	Implication	Enabled by default
Show shipment tracking URL	A shipment tracking URL allows customers to track the shipment status of their order by linking to the appropriate shipping provider's Web site.	If the store enters tracking URLs for orders shipped to customers.	Enter tracking URLs when shipping providers and methods are set up. And, customize WebSphere Commerce so that shipping information is added to every order item, for more information see "Shipment tracking URL" on page 79.	No

Changing store flows

- (Optional) Backup the store's JSP files. The JSP files are located in the following directory:

▶ 400 ▶ Linux ▶ Windows

`WAS_installdir/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/storedir`

▶ Developer

`WCDE_workspacedir/Stores/Web Content/storedir`

`storedir` is the Store Directory specified when you published the store. The default value is `ExpressStore`.

- Open the WebSphere Commerce Accelerator.

▶ Developer

`https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=common.mcLogon`

where `host_name` is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

▶ 400 ▶ Linux ▶ Windows

`https://host_name:8000/accelerator`

where `host_name` is the fully qualified TCP/IP name of your WebSphere Commerce Server machine.

- From the Store menu, select **Change Flow**.
- From the left navigation frame, select the feature you want to configure: **Registration, Catalog, Orders, Checkout, Payments, Order Status**.
- Select the appropriate check boxes or radio buttons.
- Select **Apply** or **Apply Permanently** to apply the changes. If you select **Apply Permanently**, you will not be able to reverse your choice later; however, you can restore from the files that you backed up in step 1.

7. Click **Launch Store** to view the changes in your store. If you do not see the changes, refresh your browser.

Chapter 3. Replacing the Express Store default information with your store's information

When you publish a store it includes the same default text and logo that all other Express Stores receive at creation. As a result, you may want to customize your store to make it reflect your business and your business practices.

The WebSphere Commerce Accelerator offers several tools that allow you to customize your store. You can use these tools to do the following:

- Upload the store's logo
- Change the store name and description
- Change the store contact information
- Change the store location information
- Change the language support for your store
- Change the currency support for your store
- Change the look and feel of your store, including layout, color scheme and banner
- Change text on store pages and notification messages

Uploading the store's logo

To upload the store's logo, do the following:

1. Open the WebSphere Commerce Accelerator.

▶ Developer

`https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=common.mcLogon`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

▶ 400 ▶ Linux ▶ Windows

`https://host_name:8000/accelerator`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server machine.

2. From the **Store** menu, select **Upload Logo**. The Logo page opens. If you do not see this menu, then your logon ID does not have the appropriate authority to perform this task. Contact your Site Administrator.
3. In the **Logo** file field, type the file path for the logo, or click **Browse** and navigate to the file on your system. The file must be in .jpg or .gif format to be uploaded correctly.
4. Click **Upload and Apply**.
5. To see the new logo in your store, click **View Store**. Your store displays.
6. Click **Close** to save your settings and return to the WebSphere Commerce Accelerator home page.

Changing store information

To change the store name, store description, store contact information, store location information, language, and currency, do the following:

1. Open the WebSphere Commerce Accelerator.

 Developer

`https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=common.mcLogon`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

 400 Linux Windows

`https://host_name:8000/accelerator`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server machine.

2. From the Store menu, select **Change Profile**. The Store Profile notebook opens.
3. Change information as appropriate:
 - To change the store name or store description select **General**.
 - To change the store contact information select **Contact**.
This is the contact information that is displayed when a customer clicks **CONTACT US** from any store page.
 - To change the location of store select **Location**.
 - To change the languages available in the store select **Languages**.
The languages that you choose here display on the **SELECT A LANGUAGE** list on the navigation frame of all store pages. Note that if only one language is available the **SELECT A LANGUAGE** list does not appear.
 - To change the currencies available in the store select **Currency**.
The currencies that you choose here display on the **SELECT A CURRENCY** list on the navigation frame of all store pages. Note that if only one currency is available the **SELECT A CURRENCY** list does not appear.
4. Click **OK** to save your changes and exit the Store Profile page.
5. To see the changes in the store, launch the store's home page.

Changing store style

Using the Change Style wizard in the WebSphere Commerce Accelerator, you can change the style of your store, including layout, color, and banner. To change the style of your store, do the following:

1. Open the WebSphere Commerce Accelerator.

 Developer

`https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=common.mcLogon`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

 400 Linux Windows

`https://host_name:8000/accelerator`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server machine.

2. From the **Store** menu, select **Change Style**. The Style page displays. If you do not see this menu, then your logon ID does not have the appropriate authority to perform this task. Contact your Site Administrator.
3. Select a layout and click **Apply**.
4. To view the new layout in your store, click **View Store**. Your store opens displaying the new layout.
5. When you have decided upon a layout, click **Next**. The Color page displays.
6. Select a color scheme and click **Apply**.
7. To view the new color scheme in your store, click **View Store**. Your store opens displaying the new colors.
8. Click **Next**. The Banner page displays. The Banner page allows you to select a new banner from several available banners, or to upload a banner of your own.
9. To select a banner, do one of the following:
 - a. Upload your own banner:
 - 1) Select **Use your own banner**.
 - 2) In the **Banner image** field, type the file path for the banner, or click **Browse** and navigate to the file on your system.
 - 3) Click **Upload**.
 - b. Select **Select a banner**. The available banners display.
10. Click **Apply**. To view the new banner in your store, click **View Store**. Your store opens displaying the new banner. To view the other available banners in your store, return to the Banner page, select another banner and repeat.
11. Click **Finish**.
12. To see the changes in the store, launch the store's home page.

Changing text on store pages and notification messages

In the Express Store you can change all text displayed on store pages. Most text can be changed by using the WebSphere Commerce Accelerator Change Page feature; however, some text can only be changed by modifying properties files.

The method that you use to change text on the store pages depends on the text.

Text on store pages, such as the privacy policy, location, and text in notification messages

Use the Change Page dialog in the WebSphere Commerce Accelerator. The process for changing pages using this feature is described in this section.

All other text

This text includes: Error messages, Labels for fields and the text in the header, footer or sidebar. To change this text modify properties file as described in the section, "Modifying property files" on page 64.

1. Open the WebSphere Commerce Accelerator.

 Developer

`https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=common.mcLogon`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

 400 Linux Windows

`https://host_name:8000/accelerator`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server machine.

2. From the Store menu, select **Change Pages**. The Change Pages page opens.
3. To change the text, edit or replace the existing text in the corresponding scroll box. The text displayed in the scroll box may contain HTML tagging, and placeholders for variables that are retrieved from the database.

HTML tagging

An example of HTML tagging is `Personal information`, which means Personal Information will display in bold when rendered by a browser. If you want to change how the font displays in the page, you will need an understanding of HTML.

Database placeholders

A digit displayed inside curly brackets is a database variable such as {0}. Database variables hold information such as the name of the store. If you want to maintain these placeholders, do not change them.

Links To link to other pages, add the prefix `/wcsstore` to the file path.

```

```

where *store_directory* is the name of the store directory you specified when creating your store. `wccstore` is an alias set up in the http server; you could also use `webapp/wcs/stores/servlet`.

4. Click **OK** to save your settings and return to the WebSphere Commerce Accelerator home page.
5. To see the changes in the store, launch the store's home page.

Chapter 4. Loading the Express Store sample catalog

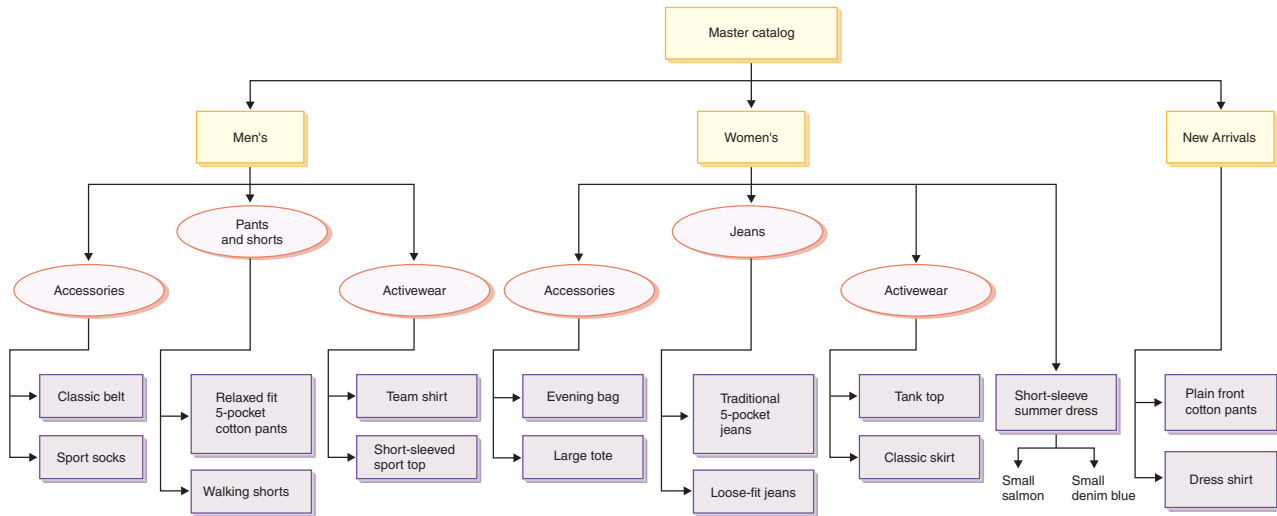
This chapter describes how to load the sample catalogs provided with the Express Store. There are two sample catalogs: a catalog featuring clothing, and a catalog featuring hardware. It is necessary to add a catalog to the store before you can complete a shopping flow. Using the sample catalogs allows you to learn how catalogs are implemented in WebSphere Commerce. You will then be ready to create your own store's catalog data. You can either load a sample catalog, or you can create a catalog by using the Product Management tools in the WebSphere Commerce Accelerator, for more information see Chapter 16, "Managing the catalog," on page 97.

Catalog structure

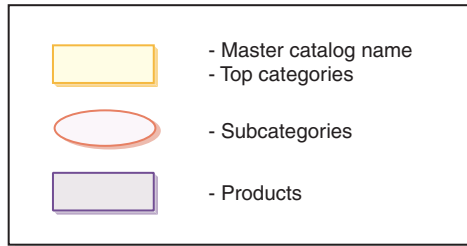
Like a traditional catalog, an online catalog consists of the goods and services that you offer for sale. Although the size and structure of online catalogs vary greatly from store to store, catalogs require the following:

- What you are selling, including:
 - Prices, which are almost always included in an online catalog.
 - Product data, such as descriptions and images of your merchandise.
 - Categories, as most, but not all catalogs divide merchandise into categories, to facilitate navigation for customers.
- A display method for what you are selling. Catalog display pages outline how a page looks to your customers and provide a consistent look and feel between various catalog pages. The structure of the catalog depends upon the merchandise.

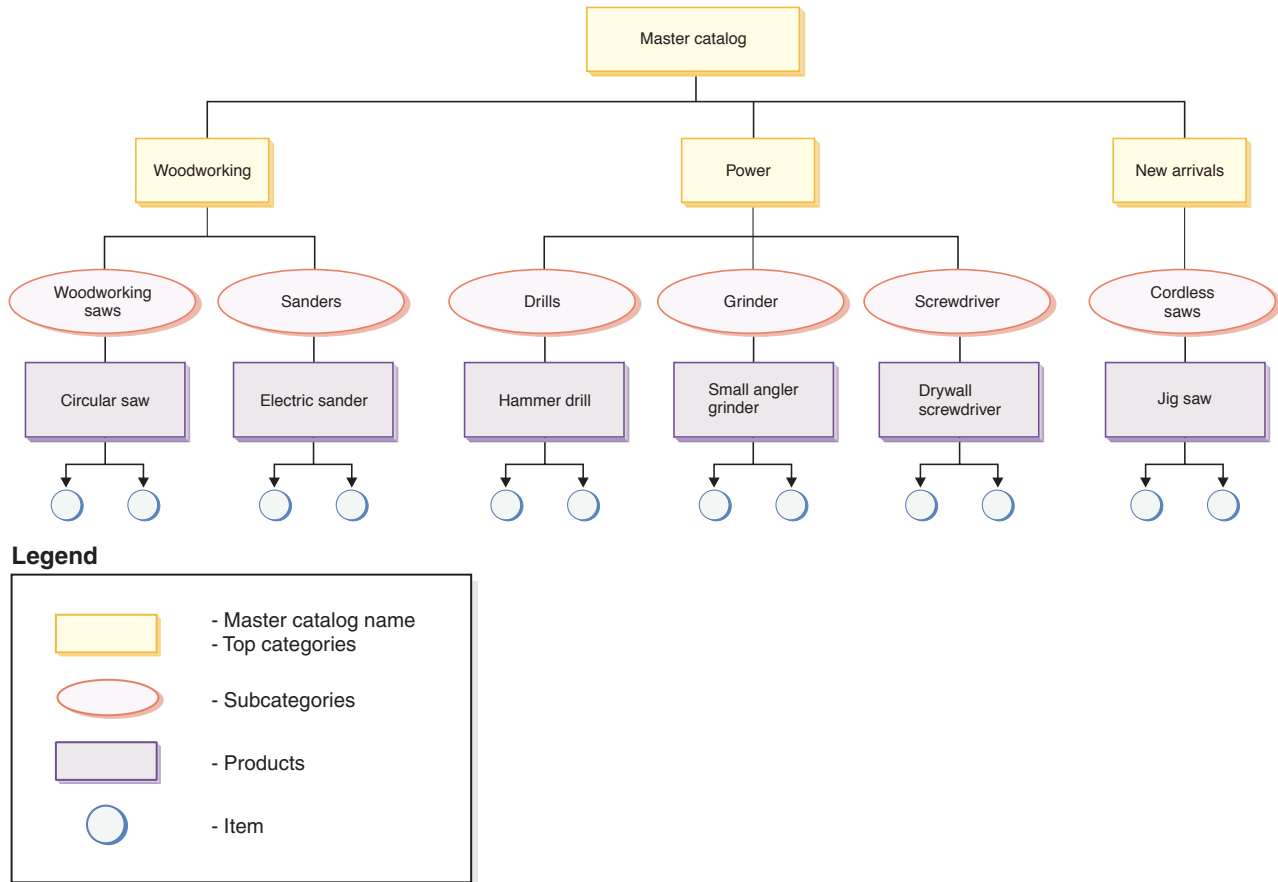
Clothing store sample catalog



Legend



Hardware store sample catalog



Catalog

The *master catalog* is the starting point. The catalog contains all hierarchical and navigational information. It is a collection of catalog groups and catalog entries that are displayed and available for purchase in an online store.

In WebSphere Commerce, a *catalog entity* represents the catalog in the database. A catalog entity consists of a unique catalog ID and a description of the catalog, for example, the catalog name. Every store in the WebSphere Commerce system must be related to at least one catalog entity.

Catalog groups

Catalog groups are generic groupings of your catalog entries, created for partitioning purposes. A catalog group belongs to a catalog and may contain more than one catalog group or catalog entries. You can associate catalog groups to more than one catalog. A catalog group is also known as a *category*.

When creating catalog groups, you must first arrange your catalog in a hierarchy, or inverted tree. The tree begins at general catalog groups (called top categories), and branches out into increasingly specific subcategories until it cannot be further divided. A catalog group is the parent to the categories immediately below it, and a child of the one above. As an example in the figure “Clothing store sample catalog” on page 24, Men’s is a grouping of the men’s apparel categories, while the catalog groups Activewear and Accessories are groupings of catalog entries.

Catalog entries

Each catalog group contains *catalog entries*. Catalog entries represent orderable merchandise in an online catalog. The entries typically have a name or part number, a description, one or more prices, images, and other details. In the sample catalogs a catalog entry can be a product, item, bundle, or prebuilt kit. More information on each type of catalog entry is available below.

Products

A *product* is a type of catalog entry. A product acts as a template for a group of items (or SKUs) that exhibit the same attributes. For example, a shirt is a product in your catalog. After you have defined attributes and attribute values for the shirt, each variation becomes an item, such as a small black shirt.

Items An *item* is a tangible unit of merchandise that has a specific name, part number, and price. For example, a small black shirt is an item while a shirt is a product. All items related to a particular product exhibit the same set of attributes. The items are distinguished by their attribute values.

Bundles

A collection of catalog entries which allow customers to buy several items with one click. For example, when a bundle for a computer is a central processing unit, a monitor, a hard drive, and a CD-ROM drive, the components can be sold separately. A bundle's price is composed of its different catalog entries. A bundle cannot be purchased directly.

Prebuilt kit

A prebuilt kit is a collection of catalog entries that cannot be sold separately. For example, a computer prebuilt kit might contain a specific central processing unit, monitor, and hard drive. A prebuilt kit has its own price and can be added to the shopping cart. Once added to the shopping cart, you cannot change the prebuilt kit contents. Viewed in similar ways to a product, a prebuilt kit can have descriptive attributes.

Attributes

Attributes are properties of products in an online store. There are two types of attributes:

- Defining attributes are properties, such as color or size. Attribute values are the property of an attribute such as a specific color (blue or yellow) or size (medium). You must predefine attribute values before assigning them to items. Attribute values are implicitly related to their attributes. Each possible combination of attributes and attribute values equals a new item. After creating attributes and their values, you can update information such as name, description, and type (text, whole numbers, or decimal numbers). The defining attributes are used for SKU resolution, where each possible combination of attributes and attribute values defines an item.
- In contrast, descriptive attributes simply provide additional descriptions. For example, some pieces of clothing should only be dry cleaned; a descriptive attribute can specify this dry-clean-only condition. Descriptive attributes are not used for SKU resolution. They enhance product descriptions and provide a method to add specific information.

Attribute values

Attribute values are properties of an attribute such as a specific color (blue or yellow) or size (small, medium, or large). You must predefine attribute values before assigning them to items. Each possible combination of defining attributes values defines an item.

Pricing

All catalog entries in the Express Store are assigned a price. Catalog entries in the Express Store also are assigned a list price. If the list price is lower than the offer price, then the list price is displayed on the store pages with a line through it. If you use list prices in your store, ensure that the list price for the product is the same as the list price for the SKU, or item. Some pages in the Express Store display the list price for the product, while other pages display the list price for the SKU.

For a globalized store, there are two methods in which to price products:

1. Assign a price to the product in every supported currency. You can create, list, and change pricing associated with a product or SKU by using the Product Management tools in the WebSphere Commerce Accelerator.
2. Assign a price to a product in the default currency; then set exchange rates for all other supported currencies. The Express Store demonstrates this method. When the Express Store was published exchange rates were loaded, to change the exchange rates see “Changing exchange rates” on page 80.

Marketing and merchandising features in the sample catalogs

The Express Store sample catalogs also demonstrate several merchandising features.

Prebuilt kit

The clothing sample catalog contains one example of an prebuilt kit. The Women’s top category page shows a Tank top and skirt set prebuilt kit.

The hardware sample catalog contains one example of an prebuilt kit. The Power Drills subcategory page shows a Hammer drill and small angle grinder prebuilt kit.

Bundle

The clothing sample catalog contains a Casual pants and belt bundle on both the Men’s and New arrivals top category pages. The hardware sample catalog shows a circular saw with electric sander bundle on the Woodworking category page.

Cross-sell

Cross-sell, where a merchant suggests other catalog entries based on an already chosen catalog entry. For example, a merchant can suggest a high-resolution color printer when a digital camera is purchased, or a particular type of batteries when a flashlight is purchased. The clothing sample catalog contains four examples of cross-sells as follows:

- Classic belt with plain front cotton pants on the top-category Men’s page
- Relaxed fit 5 pocket cotton pants with classic belt on the top-category Men’s page
- Tank top with classic skirt on the Women’s top-category page
- Classic skirt with tank top on the Women’s top-category page

The hardware sample catalog contains an example of a cross-sell on the Power Drills category page a cross-sell of a hammer drill with a circular saw is offered.

Up-sell

Up-sell, where a merchant suggests a better catalog entry based on the one selected. For example, a merchant can suggest a higher quality television when a low-end model is selected. . The clothing sample catalog shows an

up-sell on the New Arrivals page. The up-sell is Plain front cotton pants with relaxed fit 5 pocket cotton pants.

The hardware sample catalog contains one example of an up-sell. The Power Grinder category page up-sells a Small angle grinder with drywall screwdriver

Accessory

Accessory, where a merchant suggests an accessory that will complement the selected catalog entry. For example, a merchant can suggest an extra game controller when a video game system is selected. The clothing sample catalog contains one example of an accessory: Men's category page Walking shorts with sport socks is offered as an accessory.

The hardware sample catalog contains one example of an accessory. The Woodworking saws category page offers an electric sander with the purchase of a circular saw.

e-Marketing Spot

The e-Marketing Spots are used during page creation to reserve space for the dynamic content that will be shown to your customers. When a page is requested by a customer, any e-Marketing Spots present on the page will communicate with the rule server to process the rule-based code associated with the spot. Each e-Marketing Spot has one or more campaign initiatives associated with it. e-Marketing Spots are supported by special data beans that are used in conjunction with JSP technology to generate Web page content using the WebSphere Commerce rule processing system. The clothing sample catalog shows one example of an e-Marketing Spot a Tank top and Team shirt e-Marketing Spot is shown on the shopping cart page.

The hardware sample catalog shows one example of an e-Marketing Spot. The Shopping cart page shows a Hammer drill and circular saw e-Marketing Spot.

The Loading process

To load the sample catalog, use the Loader package. The sample catalog is comprised of several XML files, in which each element represents a particular database table, and the element attributes represent database columns. The following XML extract shows an element that will populate several columns in the CATALOG table.

```
<catalog
catalog_id="@catalog_id_1"
member_id("&MEMBER_ID;"
identifier("&STORE_IDENTIFIER"
description("&STORE_NAME; Catalog"
tpclevel="0"/>
```

Use the ID Resolver to generate identifiers for XML elements that require them prior to loading data into the database. An alias is substituted in place of the primary key (identifier) in the XML document. This alias is then used elsewhere in the XML file to refer to that element. This eliminates the need to know the unique indexes necessary to build the XML file. An alias begins with an @ and ends with a semi-colon, for example @catalog_id_1;. The ID Resolver replaces aliases with a unique value.

After running the ID Resolver, the preceding example would look like this:

```
<catalog
catalog_id="10001"
member_id="-2000"
identifer="ExpressStore"
description="ExpressStore Catalog"
tpclevel="0"/>
```

Where 10001 is the unique ID assigned by the ID Resolver and -2000 is the member ID. The resulting XML file then gets loaded using the Loader package.

Loading a sample catalog

Note: The Express Store contains two sample catalogs.

Fashion catalog

The fashion catalog is in the directory:
 \Commerce\samples\stores\ExpressStore\B2CCatalog\

Hardware catalog

The hardware catalog is in the directory:
 \Commerce\samples\stores\ExpressStore\B2BCatalog\

Loading the sample catalog in a development environment

 The directions in this section apply to WebSphere Commerce - Express Developer Edition.

1. If the store identifier is not ExpressStore, edit the `foreignkeys.dtd` file.

a. Open the appropriate file in a text editor:

```
Fashion catalog
WC_installdir/samples/stores/ExpressStore/
B2CCatalog/data/foreignkeys.dtd
```

```
Hardware catalog
WC_installdir/samples/stores/ExpressStore/
B2BCatalog/data/foreignkeys.dtd
```

b. Change the value of the STORE_IDENTIFIER entity:

```
<!Entity STORE_IDENTIFIER "ExpressStore">
```

where: STORE_IDENTIFIER is the **store identifier** that you entered in step 8 on page 8 publishing the store.

2. Save and close the file.

3. Set environment variables:

a. Open the file `WC_installdir/bin/setenv.bat` in a text editor.

b. Modify the values shown in *italics* with appropriate values for your system.

```
SET WAS_HOME=WCDE_installdir\runtimes\base_v5\
SET JAVA_HOME=WCDE_installdir\runtimes\base_v5\java
SET DB2_DRIVER=DB2_installdir\java\db2java.zip
SET WCS_HOME=WCDE_installdir
SET WCS_USER_HOME=WCDE_installdir
SET WCS_PATH=WCDE_installdir\bin;
WAS_installdir\bin;WAS_installdir\java\bin;
DB2_installdir\bin
SET WCS_JCE_CLASSPATH=WCDE_installdir\runtimes\base_v5\java\jre\lib\ext;
WCDE_installdir\lib\ibmjcefw.jar;
WCDE_installdir\lib\ibmjceprovider.jar;
WCDE_installdir\lib\local_policy.jar;
WCDE_installdir\lib\US_export_policy.jar;
SET WCLOGDIR=WCDE_installdir\logs
```

```

SET WCTEMPDIR=WCDE_installdir\temp
SET PM_HOME=WCDE_installdir\payments
SET PM_DB_CLASSPATH=DB2_installdir\bin;
DB2_installdir\java\db2java.zip
SET DB2_HOME=DB2_installdir

```

- c. Save and close the file.
4. From a command prompt window navigate to the directory:
WCDE_installdir\samples\stores\ExpressStore.
5. Type `load_sample_data`. You will be prompted for your database user ID, password, database name, schemaname, and the type of catalog that you want to load.

Loading the sample catalog to WebSphere Commerce running on Windows, Linux or iSeries

1. If the store identifier is not ExpressStore, edit the `foreignkeys.dtd` file.
 - a. Open the appropriate file in a text editor:

Fashion catalog

```

WC_installdir/samples/stores/ExpressStore/
B2CCatalog/data/foreignkeys.dtd

```

Hardware catalog

```



WC_installdir/samples/stores/ExpressStore/
B2BCatalog/data/foreignkeys.dtd

```

- b. Change the value of the STORE_IDENTIFIER entity:

```
<!Entity STORE_IDENTIFIER "ExpressStore">
```

where: STORE_IDENTIFIER is the **store identifier** that you entered in step 8 on page 8 publishing the store.

2. Save and close the file.
3.   Open a DB2 Command window.
4. Resolve IDs in the XML files.
 - a. Change to the directory: *WC_installdir*\bin
 - b. Run the ID Resolve command:

```

idresgen -dbname mall -dbuser dbuser -dbpwd dbpwd -infile "infile"
-outfile "outfile" -method mixed -schemaname schemaname
dbname

```

The name of the database.

`dbuser`

The name of the user connecting to the database.

`dbpwd` The password for **dbuser**.

`schemaname`

The name of the schema. The schemaname is usually the userid that WebSphere Commerce - Express was installed with.

`infile`

The name, including the full path, of the input file. The value of the `infile` parameter must be enclosed in quotation marks:

```

"WC_installdir/samples/stores/ExpressStore/B2CCatalog/store-catalog-assets.xml"
"WC_installdir/samples/stores/ExpressStore/B2BCatalog/store-catalog-assets.xml"

```

`outfile`

The name, including the full path, of the output file. The value of the `outfile` parameter must be enclosed in quotation marks:

```
"WC_installdir/samples/stores/ExpressStore/B2CCatalog/store-catalog-assets-out.xml"  
"WC_installdir/samples/stores/ExpressStore/B2BCatalog/store-catalog-assets-out.xml"
```

400

```
"WC_userdir/instances/temp/store-catalog-assets-out.xml"
```

5. Load the data.

a. Change to the directory: *WC_installdir/bin*

b. Run the Mass Load command:

```
massload -dbname mall -dbuser dbuser -dbpwd dbpwd -infile "infile"  
-method sqlimport -schemaname schemaname
```

dbname

The name of the database.

dbuser

The name of the user connecting to the database.

dbpwd The password for **dbuser**.

schemaname

The name of the schema. The schemaname is usually the userid that WebSphere Commerce - Express was installed with.

infile

The name, including the full path, of the input file. The value of the *infile* parameter must be enclosed in quotation marks:

```
"WC_installdir/samples/stores/ExpressStore/B2CCatalog/store-catalog-assets-out.xml"  
"WC_installdir/samples/stores/ExpressStore/B2BCatalog/store-catalog-assets-out.xml"
```

6. Copy the appropriate images for the sample catalog:

a. Copy product images for the sample catalog that you loaded:

Fashion catalog

```
WC_installdir/samples/stores/ExpressStore/B2CCatalog/images
```

Hardware catalog

```
WC_installdir/samples/stores/ExpressStore/B2BCatalog/images
```

b. Put the copied images in this directory:

```
WAS_installdir/installedApps/hostname/  
WC_instance_name.ear/Stores.war/Store_id/images
```

where:

hostname

The short host name of the machine.

instance_name

The name of the WebSphere Commerce instance. The default name is *demo*.

Store_id

The store identifier. The default identifier is *ExpressStore*.

7.  Change the file permissions for all files in the images directory to 555.

Placing a test order

Now that you have loaded the sample catalog, you can place a test order. To place a test order do the following:

1. If the store uses WebSphere Commerce Payments, ensure that the payment server is started.
2. If you are using any of the browser-based tools, for example WebSphere Commerce Accelerator, or the Administration Console logoff the tools.

3. Go to the store home page.
4. Add any product to the shopping cart.
5. Use the following payment information:
 - From the **Credit card type** drop-down list, select a credit card.
 - In the **Card** number field, type 4111111111111111.
 - Select any future expiration month and year.
 - Click **Order now**. The Order confirmation page displays, stating the details of your order.

Chapter 5. Shipping

This chapter describes how to use the WebSphere Commerce Accelerator to add shipping charges to orders. Using the WebSphere Commerce Accelerator you can create shipping charges based on the weight of the order, the number of items in the order, or use fixed rate charges. You can apply any combination of the three types of charges to an order. If necessary, an application developer can customize the shipping implementation. For information on customizing shipping refer to the WebSphere Commerce - Express development environment online help "Shipping" topic.

Shipping charges are based on the following factors:

jurisdiction

The jurisdiction to which you are shipping the order. Jurisdictions are geographical regions or zones representing a country or region, province or territory, or zip code range, to which you sell goods.

shipping mode

The shipping mode you are using to ship the order. A shipping mode is the combination of the shipping carrier (the company that provides shipping services) and the shipping service offered by that carrier. For example, XYZ Courier, Overnight service is a shipping mode.

shipping code

The shipping code used to calculate the shipping charge. A shipping code indicates which shipping charges are applied to which products in the order. Shipping codes are assigned to products or categories. When a customer purchases a product, the codes assigned to it determine the shipping charges that are added to the order.

shipping charge type

The type of shipping charge. Shipping charges can be applied based on the weight of the order, the number of items in the order, or on a fixed rate for the order. More than one shipping charge can be applied to a product or order.

Before you set up shipping charges using the Shipping notebook in the WebSphere Commerce Accelerator, you should have the following company-specific data available:

- Country or region in which the store is located, for example, Canada, or United States.
- The names of all countries or regions to which the store provides shipping, for example Canada, United States, and Germany.
- The names of all supported shipping providers.
- The names of all shipping services offered by each shipping provider, for example, Overnight, 3 day or 7 day.

Setting up shipping

Step 1. Defining jurisdictions

- a. Open the WebSphere Commerce Accelerator.

 Developer

`https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=common.mcLogon`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

 400  Linux  Windows

`https://host_name:8000/accelerator`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server machine.

- b. From the Store menu, select **Change Shipping**. The Shipping overview page displays.
- c. Click **Shipping Jurisdictions**. The Jurisdictions page displays.
- d. To define a jurisdiction, click **New**. The New Jurisdiction page displays.
- e. In the **Name** field, type a name for the jurisdiction.
- f. Define at least one of the following fields:

Country/Region

From the **Country/Region** drop-down list, select a country or region.

State/Province

- If the shipping rate applies to the entire country or region (or if the country or region does not have states or provinces) leave this field blank.
- If the shipping rate applies to a specific state or province, type the name of the state or province in the **State/Province** field.

City In the **City** field, type the name of the city.

ZIP code range

- In the **Start** field, type the first ZIP code in the range, for example, 00001.
- In the **End** field, type the last ZIP code in the range, for example, 00300.

- g. Click **OK** to save your settings and return to the Jurisdictions page. A confirmation dialog displays. Click **OK**. The new jurisdiction now displays in the list.
- h. Repeat steps 1d through 1g to define another jurisdiction.

Step 2. Defining shipping modes

- a. Click **Shipping modes**. The Shipping Modes page displays.
- b. To define a shipping mode, click **New**. The Create New Shipping Mode page displays.
- c. To define a new carrier, select **Create new**. Type the name of the new carrier in the **Name** field, for example XYZ Carrier.
- d. To define a new service, select **New service**, then type the name of the service in the **New service** field, for example, Overnight.
- e. Define the following optional attributes as necessary:

Description

Type a short phrase that describes the shipping mode, for example, XYZ Carrier, Overnight shipping mode.

Additional description

Add more information about the shipping mode, for example, \$15.00 for Overnight shipping.

Estimated delivery time description

Type a phrase that describes the delivery time, for example, 2-3 days, or 24 hours.

Carrier tracking URL

Type the URL address that customers can use to find out more about the status of their shipment, for example, www.xyzcarrier.com.

- f. Click **OK** to save your settings and return to the Shipping Modes page. A confirmation dialog displays. Click **OK**. The new shipping mode now displays in the list.
- g. Repeat steps 2b on page 34 through 2f to define another shipping mode.

Step 3. Defining shipping codes

- a. Click **Shipping codes**. The Shipping Codes page displays.
- b. To define a new shipping code, click **New**. The Define Shipping Code Details page displays.
- c. In the **Name** field, type the name of the shipping code.
- d. In the **Description** field, type a description of the new shipping code.
- e. Click **Next**. The Assign Shipping Codes to Products or Categories page displays.
- f. Select the products or categories that will be associated with this shipping code:

All products

All products in the catalog will be associated with this code.

Specific product(s)

Either find the products or browse the catalog to select specific products to be associated with this code.

Categories

- 1) Click **List** to view the categories in the catalog. The Category List page displays.
 - 2) Select the category you want to associate with the shipping code, and click **Add**. The category displays in the **Category** field on the Assign Shipping Codes to Products or Categories page.
- g. Click **Finish** to save your settings and return to the Shipping Codes page. A confirmation dialog displays. Click **OK**. The new shipping code displays in the list. A second dialog box displays, informing you that you must create shipping charges associated with this shipping code. Click **Ok**.

Step 4. Defining shipping charges

- a. Click **Shipping codes**. The Shipping Codes page displays.
- b. Select the shipping code to which the shipping charge will apply. Click **Charges**. The Shipping Charges page displays.

- c. To define a new shipping charge, click **New**. The Define Charge Details page displays.
- d. In the **Name** field, type the name of the shipping charge.
- e. In the **Description** field, type a description of the new shipping charge.
- f. In the **Time period** section, select the time period when the charge is in effect.
- g. On the Define Charge Type page, select one of the following radio buttons:

By quantity

Charges are based on the number of items shipped.

- 1) From the **Currency** drop-down list, select the currency for the charge.
- 2) In the Ranges table, do the following:
 - To add a range, click **Add**. In the **Start of range** field, type the number that starts the range. For example, 0 if the range is from 0 to 5 items per order.
 - To add another range repeat the step above. Once you enter a second range, the end number of the first range is automatically entered. For example, if your first range was 0 to 5, your second range would start with 6.
 - In the **Charge** column, type the amount to be charged for that range. For example, the amount to ship an order that contains 0 to 5 items might be \$5.00. Note that this amount is in the currency you selected from the drop—down list. If you want to define amounts in a second currency, select a new currency from the list.

By Weight

Charges are based on the weight of the order shipped.

- 1) From the **Unit of measure** drop-down list, select an appropriate weight unit of measurement.
- 2) From the **Currency** drop-down list, select the currency for the charge.
- 3) In the Ranges table, do the following:
 - To add a range, click **Add**. In the **Start of range** field, type the number that starts the range. For example, 0 if the range is from 0 to 5 kilograms.
 - To add another range repeat the step above. Once you enter a second range, the end number of the first range is automatically entered. For example, if your first range was 0 to 5, your second range would start with 6.
 - In the **Charge** column, type the amount to be charged for that range. For example, the amount to ship an order that weighs 0–5 kilograms might be \$5.00. Note that this amount is in the currency you selected from the drop-down list. If you want to define amounts in a second currency, select a new currency from the list.

Fixed Charges that are independent of quantity or weight.

- 1) From the **Currency** drop-down list, select the currency for the charge.

- 2) In the **Amount** field, type the amount of the fixed charge, for example, \$6.00.

Note that this amount is in the currency you selected from the list. If you want to define amounts in a second currency, select a new currency from the list.

- h. Click **Finish** to save your settings and return to the Shipping Charges page. The new shipping charge displays in the list.

Step 5. Activating shipping codes

- a. Click **Shipping codes**. The Shipping Codes page displays.
- b. Select the shipping code to be activated. Click **Activate**. The value in the Status column changes to Activated.

Step 6. Updating registry components

- a. Open the Administration Console:

▶ Developer

`https://host_name/webapp/wcs/admin/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogon`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

▶ 400

▶ Linux

▶ Windows

`https://host_name:8002/adminconsole`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server.

- b. Select **Site** on the Administration Console Site/Store Selection page.
- c. From the Configuration menu, click **Registry**. A list of registry components for the site displays.
- d. Click **Update All** to update all of the listed Registry components. The Registry window reloads listing the status for all components as Updating.
- e. Click **Refresh** to reload the Registry window and check on the status of components you are updating. When updating is complete, the status column reads Updated.

Chapter 6. Taxes

This chapter describes how to use the WebSphere Commerce Accelerator to add taxes to orders. A functional store must include data on taxes. Taxation is about charging and collecting taxes on the goods and services of the store. The combination of tax categories, calculation codes, jurisdiction, and jurisdiction groups create the tax charges for the store.

Specify tax settings for your store using the Tax notebook. This notebook allows you to set up or change tax information so that taxes are displayed on each order. Before you specify tax rates for your store, ensure that you contact your taxation agency to determine the types and rates you should charge.

Implementation note

If the tax implementation supported by the Tax notebook is not appropriate for the store, WebSphere Commerce supports integrating with a third-party tax system. WebSphere Commerce also supports customizing the tax implementation. For information about customizing the tax implementation refer to the topic "Taxes" in the WebSphere Commerce - Express development environment online help.

Use the Tax notebook to specify the following tax information:

Tax jurisdictions

The tax jurisdictions to which you deliver orders. The jurisdictions are geographical regions, made up of a country or region and state or province combination. For example, CANADA, Ontario is a jurisdiction. WebSphere Commerce creates a default tax jurisdiction, WORLD, for every store. The WORLD jurisdiction cannot be deleted.

Tax categories

The tax categories for your store. Create a category for each type of tax you must collect for your store. For example, create a GST category for the GST, or a State category for the state tax. You can create as many categories as necessary for your store. In the Tax notebook, sales tax categories are applied to the total price of the order, while shipping tax is applied to the total shipping charge for the order. The Tax notebook does not apply taxes on top of other taxes.

Display name of the tax category

The tax category name that displays to your customers. If your store displays in multiple languages, you can define a display name for each language. For example, the name of the tax category may be translated into Japanese or German, or any language in your store.

Tax rates

The tax rates (as percentage values) charged to customers. Each rate is associated with a tax jurisdiction and tax category. For example, you can specify a 10% shipping tax for all orders shipped to Canada and a 12% shipping tax for orders shipped outside of Canada.

Tax calculation codes

The sales and shipping tax calculation codes that specify how sales and

shipping taxes are calculated. For example, all sales taxes applicable to groceries could be grouped under a single sales tax code called groceries. A store typically only supports a single tax code scheme. WebSphere Commerce creates a default code that applies to any product that is not assigned to a tax code. After creating tax codes for your store, you assign the applicable tax categories to each code.

Setting up taxes

Step 1. Open the Tax notebook

a. Open the WebSphere Commerce Accelerator.

 Developer

`https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=common.mcLogon`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

 400  Linux  Windows

`https://host_name:8000/accelerator`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server machine.

b. From the **Store** menu, select **Change Tax**. The Tax notebook opens.

Step 2. Change tax jurisdictions

a. From the left navigation frame, click **Jurisdictions**.

b. To add a jurisdiction, do the following:

- From the **Country/Region** drop-down list, select a country or region.
- Use the **State/Province** drop-down list or field as follows:
 - If you selected Canada, United States, or Japan as your country or region, do one of the following:
 - From the drop-down list, select a state or province for the chosen country or region.
 - If one tax rate applies to the entire country or region, from the bottom of the list, select **All**.
 - If you selected a country or region other than Canada, United States, or Japan, do one of the following:
 - From this field, type the name of the state or province for the chosen country or region.
 - Leave the field blank if:
 - The tax rate applies to the entire country or region.
 - The country or region does not have states or provinces.

c. Click **Add** to add the combination of country or region and state to the **Defined jurisdictions** list.

d. Repeat steps 2b to 2c to add more tax jurisdictions.

Note: You can create an unlimited number of tax jurisdictions; however, if you create more than 50 jurisdictions, the Rates page will take a long time to display.

e. Click **OK** to save your settings and exit.

Step 3. Change sales tax codes

- a. From the left navigation frame, click **Sales Tax**.
- b. To add a code, do the following:
 - 1) In the **New sales tax code** field, type the name of the sales tax code.
 - 2) Click **Add**. The tax category displays in the **Defined sales tax codes** list.
- c. Set the default tax code
 - From the **Define sales tax codes** list, select the code, then click **Set as Default**. The default sales tax code is applied to all products that are not currently assigned to a sales tax code.
- d. To delete a code select the code, in the **Defined sales tax codes** list; then click **Remove**.
- e. Click **OK** to save your settings and exit.

Step 4. Change the shipping tax codes

- a. From the left navigation frame, click **Shipping Tax**.
- b. To add a code, do the following:
 - 1) In the **New shipping tax code** field, type the name of the shipping tax code.
 - 2) Click **Add**. The tax category displays in the **Defined shipping tax codes** list.
- c. To set a default shipping tax code select a tax code from the **Defined shipping tax codes** list; then click **Set as Default**. Default shipping tax codes will be assigned to all products that are not currently assigned to a tax code.
- d. To delete a code select the code in the **Defined shipping tax codes** list; then click **Remove**.
- e. Click **OK** to save your settings and exit.

Step 5. Change the tax categories

- a. From the left navigation frame, click **Categories**.
- b. To add a category, do the following:
 - 1) In the **New tax category** field, type the name of the tax category.
 - 2) Select the tax type of the category. Each category must be either a sales tax or a shipping tax type. Sales tax is charged on the total amount of the order. Shipping tax is charged on the shipping charges for the total order.
 - 3) Select **Include tax as display price** if you want to include the tax amount in the price that displays for the product. If you do not select this option, the tax amount will display separately.

Important: To display the tax as a display price on your store page, your Site Administrator or store developer will have to make some changes to the store page. For more information, see “Displaying tax as part of the display price” in the WebSphere Commerce development environment online help.
 - 4) Click **Add**. The tax category displays in the **Defined tax categories** list.
 - 5) Repeat step 4 to add more tax categories.
- c. To delete a category, select the category, then click **Remove**.
- d. Click **OK** to save your settings and exit.

Step 6. Change the tax category display name

- a. From the left navigation frame, click **Display Names**.
- b. From the **Language** drop-down list, select the language.
- c. In the **Display Name** fields, type the name for the tax category that will display to the customers, in the language you selected in step 4 on page 41.
- d. Repeat steps 6b and 6c for each language your store supports.
- e. Click **OK** to save your settings and exit.

Step 7. Change tax rates

- a. From the left navigation frame, click **Rates**.
- b. The tax rates table contains a default tax, **Shipping**, which has an initial value of 0.00. In the **Shipping** column type the tax rate charged for shipping for each jurisdiction within the table.
Leave the value as zero if you do not charge a shipping tax.
- c. In the rest of the cells, type the tax rate for each combination of tax jurisdiction and tax category. Specify the rate as a percentage value.
- d. Click **OK** to save your settings and exit.

Step 8. Assign categories to codes

- a. From the left navigation frame, click **Category Assignment**.
Note that the table contains a default tax code, **GST**, and a default tax, **Shipping**.
- b. Assign tax categories to the appropriate tax codes, by selecting the corresponding check box.
- c. Click **OK** to save your settings and exit.

Step 9. Update registry components

- a. Open the Administration Console:

 Developer

`https://host_name/webapp/wcs/admin/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogon`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

 400  Linux  Windows

`https://host_name:8002/adminconsole`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server.

- b. Select **Site** on the Administration Console Site/Store Selection page.
- c. From the Configuration menu, click **Registry**. A list of registry components for the site displays.
- d. Click **Update All** to update all of the listed Registry components. The Registry window reloads listing the status for all components as Updating.
- e. Click **Refresh** to reload the Registry window and check on the status of components you are updating. When updating is complete, the status column reads Updated.

Chapter 7. E-mail notification

This chapter describes how to set up messages for various events. Before completing the steps in this chapter ensure that you have a mail server set up.

The following messages are supported for the Express Store:

Authorized order

Notification message for an authorized order

Canceled order

Notification message for a canceled order

Password reset

Notification message for password reset

Shipping notification

Notification message to the customer of an order release manifestation

Submission order

Notification message for a received order

Wish list

A broadcast message

To change the default text of the message, follow the procedure described in the section “Changing text on store pages and notification messages” on page 21.

Setting up e-mail notification

To enable e-mail notification, do the following:

1. Ensure that you have a mail server set up and configured.

Important

The SMTP server used should adhere to the following RFC specifications:

- a. E-mail address adhering to RFC 822 (stores should enforce this for registration).
- b. POP3 server as defined in RFC 1939 capable of delivering Delivery Status Notification (DSNs) as defined in RFC 1894.
- c. SMTP server as defined in RFC 2821.

For information on RFC specifications, refer to www.ietf.org/rfc.html.

Developer If you will not use e-mail notification, ensure that you disable the smtp transport.

2. Ensure that the IBM WebSphere Application Server administration server is started.
3. Open the Administration Console:

Developer

`https://host_name/webapp/wcs/admin/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogon`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.



`https://host_name:8002/adminconsole`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server.

4. On the Administration Console Site/Store Selection page select **Store**. The Select Store and Language section displays.
5. From the **Name** list, select the store.
6. From the **Language** drop-down list, select the language. Click **OK**. The Store Administration Console home page displays.
7. From the **Configuration** menu, click **Transports**. The Transport Configuration page displays.
 - a. Verify that the status of the **E-mail** transport is Active.
If e-mail is inactive, select it, then click **Change Status**.
 - b. Select **E-mail**; then click **Configure**. The Transport Configuration Parameters page displays.
 - c. In the **Host** field, type your fully qualified mail server name or ip address, for example, `myserver.ibm.com`.
 - d. In the **Protocol** field, type `smtp` in lowercase letters, or the protocol of your choice. Click **OK**.
 - e. If required, in the **Port** field, change the default port number, 25, to an appropriate value.
8. From the **Configuration** menu, click **Message Types**. The Message Type Configuration page displays.
9. Create the notification to be sent when payment is authorized as follows:
 - a. Click **New**. The Message Transport Assignment page displays.
 - b. Select the message type from the **Message Type** drop-down list. See the table above for the message types to use in your store.
 - c. In the **Message Severity** field, type 0 to 0
 - d. From the **Transport** drop-down list, select **E-mail**. If e-mail is not there, do the following:
 - 1) From the **Configuration** menu, select **Transports**. The Transport Configuration page displays.
 - 2) Click **Add**. The Add Transport window opens.
 - 3) Select the check box next to the transport you want to add to the store. You can select all transports by selecting the check box at the top-left. If there are no transports available, then you have already added all of the transports made available by the Site Administrator.
 - 4) Click **Add** to add the transport.
 - e. From the **Device Format** drop-down list, select **Standard Device Format**.
 - f. Click **Next**. The Message Transport Assignment Parameters page displays.
 - g. Complete the fields as follows:

Host The fully qualified name of your mail server or ip address, for example, `example.ibm.com`

Protocol

Type smtp in lowercase letters. Or, if you are not using smtp, type the protocol you are using.

Recipient

Enter a valid e-mail address. This address will be replaced by the customer's e-mail address at run time.

Sender

Enter an e-mail address that you want to use as the sender of the message, for example, orders@example.ibm.com. The address must be an e-mail address for a valid user on the mail server.

Subject

Enter the text that you want to display as the subject line of the message, for example, Your order has been accepted.

- h. Click **Finish**. The Message Type Configuration page displays.
- 10. For each message type in your store, repeat step 9 on page 44.

Notes:

It will often take a long time before canceled order notification e-mail is sent. The time can be shortened by changing the PayCleanup job to have a smaller schedule interval.

Part 2. Loading store information

The Loader package loads catalog and other data into your WebSphere Commerce database. The Loader package is ideal for importing large amounts of existing product information into the database or making cell-level updates to existing data. An example of cell-level updates includes updating a value in a specific row and column. Superior performance and scalability features make the Loader package appropriate for the mass population and subsequent updates of the WebSphere Commerce Server data.

The Loader package uses data in XML format as input into the WebSphere Commerce database. The package lets you generate a document type definition (DTD), resolve identifiers in the data, and load data. The Loader package also includes utilities for data extraction and XML data transformation. The Loader package now contains TextTransformer, which transforms an ASCII-delimited document into an XML document.

If you create an XML file using the DTD provided with WebSphere Commerce or modify the XML files provided with the sample store archive, you do not need to generate another DTD.

Included with the Loader package is a logging facility that lets you capture and store messages from package components. For example, you can log errors that occur while generating a DTD or loading data for debugging purposes.

Chapter 8. Loading the catalog

In WebSphere Commerce you can create a catalog either by using the Product Management tools in the WebSphere Commerce Accelerator, or by using the Loader package to mass load the catalog. Usually, a catalog is mass loaded, then maintained by using the Product Management tools.

Creating catalog data in XML format

Use the sample catalog as a base to learn how to create your store's catalog. Review the comments in the XML files for information about which database tables are populated and the relationships between the data. The chapter "*Catalog assets*" in the *WebSphere Commerce Store Development Guide* provides additional directions on how to create catalog assets. Some of the assets described in the *Catalog assets* chapter are specific to a store that has ATP enabled. Since the Express Store does not enable ATP you do not need to create ATP-specific assets.

Creating catalog data in CSV format

If catalog data is provided in the form of text delimited data, several steps are necessary to load CSV data to the WebSphere Commerce database.

1. Cleanse the CSV data.
2. Transform the CSV data to generic XML data
3. Generate a DTD for the WebSphere Commerce Server database
4. Transform the generic XML data to WebSphere Commerce XML data
5. Resolve the identifiers in the XML document
6. Load the data into the WebSphere Commerce Server database

For more information refer to the following resources:

- For a detailed description on how to create a commerce catalog from CSV data, refer to the chapter entitled "*Aggregate and transform data for commerce catalogs*" in the publication, *WebSphere Commerce V5.4 Catalog Design and Content Management*. This publication is available from the IBM Redbooks site, <http://www.ibm.com/redbooks/>.
- A sample application, *Example of loading CSV data into a WebSphere Commerce Server 5.4 database*, is available from the [http://www.ibm.com/software/commerce/support/ support](http://www.ibm.com/software/commerce/support/support) page. This sample contains scripts and sample data that load additional categories, products and SKUs into an existing catalog.

Note, if you load the sample data from this sample application to an Express Store, you will need to set inventory levels before you can order the example products from your store. The sample also contains some ATP specific data which is not required in the Express Store; if you use the sample as a base you can remove references to ATP specific tables such as, ITEMVERSN, VERSIONSPC, DISTARRANG and STOREITEM.

Chapter 9. Loading existing customer information

This chapter describes how to load existing customer information from a comma-separated value (CSV) file to the WebSphere Commerce database by using the Loader package. A CSV file contains table data as a series of ASCII text lines. The lines are organized by column. Each column value is separated by a comma from the next column's value, and each row starts a new line. Many spreadsheet programs can export data as CSV files.

1. Create CSV files in the correct format.
2. Transform the CSV data to generic XML data.
3. Generate a DTD for the WebSphere Commerce database.
4. Transform the generic XML data to WebSphere Commerce XML data.
5. Resolve the identifiers in the XML document.
6. Load the data into the WebSphere Commerce database.

Creating CSV files

___ Step 1. The section will describe how to create CSV files for the following tables:

Table name	Description
MEMBER	create a new member entry
USER	creates the user
MBRREL	creates the member relationships, to define the hierarchy to get to the new user node

Optionally, you may also want to load the following data:

USERPROF	user profile
BUSPROF	business profile
ADDRESS	addresses
MBRROLE	roles
MBRATTRVAL	custom member attributes

Transforming the CSV data to generic XML data

___ Step 2. Transform the CSV data received from the Line of Business Manager into a single XML data file. To convert CSV files to XML files you use the Text transform command.

- a. Create schema files.

The text schema files tell the Text Transformer how to parse a particular CSV source file. In particular, each text schema file indicates the values for field, record, and text delimiter as well as whether or not header lines are included in the source file.

- 1) Open an XML or text editor.
- 2) For each CSV file create a corresponding schema file. Replace the ElementName attribute value with the appropriate value.

```
<?xml version="1.0" encoding="UTF-8" ?>
<TextSchema
  DataType = "CSV Format">
  <RecordDescription
    FieldSeparator = ","
```

```

RecordSeparator = "

"
StringDelimiter = ""
HeaderIncluded = "true"
HeaderLines = "1"
ElementName = "xyz">
</RecordDescription>
</TextSchema>

```

3) Save the schema files.

b. Create a manifest file.

The manifest file, `manifest.txt` also referred to as a "command" or "parameter" file, instructs the text transformation tool. The manifest file specifies:

- which files to parse as input
- which text schema files to use for each source file
- which files to use for output
- how to write to the output file (create or append)

The files referenced in the `manifest.txt` are then used as input to the text transformation.

1) In a text editor create the following file:

```

a.csv,a_schema.xml,a_data.xml,Create
b.csv,b_schema.xml,bdata.xml,Append
c.csv,c_schema.xml,c_data.xml,Append

```

2) Save the file as `manifest.txt`.

c. Run the text transform command:

```

Linux 400 txttransform.sh manifest.txt
Windows txttransform.cmd manifest.txt

```

The output file is specified as `c_data.xml`. Matching DTD and schema files, `c_data.dtd` and `c_data.xsd`, are also created.

Generating a DTD for the WebSphere Commerce database

1. Generate DTD file for those tables in the WebSphere Commerce database to which you will be importing data. The DTD Generate command accepts the database name, database user name, database user password, and file containing a list of table names as input. The command writes out a DTD file.

a. Create a file containing a list of table names. Each table name should be on a separate line, specify table names in lowercase letters, for example:

```

address
busprof
mbrattrval
mbrrel
mbrrole
member
user
userprof

```

b. Save the file as `tablenames.txt`.

c. Run the DTD Generate command:

```

dtdgen -dbname db_name -dbuser db_user -dbpwd db_password -infile tablenames.txt -outfile wcsample.dtd

```

where:
dbname

```

Linux Windows Name of the target database.

```

400 This is the database name as displayed in the relational database directory (WRKRDBDIRE).
 dbuser
Linux Windows Name of the user connecting to the database.
400 This is usually the same as the instance user name.
 dbpwd Password for the user connecting to the database.
 infile Name of an input file containing a database table name on each line.
 outfile Name of the output DTD file.

The command creates the *wcsample.dtd* file.

Transforming the generic XML data to WebSphere Commerce XML data

2. Transform the XML data created in 2 on page 51 into an XML document that conforms to the DTD created in 52.

To transform generic XML data to WebSphere Commerce XML data you use the XML Transform command. The XML Transform command takes an XML source file and an XSL file as input. The XML Transform command writes out a new XML document that conforms to the DTD defined by *wcsample.dtd*.

The XSL file is essential in this process. It accomplishes two important tasks:

- a. It defines the mapping of data from the source DTD into the target DTD.
- b. It defines the special identifiers needed for the ID resolution that occurs in 3.

The sample XSL file includes comments that explain many of the variables, parameters, and logical elements that are contained in the XSL document.

- a. Run the XML transform command,

```
xmltransform -infile c_data.xml -transform file.xsl
-outfile wcdata.xml -param
```

where:

dbname
Linux Windows Name of the target database.
400 This is the database name as displayed in the relational database directory (WRKRDBDIRE).
 dbuser
Linux Windows Name of the user connecting to the database.
400 This is usually the same as the instance user name.
 dbpwd Password for the user connecting to the database.
 infile Name of the input XML document containing table records
 outfile Name of the output DTD file.

Resolving identifiers in the XML document

3. Next, resolve the identifiers in the XML document, *wcdata.xml* created in 2. To resolve entities you use the ID Resolve command. The ID Resolve command accepts the database name, database user name, database user password, source XML document, and method of resolution (mixed) as input. The

command writes out a new XML document as output. This file has all of the primary and foreign keys values in place within the XML document. (wcdataid.xml)

a. Run the ID Resolve command:

```
idresgen -dbname db_name -dbuser db_user -dbpwd db_password
        -infile wcdata.xml -outfile wcdataid.xml -method mixed
```

where:

dbname

Linux Windows Name of the target database.

400 This is the database name as displayed in the relational database directory (WRKRDBDIRE).

dbuser

Linux Windows Name of the user connecting to the database.

400 This is usually the same as the instance user name.

dbpwd Password for the user connecting to the database.

infile

Name of the input XML document containing table records.

outfile

Name of the output XML file to be produced; this file can be used as input to the Loader.

method

Method to be used in processing the input file. The default method is load. The load method treats the input file as though the records do not exist in the database. The update method assumes that there are already identifiers for the input objects. Use the mixed method when some records do not exist in database and some do.

Loading the data into the WebSphere Commerce database

4. Finally, load the XML data into the WebSphere Commerce database.

a. `massload -dbname db_name -dbuser db_user -dbpwd db_password
 -infile wcdataid.xml -method sqlimport`

Part 3. Adding additional features to the Express Store

This part of the book introduces some tasks involved adding additional features to the Express Store. This section is intended to be used by application developers who have knowledge in the following areas:

- IBM WebSphere Studio Application Developer
- IBM DB2 Universal Database Express Edition
- Java programming language
- JavaServer Pages technology
- HTML
- XML

Chapter 10. Store front architecture

WebSphere Commerce uses a system of *commands*, *views* and *data beans* to display the Web assets in a store front to customers.

Commands

A *command* is a Java bean that contain the programming logic associated with handling a particular request. Commands perform a specific business process, such as adding a product to the shopping cart, processing an order, updating a customer's address book, or displaying a specific product page. When the action is completed, the command returns a view.

A command is invoked by using a URL. For example, when a customer clicks **Shopping Cart** in the Express Store, the customer invokes the URL `https://hostname/path/OrderItemDisplay?`, which is passed into the WebSphere Commerce Server. The WebSphere Commerce Server calls the `OrderItemDisplay` command, and the shopping cart page is displayed to the customer.

The `ProductDisplay` and `CategoryDisplay` commands return views as well as JSP filenames. These JSP filenames, which display products and categories are stored in the catalog data. You can optionally assign different JSP filenames to display products and categories for each member group or language supported by your store.

The WebSphere Commerce Server can also map multiple commands to a URL, which allows each store to optionally have its own implementation of that command.

Some commonly used commands are:

StoreCatalogDisplay

Displays the home page for the store.

CategoryDisplay

Displays a category, or department, of a store; when a customer clicks a category, this command displays the products or subcategories under the category.

ProductDisplay

Displays a product page.

OrderItemDisplay

Displays the Shopping Cart page.

UserRegistrationForm

Displays a Registration page for entering or updating profile information.

LogonForm

Displays a logon page.

Views Display the results of commands and user actions, that is, views present your store pages (JSP files) to the customers. In order for the view to invoke a JSP file, the JSP filename must be registered with the view in the view registry (VIEWREG) table. The corresponding JSP file is stored using the JSP filename in the subdirectory (storedir) for the store under the WCS Stores webapp doc root.

When a customer clicks **Help** in the Express Store, the customer invokes the URL `https://hostname/path/HelpView?`, which is passed into the WebSphere Commerce Server. The WebSphere Commerce Server calls the `HelpView`, which returns the Help page.

Similarly, the WebSphere Commerce Server also allows you to map multiple JSP files to a single view, where each store can optionally register different JSP filenames for different device types

Data beans

A type of Java bean that is placed into a JSP file. The data bean represents data that is accessible to the application developer.

Some commonly used data beans are:

ErrorDataBean

Displays error information on a JSP.

ResourceBundleDataBean

Obtains information from properties files.

StoreDataBean

Contains store-specific information, such as the store identifier and catalog identifier for the store.

OrderDataBean

Represents an order.

UserRegistrationDataBean

Represents the registration information for a member.

ProductDataBean

Represents a product and all of its attributes.

Default commands and views

WebSphere Commerce provides default commands and views which you can use in your store. These default commands and views are listed in the `wcs.bootstrap.xml` file. The bootstrap files are located in the following directory:

`WC_installdir/schema/xml`

Additionally, many of the views used in Express Store were created specifically for Express Store. These views are listed in the `command.xml` file in the Express Store store archive. If a needed command or view is not provided, you can create your own. For information on creating commands and views, see the *WebSphere Commerce Programming Guide and Tutorials*.

Express Store conventions

The Express Store follows naming conventions for the following items:

- JavaServer Pages
- JSP templates
- Properties files
- Location structure and directory structure

JavaServer Pages (JSP)

All JSP files used in the Express Store conform to the XHTML 1.0 Transitional DOCTYPE. A few of the Express Store JSPs contain the following XHTML validation errors:

- **background** attribute on the TD tag (used to display an image). These images are dynamic; because they change depending on the style and banner selection in the store.
- **EMBED** tag. The EMBED tag was deprecated in XHTML 1.0. It is maintained on some store pages so that customers using older versions of Netscape browsers see a download URL for the Flash plugin.

You may safely ignore these validation errors.

Follow these conventions when naming JSP files:

- Use camel case style for all JSP files, for example CamelCase.jsp.
- Begin JSP file names with a noun (CatalogSearchForm.jsp) or adjective (AdvancedCatalogSearchForm.jsp).
- Do not repeat the section name in the JSP file name. For example, AddressBookDisplay.jsp belongs to the Account section, but the JSP file does not need to include *Account* as part of the name.
- Do not use short forms in the JSP name, unless using the full name would exceed the maximum directory length limit.

There are two types of JSPs:

1. Display JSP files

Present the visual exterior of a store page; and contain Java display logic, HTML and JSP custom tags. They should not contain business logic or Java code. Note that some display JSPs in the Express Store sample contain Java code.

2. Non-display or setup JSP files

Any JSP file which does not display information. A setup JSP file may prepare data. For example, EnvironmentSetup.jsp retrieves and prepares the JSP path, file path, and resource bundle. Other JSP pages use this information. A setup JSP file may also be used for conditional routing. An example of a JSP used for routing is AllocationCheckSetup.jsp. This JSP file includes different JSP files that are based on different conditions in the database. It shows the ProductAvailability page if only some items are available. And it shows the OrderSummary page, if all items are available, or if all items are unavailable.

The suffix, or sometimes the prefix, of a JSP file name indicates its type. Use the following conventions to classify JSP files:

JSP file suffixes

Setup Setup JSP files, for example, EnvironmentSetup.jsp.

Form or Display

Display JSP files, for example, SideBarDisplay.jsp. A display JSP may contain an input form.

Notify An e-mail JSP, such as PasswordResetNotify.jsp.

JSP file prefixes

Cached

A JSP file designed to be cached. Sometimes a JSP is divided so that it can be cached. Examples of cached JSP files are: CachedSidebarDisplay.jsp, and CachedHeaderDisplay.jsp.

Naming conventions for variables within JSP files

Use these conventions for variable names:

- Use descriptive names
- Start with a lower case letter.
- Use camel case. In the following example the variable, *storeId*, has the same name as the URL parameter.

```
String storeId = request.getParameter("storeId")
```

JSP templates

The Express Store uses a JSP template mechanism. JSP templates maintain a set of common files. This allows the application developer to reflect changes easily. It also provides a common look and feel for all store pages. These files implement a common set of functions. Most Express Store pages link to these JSP template files:

index.jsp

Provides an entry point to the store. This JSP forwards the page to appropriate URL for the store front. Each of the commands that are used as a storefront command has some required parameters, namely, *storeId*, *languageId*, and *catalogId*. The value of the *storeId* variable and the *catalogId* variable are known after the store has been published. The file, *index.jsp*, is not the include file; the include file is *parameters.jsp*. The *parameters.jsp* include file contains the value for the *storeId* variable and the value for the *catalogId* variable. Publishing the store updates the *parameters.jsp* file.

HeaderDisplay.jsp

This file contains common links at the top of the page.

FooterDisplay.jsp

This file contains common links at the bottom of the page.

SidebarDisplay.jsp

This file contains a choice of language it is included after the *Header.jsp*.

EnvironmentSetup.jsp

This JSP file includes access to the resource bundle, which contains the locale-specific text for all the store pages. In this way, you can maintain a multilingual store from one set of JSP files. The *EnvironmentSetup.jsp* file contains all commonly needed attributes, such as *StoreDir*, *includeDir*, *fileDir*, and *bundleDir*. Almost all JSP files in the store call this JSP file.

Properties files

The name of the properties file should be: *StoreName_locale.properties*. Stores that support multiple languages should have a default properties file called *StoreName.properties*. WebSphere Commerce searches for strings in the *StoreName.properties* file then the locale-specific property file.

Use these conventions for properties files:

- Begin each section with a comment that identifies the section.
- Place all text strings in a properties file.
- Prepend all strings on the page with the same identifier. The identifier uniquely identifies the page.
- Assemble text messages dynamically. Use placeholders in the properties file for the substitutions.

- Name strings descriptively. For example, use `logon_userid`, `logon_password` instead of `logon_column1` or `logon_field1`.

Location structure and directory structure

Environment	File path
JSP files	
▶ Developer	<code>workspace_dir/Stores/Web Content/ExpressStore</code>
▶ 400 ▶ Linux ▶ Windows	<code>WAS_installdir/installedApps/nodename/WC_instancename.ear/Stores.war/ExpressStore</code>
Properties files	
▶ Developer	<code>workspace_dir/Stores/Web Content/WEB-INF/classes/ExpressStore/</code>
▶ 400 ▶ Linux ▶ Windows	<code>WAS_installdir/installedApps/nodename/WC_instancename.ear/Stores.war/WEB-INF/classes/Express</code>
Change Flow files	
▶ Developer	<code>workspace_dir/Stores/Web Content/WEB-INF/xml/tools/stores/ExpressStore/</code>
▶ 400 ▶ Linux ▶ Windows	<code>WAS_installdir/installedApps/nodename/WC_instancename.ear/Stores.war/WEB-INF/xml/tools/stores/ExpressStore/</code>
Store 'Style Editor' thumbnails	
▶ Developer	<code>workspace_dir/Stores/Web Content/tools/stores/ExpressStore/style</code>
▶ 400 ▶ Linux ▶ Windows	<code>WAS_installdir/installedApps/nodename/WC_instancename.ear/Stores.war/tools/stores/ExpressStore/style</code>
Change Pages' preview pages	
▶ Developer	<code>workspace_dir/Stores/Web Content/tools/stores/ExpressStore/text</code>
▶ 400 ▶ Linux ▶ Windows	<code>WAS_installdir/installedApps/nodename/WC_instancename.ear/Stores.war/tools/stores/ExpressStore/text</code>

Chapter 11. Changing store pages

This chapter describes how to change JSP pages provided with the Express Store sample. The Registration page is used as an example; but the same methods can be applied to any pages in the Express Store sample.

Some of the most commonly used data beans and commands in the Express Store are listed below:

Commonly used data beans

ErrorDataBean

Displays error information on a JSP.

ResourceBundleDataBean

Obtains information from properties files.

StoreDataBean

Contains store-specific information, such as the store identifier and catalog identifier for the store.

OrderDataBean

Represents an order.

UserRegistrationDataBean

Represents the registration information for a member.

ProductDataBean

Represents a product and all of its attributes.

Many form elements in the Express Store use the **ECUserConstants** class. This class contains constants that specify the name of a form element, for example:

```
<input size="25" maxLength="50"  
names="<%=ECUserConstants.EC_ADDR_FIRSTNAME %>"  
type="text" value="<%=strFirstName%>">
```

Changing the Express Store Registration page

The Express Store registration page allows customers to register at the store. When customers register, they must provide their first and last names and an e-mail address, and then create a password; these fields are required. Additionally, the Registration page may also display the following fields: **Preferred language**, **Preferred currency**, **Age**, **Gender** and **Send me promotional e-mail**; these fields

are optional and may be removed by using Change Flows.

The Express Store Registration page can be changed to enable it to collect additional customer information.

1. Open the Registration page, `UserRegistrationAddForm.jsp`.
2. Place one or more fields in the desired location on the form.
3. Name each field by its associated `UserRegistrationAdd` parameters using the `ECUserConstants` class.
4. Set the values for the page element in the properties file.
5. Edit the JSP to get the text values for that element from the properties file.
6. Test the changes.

Detailed reference information about commands and data beans is provided in the WebSphere Commerce - Express development environment online help. A list of the commands, data beans and access beans used in the Express Store is described in the section Appendix B, "Data beans, access beans and commands used in the Express Store," on page 171.

Modifying property files

In the Express Store all text displayed on store pages is contained in locale-specific properties files. This means that for each of the ten languages available in the Express Store the text is located in a separate file, one per language.

- `/Express/properties/storetext_dynamic_locale.properties`

This file contains most of the text on Express Store pages; and the text in notification messages. Use the Change Page option to modify the text in this property file. The Change Page feature is described in the section "Changing text on store pages and notification messages" on page 21.

- `/Express/properties/storetext_locale.properties`

This file contains error messages and field labels used in the Express Store. You can not use the Change Page option to modify this text. To change this text edit the file, then stop and restart the WebSphere Commerce Server. Directions for restarting the server are included in the *WebSphere Commerce Installation Guide* for your platform and edition.

Chapter 12. Adding a page to the store

This section provides an overview of how to add an additional page to your store. This section does not describe how to add additional business logic, such as extending an existing command or writing a new command, for information on extending WebSphere Commerce refer to the *WebSphere Commerce Programming Guide and Tutorials*. The general steps to add a page to the store are as follows:

- Develop a list of store pages needed
- Develop a list of command and view URLs
- Associate JSP filenames with views
- Set access control for the page
- Plan your caching strategy

For a more detailed explanation of developing store pages, refer to the *WebSphere Commerce Store Development Guide*. For more information about access control refer to the *WebSphere Commerce Security Guide*.

Developing a list of store pages

In order to develop a list of the pages needed to create your store, you need to know the business and functional requirements of the store, as well as any business processes that have been defined.

Many people gather requirements in the form of use cases. Use cases define the business processes in your store, in the form of interactions between the customer and the proposed system. In the case of an online store, use cases may define how a customer registers at the store, browses the catalog, or orders an item.

A set of use cases, detailing the business processes for the Express Store are provided in Appendix A, "Use cases," on page 133. These use cases can help you to more thoroughly understand the flow of the Express Stores, and can be used as a guide to create additional use cases for your own store.

Determine the store shopping flow

Regardless of whether you develop use cases to illustrate your store's business processes, or use another method, once business processes are available, you can create the shopping flow for your store. Since use cases often contain flow information such as, "If the customer selects **Submit**, the Order page displays," use cases can provide useful information for creating shopping flow diagrams. The shopping flow reflects the requirements and business processes defined for your store, illustrating how a customer will move through the store. For example, a customer may enter your site through the home page and be asked to register before browsing the catalog, or you may choose to allow customers to view the catalog as guests, without registering. Some shopping flows allow customers to complete a "quick checkout", while others require that a customer completes all checkout steps every time they make a purchase. Or, your shopping flow can offer customers the choice of both checkouts. To verify that the store flow diagram is complete, ensure that all steps in the use cases for your store are illustrated in the store flow diagram.

Mapping out the shopping flow visually, as the following diagram for the Express Store's shopping flow does, allows you to see how customers will travel through your store.

The diagram for the Express Store shopping flow is quite simple. Although it includes the main flow of the a customer's journey through the store, it does not include any error scenarios. For example, what happens when a customer logs in using the wrong password, or enters an invalid credit card number? However, even a simple diagram like this allows you to develop a list of pages needed for the store. To start you will need to create a view for every page listed in the shopping flow diagram.

Error pages

The exception flows in your use cases can also help you determine what error pages you need to create for your store. The registration use case for Express Store specifies the following exceptions flows:

- E-mail address already exists: If the e-mail address already exists in the system, the system displays an error message asking the user to enter another e-mail address. The use case resumes from beginning.
- Missing mandatory fields: If any of the following fields (E-mail, Password, Verify password, First name, Last name) are not completed, the system issues an error message. The use case resumes from beginning.
- Invalid password: If the password does not match the verification password, the system issues a warning.

For every new exception flow that you create, you will need to create either an error page or an error message.

Developing a list of command and view URLs

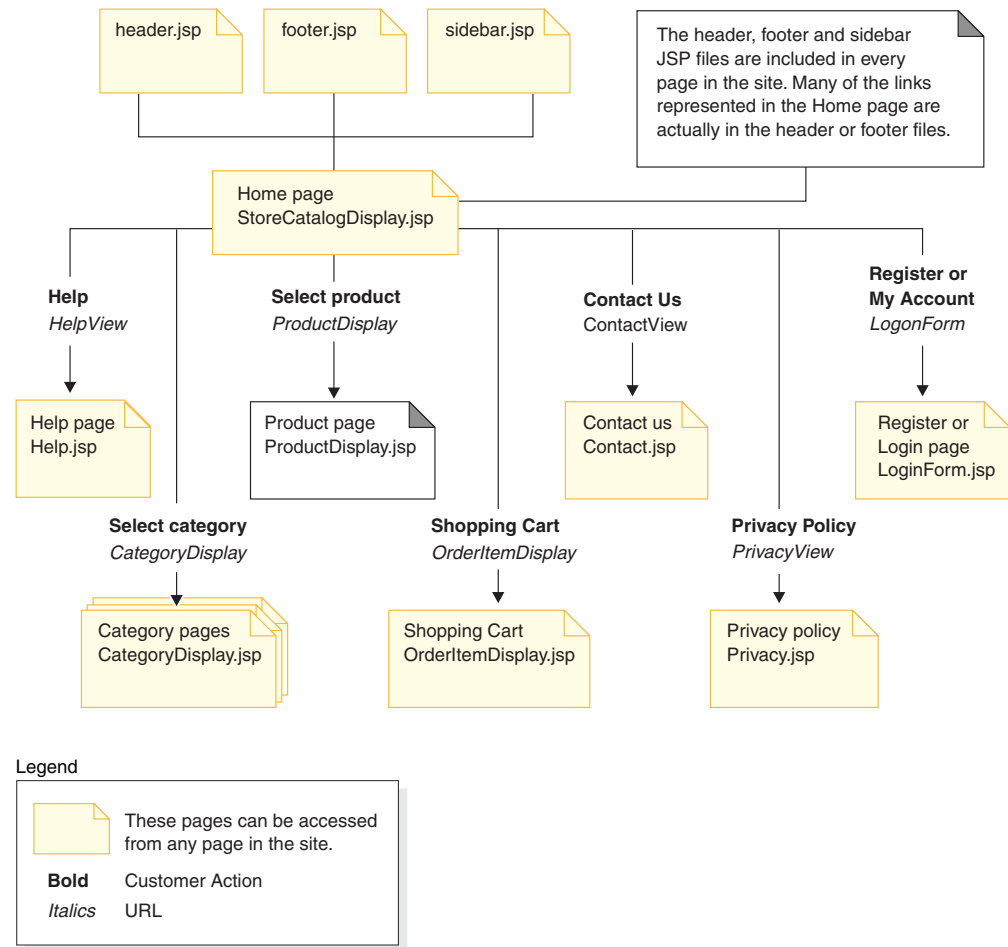
As demonstrated in the Express Store shopping flow diagram, business processes, such as checkout and register, may require several pages. In order to combine these pages into a working business process or flow, rather than just a collection of pages, you must include commands and views in your pages.

Developing a list of URLs needed

Just as you developed a list of pages necessary to create the store, you also need to develop a list of the command and view URLs necessary to implement the business processes for your store. Using the shopping flow diagram for your store, and the list of default commands and views, identify the URLs necessary to complete each action.

Understanding which command and view URLs are used in the Express Stores may also help you determine what URLs you need in your store. The following illustration identifies the URLs for some of the actions in the Express Store

shopping flow diagram.



Associating JSP filename to views

The WebSphere Commerce Server uses view commands to compose a view as a response to a request. WebSphere Commerce Server provides the following view commands:

HttpForwardViewCommandImpl

This view command forwards the view request to a JSP file.

Use the `HttpForwardViewCommandImpl` view command to render JSP files directly. For example, in the diagram illustrating the URLs used in Express Store, in order to display the Help page (`HelpDisplay.jsp`), the `HelpView` is registered in the view registry and associated with the `HelpDisplay.jsp` and the `HttpForwardViewCommandImpl` command. This is demonstrated in the following example:

```
<viewreg
viewname="HelpView"
devicefmt_id="-1"
storeent_id="@storeent_id_1"
interfacename="com.ibm.commerce.command.ForwardViewCommand"
classname="com.ibm.commerce.command.HttpForwardViewCommandImpl"
properties="docname=HelpDisplay.jsp"
internal="0"
https="0"
/>
```

Note that the the fully qualified classname for the interface and the implementation class is used.

Use the `HttpForwardViewCommandImpl` view command to render views returned from a display command. A display command reads data from the database, but does not change it. For example, in the diagram illustrating the URLs used in Express Store, the `OrderItemDisplay` command returns the `OrderItemDisplayViewShiptoAssoc` view. When this view was registered in the view registry, the `OrderItemDisplay.jsp` and the `HttpForwardViewCommandImpl` were associated with it. This is demonstrated in the following example:

```
<viewreg
viewname="OrderItemDisplayViewShiptoAssoc"
devicefmt_id="-1"
storeent_id="@storeent_id_1"
interfacename="com.ibm.commerce.command.ForwardViewCommand"
classname="com.ibm.commerce.command.HttpForwardViewCommandImpl"
properties="docname=OrderItemDisplay.jsp"
internal="0"
https="0"
/>
```

You must associate a JSP filename for every view associated with every display command (for example, `OrderItemDisplay`) you use.

Note: `ProductDisplay` and `CategoryDisplay` list the associated JSP filename in the catalog data rather than the view registry.

HttpRedirectViewCommandImpl

This view command redirects the view request to another URL.

Use the `HttpRedirectViewCommandImpl` view command to render views returned from a command that changes the database. To use the redirect view, specify the view name using the `&URL=` parameter on the URL. For example, when you add address information in the Express Store `AddressForm` and click **Submit**, it invokes the `AddressAdd` command. The URL used to invoke the `AddressAdd` command specifies `AddressBookForm` as the view using the `&URL=` parameter. This results in a redirect to the `AddressBookForm` view. When the `AddressBookForm` view was registered in the view registry, the `AddressBookForm.jsp` and the `HttpForwardViewCommandImpl` were associated with it.

You must use the `URL=parameter` technique for all non-display commands. Non-display commands are commands that cause changes to the data in the database.

HttpDirectViewCommandImpl

This type of view command sends the response view directly to the client. It does not call a JSP file. Direct views allow controller commands to produce the output response (rather than the view command).

Set access control for a page

When the Express Store was published access control policies for the store were loaded to the database. Refer to the file `accesscontrol.xml`, for an example of how to create an access control file.

1. To add a new view that is accessible by roles with existing role-based View policies, create an XML file similar to the one shown:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE Policies SYSTEM "../dtd/accesscontrolpolicies.dtd">
<Policies>
```

```
</Policies>
```

2. Create a new action definition in the XML file for the view that you added, for example,

```
<Action Name="AddToExistReqListView" CommandName="AddToExistReqListView"/>
```

3. Determine which roles should have access to this view, and associate the new action with the corresponding action groups in the XML file as in the following example:

```
<ActionGroup Name="ExpressAllUsersViews" OwnerID="RootOrganization">
  <ActionGroupAction Name="AddToExistReqListView"/>
</ActionGroup>
```

There is already a role-based policy, `ExpressAllUsersViews`, that includes this action group, so a new policy does not have to be created. All shopping commands in the Express Store belong to the `ExpressAllUsersViews` action group.

4. Load your XML changes into the database.
 - a. Copy the XML file you created to the `WC_installdir/xml/policies/xml`.
 - b. Login as the database user ID.

► **Linux** The database user ID must have the following permission in order to proceed with the following steps:

- read/write/execute authority to the directories, subdirectories, and files of `WC_installdir/xml/policies` and `WC_installdir/logs`.
- read/execute authority to the `WC_installdir/bin` directory and its files.

If the database user ID does not have the above required authority, you need to grant this authority using the `chmod` command.

- c. From the directory `WC_installdir/bin`, run the following command:

```
acpload database name database user database user password
Policies xml filename[schema name]
```

For example: `acpload mall dbuser dbusrpwd defaultAccessControlPolicies.xml`

For ► **400** : if you create customized XML files, you must use the full path to the DTD in your file. The access control policies DTDs are located in `WC_installdir/xml/policies/dtd`.

- d. Check the log files `acugload.log`, `acpload.log`, and `acpnload.log` in `WC_installdir/logs` for any errors. For ► **400** the log files are located in `WC_userdir/instances`.
5. Update the Access Control Policies Registry in the Administration Console by doing the following:
 - a. Open the Administration Console:

► **Developer**

https://host_name/webapp/wcs/admin/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogon

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.



`https://host_name:8002/adminconsole`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server.

- b. Click **Configuration > Registry**.
- c. From the list of registries, select **Access Control Policies**.
- d. Click **Update**.

Caching your store pages

While developing your storefront you also need to determine how to cache your store pages. This section discusses creating and implementing a caching strategy for your store.

Stale JSP files display

You will need to set a JSP reload interval in the WebSphere Commerce Server environment. If you do not set the JSP reload interval, then you will not see changes to JSP pages when a page is reloaded. Changes to JSP pages will be visible after 30 minutes, or when the server is restarted.

1. Open the file `ibm-web-ext.xmi` in an editor. The file is located in the following directory:
`WAS_installdir/config/cells/hostname/applications/WC_instancename.ear/deployments/WC_instancename/Stores.war/WEB-INF`
2. Locate the `webappext` element. This element is near the beginning of the file.
3. Set the `reloadingEnabled` attribute to `true`. If the attribute does not exist, add it.
4. Set the `reloadingInterval` attribute value to `0`. If the attribute does not exist, add it.
5. Save the file.
6. Restart the server.

Planning your caching strategy

Consider the following issues:

- Which pages should be cached?

Pages that are good candidates for caching are pages that are accessed frequently, but are also stable for a period of time, and contain content that can be reused by a variety of users. For example, catalog display pages are usually good candidates to enable caching.

- Should pages be cached as whole pages, or as page fragments?

WebSphere Commerce uses the WebSphere Application Server dynamic cache service. This allows WebSphere Commerce to support caching entire pages, and caching page fragments. Your store pages can be cached using whole page caching or fragment caching or a combination of the two methods.

Caching a Web page as a whole simply caches the entire page as one entity, even if it is composed of several smaller fragments. Page fragments may include a

separate header, sidebar, or footer. Even the main body of the page may be broken into several fragments. For example one fragment on the main body page of the page may show a product, while a second fragment shows the price. Fragmenting pages allows you to show personalized content for individual users. The sample store pages provided with WebSphere Commerce are composed of several fragments (header, sidebar, footer, main content).

Caching individual fragments allows you to cache the portions of the page that are reusable for a wider audience. If a page contains personalized information for only a small segment of your audience, caching this page as a whole page will not allow the page to be reused very often. Only a small audience segment can ever reuse that cached page. For example, a page displays customer-specific welcome message in the header. The page is cached based on the user ID; therefore, only that customer can ever reuse that cached page. However, if you decompose that page into fragments, you can cache the fragments that get reused for most of your audience. For example, the footer, sidebar, and product display fragments may be applicable to all your users, while the price and header fragments may be personalized. When the page is requested, the individual fragments are reassembled to produce the page.

- How is the page or page fragment requested?

How the JSP file (whether it is a single page, or a page fragment) is requested determines how the WebSphere Application Server will cache it. For example, the WebSphere Application Server needs to know whether the JSP file is displayed as a response to a servlet, object, EJB, or a command. Compile a list of how each page or fragment you plan to cache is requested.

- How is the cached data invalidated?

For each page, or page fragment that you plan to cache, determine when the cached page, or fragment, is no longer valid. Then remove the corresponding cache entries from the cache. This process is known as invalidation. To determine when a cached page has changed, and therefore is no longer valid, determine what could make the cached page out-of-date. For example a cached shopping cart page is invalid when a customer adds a new item to the cart. Cached pages may also be invalidated when an administrator updates the store with the WebSphere Commerce Accelerator, or when new catalog data is added with the Loader package.

Compile a list of all possible ways the cached page or fragment can be invalidated; then determine which events cause the invalidation. Events that cause invalidation can include a servlet request, a controller command or a task command and so on. For example, the product management tools in the WebSphere Commerce Accelerator invokes the commands `AddCatalogEntryDescCmd` or `UpdateCatalogEntryDescCmd` to update catalog entry descriptions. To invalidate cached pages that are changed by these commands, you need to add invalidation policies to the `cachespec.xml` file that intercept the execution of the commands, and trigger the invalidation. For information on implementing invalidation, see the following:

- The *WebSphere Commerce Administration Guide*, "Dynamic caching" chapter for instructions on setting up new invalidation policies, and an example of cache invalidation.
- "Invalidating cached data in the `cachespec.xml` file" on page 76 for instructions on how to merge the sample invalidation policies provided by WebSphere Commerce with your store's `cachespec.xml` file.

Implementing your caching strategy

To implement your caching strategy create a cache policy file describing what is to be cached and how, and how cached pages will be invalidated. The WebSphere Application Server dynamic cache service uses this cache policy file, known as `cachespec.xml` to implement caching in your store.

Each sample store provided with WebSphere Commerce includes a `cachespec.xml` file that defines the caching strategy for that store. These files are located in the following directory:

`WC_installdir/samples/dynacache/Express`

You can change these files if your store is based on a sample, or use these files as a base for the `cachespec.xml` file for your store.

Understanding the `cachespec.xml` file

In order to cache store pages, you must define objects that can be cached in the `cachespec.xml` file. WebSphere Commerce only uses a subset of the elements defined in the `cachespec.xml` file. This subset of elements is explained in this section. For more detailed information about the `cachespec.xml` file, see the WebSphere Application Server Information Center (<http://www.ibm.com/software/webservers/appserv/infocenter.html>), topic "Cachespec.xml file". For more information, see the *WebSphere Commerce Administration Guide*, "Dynamic caching" chapter.

WebSphere Commerce uses the following elements in the `cachespec.xml`:

```
<cache-entry>
  <class>servlet</class>
  <name>/FashionFlow/ShoppingArea/CatalogSection/CategorySubsection
    /StoreCatalogDisplay.jsp</name>
  <property name="save-attributes">false</property>
```

class The class element is a required element. It determines how the WebSphere Application Server will interpret the remaining cache policy definition. WebSphere Commerce uses the following class values:

command

The value `command` refers to classes that use the WebSphere Commerce programming model.

servlet

The value `servlet` refers to servlets or JSP files deployed in the WebSphere Application Server servlet engine. For WebSphere Commerce version 5.5, only `command` invalidation is supported.

name Name is a required element. Name is the fully qualified class name (including the package name), of the servlet or command.

```
com.ibm.commerce.dynacache.commands.MemberGroupsCacheCmdImpl
com.ibm.commerce.server.RequestServlet.class
/ToolTech/ShoppingArea/CatalogSection/CategorySubsection/
StoreCatalogDisplay.jsp
```

property

The property element contains one attribute, **name**. The following list shows the possible values for the name attribute.

name attribute values

EdgeCacheable

Value True or False. Default is false.

Valid classes

Servlet

Description

If the property is true, then the given servlet or JSP file is externally requested from an Edge Server. Whether the servlet or JSP file can be cached depends on the rest of the cache specification.

```
<property name="EdgeCacheable">>false</property>
```

consume-subfragments

Value True or False. Default is false

Valid classes

Servlet

Description

When a servlet is cached only the content of that servlet is stored. Placeholders for any other fragments to which it includes or forwards are created. Consume-subfragments (CSF) tells the cache to continue saving content when it encounters a child servlet via an include. The parent entry (the one marked CSF) will include all the content from all fragments in its cache entry, which result in one big cache entry that has no includes or forwards, but the content from the whole tree of entries. This method can save a significant amount of application server processing, but is typically only useful when the external HTTP request contains all the information needed to determine the entire tree of included fragments.

```
<property name="consume-subfragments">>true</property>
```

save-attributes

Value True or False. Default is true.

Valid classes

Servlet

Description

When save-attributes is set to false, the request attributes are not saved with the cache entry.

The following property name and value is mandatory for all cache entries defined in JSP files:

```
<property name="save-attributes">>false</property>
```

store-cookies

Value True or False. Default is True.

Valid classes

Servlet

Description

When store-cookies is set to false, the request cookies are not saved with the cache entry.

By default, DynaCache caches the cookies (when caching by servlet class) and all request attributes (servlets and JSPs) along with the cache entries. However, WebSphere Commerce cookies and request

attributes contain user-specific information that should not be cached. As a result, the following property names and values are mandatory when caching full pages:

```
<property name="store-cookies">false</property>
```

Understanding cache-ID rules: A cache-ID uniquely identifies a cache entry. In order for the WebSphere Application Server to cache an object, it must know how to generate a unique ID for different invocation of that object. These IDs are created from either user-written custom Java code or from rules defined in a cache entry's cache policy.

In the `cachespec.xml` file the cache-ID element defines the rules for generating IDs. Each cache entry may have multiple cache-ID rules. These rules execute in the defined order until either a rule returns a non-empty cache ID, or no more rules are left to execute. If none of the cache-ID generation rules produce a valid cache ID, then the object is not cached.

These IDs are developed in one of the following ways:

- Using the component elements defined in the cache policy of a cache entry
- Writing custom Java code to build the ID from input variables and system state

Understanding dependency-ID rules: Dependency ID elements specify additional cache group identifiers that associate multiple cache entries to the same group identifier. The dependency ID is generated by concatenating the dependency ID base string with the values returned by its component elements. If a required component returns a null value, then the entire dependency ID is not generated and is not used.

You can validate the dependency IDs explicitly through the WebSphere Dynamic Cache API, or by using another cache-entry invalidation element. Multiple dependency ID rules can exist per cache-entry element. All dependency ID rules execute separately. For more information on how to define dependency ID rules, see the *WebSphere Commerce Administration Guide*, "Dynamic caching" chapter.

Understanding invalidation rules: Invalidation rules can be defined in exactly the same manner as dependency IDs. However, the IDs that are generated by invalidation rules are used to invalidate cache entries that have the same dependency IDs. The invalidation ID is generated by concatenating the invalidation ID base string with the values returned by its component element. If a required component returns a null value, then the entire invalidation ID is not generated and no invalidation occurs. Multiple invalidation rules can exist per cache-entry element. All invalidation rules execute separately. For more information on how to define invalidation rules, see the *WebSphere Commerce Administration Guide*.

Invalidating cached data in the cachespec.xml file

By default, the `cachespec.xml` files shipped with the sample store archives do not include invalidation policies. To automate cache invalidation using DynaCache in a sample store, or a store based on a sample; add invalidation policies to the store's `cachespec.xml` file. Sample invalidation policies are provided in several `cachespec.xml` files in the following directory:

```
WC_installdir/samples/dynacache/invalidation
```

This directory contains separate `cachespec.xml` files for functional areas, including catalog, shopping cart, store and so on. Each file contains invalidation policies for that specific area.

If you plan to cache catalog pages in your store, you should add the invalidation policies from the following files into your store:

- `WC_installdir/samples/dynacache/invalidation/catalog/cachespec.xml`
- `WC_installdir/samples/dynacache/invalidation/membergroup/cachespec.xml`

Note: For member group invalidation rules you need to add additional dependency IDs to the cache-entry element. See the content of the `cachespec.xml` file for more details.

- `WC_installdir/samples/dynacache/invalidation/store/cachespec.xml`

Adding sample invalidation policies to your store's `cachespec.xml` file: In order to add the invalidation policies provided in the sample invalidation files into your store, do the following:

1. Open the `cachespec.xml` file for your store. If your store is based on a sample provided with WebSphere Commerce, the `cachespec.xml` file is located in the following directory:
 - `WC_installdir/samples/dynacache/Express`
2. Open sample invalidation `cachespec.xml` file. The sample invalidation `cachespec.xml` files are located in the following directory:
 - `WC_installdir/samples/dynacache/invalidation`
3. Copy the invalidation policies from the sample invalidation file to the `cachespec.xml` file for your store. You can place the invalidation policies at the end of your store's `cachespec.xml` file, before the last element.
4. Ensure the invalidation IDs match the corresponding dependency IDs in the caching policies. Change either the ID of the invalidation ID or the ID of the dependency ID so they match. If a matching dependency ID does not exist, the invalidation policies will not be executed.

Note: Your store may have additional or different business requirements that require you to add additional invalidation policies and dependency IDs.

5. Change the names and directories of the JSP files in the sections copied from the sample invalidation files to match your store's information.
6. Save the file.

Chapter 13. Changing store data

Shipment tracking URL

The Express Store contains a shipment tracking URL on the order status page. This field allows customers to track the status of each order item from purchase to delivery. To use the shipment tracking URL every order item needs to create a manifest with a trackingID. You will need to customize WebSphere Commerce to add information to the ORDRELEASE, MANIFEST and ORDERITEM tables. This information may come from a back-end system as part of integration. These instructions outline the information that needs to be added to the database for every order item.

1. Set up a shipment tracking URL for every Shipping Provider, as described in the section "Setting up shipping".
2. For every order item enter the following information in the database:

- a. insert into ordrelease (orders_id,ordreleasenum,status)
values(11001 ,10001,'MNF')

where:

orders_id

ORDERS_ID from ORDERS table for which the manifest is performed.

ordreleasenum

A unique identifier for the release. The ordreleasenum must be an integer.

status

Use the status MNF. This indicates that the release shipment confirmation was received.

- b. insert into manifest (DATESHIPPED, MANIFEST_ID, ORDERS_ID, ORDRELEASENUM, shippingcosts,trackingid)
values('2003-07-21-14.47.17.797000',10001,11001,10001, 5.3,'trackingID')

dateshipped

The date and time that the package was shipped. This value must be in the format shown YYYY-MM-DD-HH.MM.SS.mmmmm.

manifest_id

A generated unique ID

orders_id

ORDERS_ID from ORDERS table for which the manifest is performed.

ordereleasenum

Uniquely identifies the release for this order.

shippingcosts

A decimal value indicating the shipping costs

trackingid

The tracking id for the shipping provider as entered on the Shipping notebook.

- c. update ORDERITEMS set ORDRELEASENUM=10001 where
CATENTRY_ID=10000

where the CATENTRY_ID is the ID of the item in the order.

Changing exchange rates

When the Express Store was published exchange rates were provided for all supported currencies. To change the exchange rates follow these steps:

1. Create an XML file, for example currency.xml, that defines the currency conversion rates. The file should look like this:

```
<?xml version="1.0"?>
<!DOCTYPE import SYSTEM "store-catalog-assets.dtd">
<import>
<curconvert storent_id="store_id" fromcurr="USD" tocurr="EUR"
  factor="2.09804" multiplyordivide="M" bidirectional="Y"
  updateable="Y" curconvert_id="@curconvert_id_1" />
<curconvert storent_id="store_id" fromcurr="USD" tocurr="JPY"
  factor="211.14" multiplyordivide="M" bidirectional="Y"
  updateable="Y" curconvert_id="@curconvert_id_2" />
<curconvert storent_id="store_id" fromcurr="USD" tocurr="KRW"
  factor="2201.00" multiplyordivide="M" bidirectional="Y"
  updateable="Y" curconvert_id="@curconvert_id_3" />
<curconvert storent_id="store_id" fromcurr="USD" tocurr="BRL"
  factor="2.96" multiplyordivide="M" bidirectional="Y"
  updateable="Y" curconvert_id="@curconvert_id_4" />
<curconvert storent_id="store_id" fromcurr="USD" tocurr="CNY"
  factor="9.28" multiplyordivide="M" bidirectional="Y"
  updateable="Y" curconvert_id="@curconvert_id_5" />
<curconvert storent_id="store_id" fromcurr="USD" tocurr="TWD"
  factor="37.98" multiplyordivide="M" bidirectional="Y"
  updateable="Y" curconvert_id="@curconvert_id_6" />
</import>
```

where:

store_id

The store_id. (To determine the store_id use the following SQL statement select * from storent)

fromcurr

The 3 character ISO 4217 currency code representing the converted currency. This code must appear in the SETCCURR column of the SETCURRE table.

tocurr

The 3 character ISO 4217 currency code representing the converted currency. This code must appear in the SETCCURR column of the SETCURRE table. factor is the conversion rate.

multiplyordivide

Whether to multiply ('M') or divide ('D'). bidirectional indicates whether the conversion is bi-directional ('Y') or unidirectional ('N'). updateable whether the conversion rate can be changed ('Y') or not ('N').

curconvert_id

An entity representing the currency conversion unique identifier. The entity will be resolved when the ID resolve command is run.

2. Save the file to the directory,
WCDE_installdir\samples\stores\ExpressStore\B2CCatalog\.
3. Ensure that the environment variables are set in the *WCDE_installdir*/bin/setenv.bat file, if you have loaded the sample catalog already then these variables should be set correctly. If not follow the directions to set the variables in the section 3 on page 29.

4. Run the ID Resolve command to resolve entities.

- a. Open a DB2 Command window.
- b. Change to the directory: *WCDE_installdir\bin*
- c. Run the ID Resolve command:

```
idresgen -dbname demo_dev -dbuser dbuser -dbpwd dbpwd  
-infile "infile" -outfile "outfile" -method mixed -schemaname schemaname
```

dbname

The name of the database.

dbuser The name of the user connecting to the database.

dbpwd The password for dbuser.

schemaname

The name of the schema.

infile The name, including the full path, of the input file

outfile The name, including the full path, of the output file.

d. Run the Load command

```
massload -dbname demo_dev -dbuser dbuser -dbpwd dbpwd  
-infile "infile" -method sqlimport -schemaname schemaname
```

dbname

The name of the database.

dbuser The name of the user connecting to the database.

dbpwd The password for dbuser.

schemaname

The name of the schema.

infile The name, including the full path, of the input file.

outfile The name, including the full path, of the output file.

5. Refresh the Currency Manager.

a. Open the Administration Console:

▶ Developer

https://host_name/webapp/wcs/admin/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogon

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

▶ 400

▶ Linux

▶ Windows

https://host_name:8002/adminconsole

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server.

- b. Click **Configuration > Registry**.
- c. From the list of registries, select **All**.
- d. Click **Update**.

Chapter 14. Managing your payments

There are two ways that you can manage your payments with WebSphere Commerce - Express: using WebSphere Commerce Payments, or manually.

With Express Store, you can choose to publish your store with or without WebSphere Commerce Payments, and your store will still be fully functional. In other words, even if you do not install or enable WebSphere Commerce Payments, you will be able to complete your orders.

Follow the appropriate directions:

- Processing orders with WebSphere Commerce Payments enabled
- Processing orders without enabling WebSphere Commerce Payments

Processing orders with WebSphere Commerce Payments enabled

WebSphere Commerce - Express provides an optional component called WebSphere Commerce Payments (formerly known as Payment Manager). WebSphere Commerce Payments, or Payments, is a protocol-independent payment transaction server for online merchants. It provides cash register-like functionality to a site, supporting multiple payment methods using protocol-specific cassettes. These cassettes are software components that can be attached to the Payments framework to interpret generic payment and administrative commands into payment protocol-specific requests, which are then forwarded to the appropriate recipient, such as the payment gateway of an Acquirer institution. The end result is similar to when a cashier swipes a payment card at the checkout counter in a traditional store.

Payments handles all the background details of Internet payments for the merchant (or a group of merchants, such as a store group), and provides a graphical interface to simplify transaction management. Payments performs the following functions:

- It verifies, upon startup and dependent on the cassettes, that all required Acquirer certificates are present. If a certificate is missing, it is requested from the Acquirer or Payment Gateway automatically.
- It runs permanently, listening on dedicated ports and serving all incoming requests from the wallet and the merchant server.
- It communicates with the Acquirer over the Internet using the TCP/IP protocol or other protocols depending on the cassette.
- It uses database tables to maintain information on the status of transactions, approval requests, and deposit requests. The records in these tables are kept for working purposes and for tracking and record-keeping. The database also contains configuration tables that retain information about the merchant, the brands of payment cards, and the Acquirers that provide card authorization and payment capture services to the merchant.

The merchant server interacts with Payments by using APIs. Payments APIs are general enough to support other payment technologies.

WebSphere Commerce - Express provides the following WebSphere Commerce Payments cassettes.

CustomOffline

Supports processing of custom payment transactions such as cash on delivery or COD, Bill-Me-Later, or coupons that are often executed outside of WebSphere Commerce Payments.

BankServACH

Supports processing of online electronic check payments using the BankServ payment gateway that interfaces with the Automated Clearing House Network (ACH).

Paymentech


Supports online authorization and settlement of credit card and non-PIN based debit card payments.

VisaNet

Supports processing of credit card transactions using the Vital Processing Services or First Horizon Merchant Services (FHMS) financial network including support for a purchasing card.

Installing and enabling WebSphere Commerce Payments

If you are using WebSphere Commerce Payments, do the following:

1.  If you have not already installed WebSphere Commerce Payments, install it now. Installation directions are provided in the *Installation Guide* for your platform. Your payments instance must be set up properly in order to connect the payment server to WebSphere Commerce.
2. Start WebSphere Commerce Payments.

Developer

- a. Select **Start > Programs > IBM WebSphere Commerce Studio > WebSphere Commerce development environment**.
- b. In the **Servers** view, right-click **WebSphereCommercePaymentsServer** and select **Start** from the pop-up menu.

The server started successfully if there are no errors in the Console view and the following messages appears:

```
Application Started: WebSphereCommercePaymentsServer
. . .
Server server1 open for e-business
```

400 Linux Windows

- a. Open the Configuration Manager.
 - b. Expand your host name.
 - c. Expand **Payments**.
 - d. Expand **Instance List**.
 - e. Select **Action > Start Payments Instance**.
 - f. After receiving the Instance started successfully dialog, click **OK** to dismiss the dialog and confirm that WebSphere Commerce Payments has started.
3. Enable Payments using the Change Flow feature in the WebSphere Commerce Accelerator.
 - a. Open the WebSphere Commerce Accelerator.

Developer

https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=common.mcLogon

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.



`https://host_name:8000/accelerator`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server machine.

- b. From the **Store** menu, select **Change Flow**. Then select **Payments**.
- c. Select **Enable WebSphere Commerce Payments**, and click **OK**.
4. To enable the cassettes in the WebSphere Commerce Accelerator, using the Configuration Manager:
 - a. Open the Configuration Manager.
 - b. Expand the hostname.
 - c. Expand **Payments**.
 - d. Expand **Instance List**.
 - e. Expand **instance_name**.
 - f. Expand **Instance Properties**.
 - g. Click **Cassettes**.
 - h. For each cassette you wish to enable, select it, and click >
 - i. Click **Apply**.

You need to enable these cassettes for the merchant that corresponds to the store, and then create the accounts. This will cause them to display in the WebSphere Commerce Accelerator.

5. Configure WebSphere Commerce Payments.
 - a. Access the Payments user interface from a browser by typing the URL:
`http://host_name:port/webapp/PaymentManager`

where:

host_name The domain-qualified host name of the machine where Payments is installed.

port The SSL port used by WebSphere Commerce Payments. The default SSL port is 5433.

- b. Type your Payments user ID. Type your Payments password. Click **Logon**.
- c. If one was not created when you published the store, add a merchant. From the **Merchant Settings** menu, select **Add a merchant**. Provide the appropriate information, making sure that the Merchant Number you enter is identical to the Store ID number of the store that you are connecting to Payments.
- d. Return to **Merchant Settings**, and click on the cassette icon next to the merchant. Select an appropriate cassette.

Adding accounts

- e. If one was not created when you published the store, add an account. From **Accounts**, select **Add an account**, and provide the appropriate information for the account (for example, currency, account number.) For more details on creating the accounts, refer to the appropriate cassette guide listed below. For the BankServ cassette, you should create two accounts: one with the value of BillMe and another one with COD. Once you complete this step for each of the accounts, the payment methods should be available in the WebSphere Commerce Accelerator without having to do any changes to the JSP files.

- f. Repeat step 5e on page 85 for every currency you would like to support; each currency requires its own account.
- g. Click on the link for the account name you created.

Adding brands (for offline card cassette only)

- h. Click **Brands** on the next page.
- i. Select **Add a brand**, type a brand name and click **Create brand**.
- j. Repeat step 5i for every brand you would like to support for that account (for example, Visa, MasterCard).
- k. Log off WebSphere Commerce Payments.

- 6. Modify the store page on which customers submit orders, OrderSubmitForm.jsp. This step is required if you are using any of the following cassettes: BankServACH, Paymentech, or VisaNet. If you are using the OfflineCard or CustomOffline cassette; then you do not need to perform this step.

- a. Open the following file, OrderSubmitForm.jsp:

```

Linux Windows WAS_installdir/installedApps/host_name/
WC_instance_name.ear/Stores.war/storedir/ShoppingArea/CheckoutSection
/StandardCheckoutSubsection

```

```

400 WAS_userdir/WAS_instance_name/
installedApps/host_name/WC_instance_name.ear/Stores.war/storedir
/ShoppingArea/CheckoutSection/StandardCheckoutSubsection

```

```

Developer workspace_dir/installedApps/host_name/
WC_instance_name.ear/Stores.war/storedir/ShoppingArea/CheckoutSection
/StandardCheckoutSubsection

```

- b. Search for the following text:

```
if (info[i].getPolicyName().trim().equals("OfflineCard"))
```

Change the name of the payment policy from OfflineCard to one of the following as appropriate:

```

BankServACH
Paymentech
VisaNet
VisaNet_PCard

```

Note: If you are using the Cassette for VisaNet with purchasing card support, select the VisaNet_PCard profile rather than VisaNet.

- c. If the store supports Quick Checkout repeat step 6b in the following files:

```

ShoppingArea/CheckoutSection/QuickCheckoutSubsection/QuickCheckoutForm.jsp
UserArea/AccountSection/QuickCheckoutProfileSubsection/QuickCheckoutProfileForm.jsp

```

- d. For enabling BankServ Cassette, copy the StandardBankServACH.jsp file from:

```

Linux Windows WAS_installdir/installedApps/host_name
/WC_instance_name.ear/CommerceAccelerator.war/tools/
order/buyPages

```

```

400 WAS_userdir/WAS_instance_name/installedApps/host_name
/WC_instance_name.ear/CommerceAccelerator.war/tools
/order/buyPages

```

```

Developer workspace_dir/installedApps/host_name/WC_instance_name.ear
/CommerceAccelerator.war/tools/order/buyPages

```

to:

```

+ ▶ Linux ▶ Windows WAS_installdir/installedApps/host_name
+ /WC_instance_name.ear/Stores.war/storedir/ShoppingArea
+ /CheckoutSection/StandardCheckoutSubsection
+
+ ▶ 400 WAS_userdir/WAS_instance_name/installedApps/host_name
+ /WC_instance_name.ear/Stores.war/storedir/ShoppingArea
+ /CheckoutSection/StandardCheckoutSubsection
+
+ ▶ Developer workspace_dir/installedApps/host_name/WC_instance_name.ear
+ /Stores.war/storedir/ShoppingArea/CheckoutSection
+ /StandardCheckoutSubsection

```

Edit this file so that any Java code using the resource bundle is eliminated. For example, replace all the `(String)resourceBundle.get("xxxxxx")` calls with hard coded strings, such as `"XXXXXXXXXX"`.

- e. If using VisaNet Cassette or Paymentech Cassettes, modify the page used to collect credit card information from customers, `StandardCreditCard.jsp`:
`ShoppingArea/CheckoutSection/StandardCheckout/StandardCreditCard.jsp`

VisaNet Cassette:

```
<option value="VisaNet">VisaNet</option>
```

or

Paymentech Cassette

```
<option value="Paymentech"></option>
```

Modify only the line of the cassette you want to use for the store. This line is added after the following line:

```
<select name="cardBrand">
```

Refer to the appropriate cassette's documentation for further information.

7. Review, and if necessary, modify the Cashier profile to set certain parameters, such as the APPROVEFLAG and DEPOSITFLAG parameters. Refer to the appropriate cassette supplement documentation:
 - *WebSphere Commerce Payments Cassette for VisaNet Supplement*
 - *WebSphere Commerce Payments Cassette for Paymentech Supplement*
 - *WebSphere Commerce Payments Cassette for BankServACH Supplement*
 - *WebSphere Commerce Payments CustomOffline Cassette Supplement*
 - *WebSphere Commerce Payments OfflineCard Cassette Supplement*

All of these publications are available from the WebSphere Commerce Technical Library page at:

<http://www.ibm.com/software/commerce/library/>

8. ▶ 400 ▶ Linux ▶ Windows Compile all of the WebSphere Commerce Payments JSP files for your site. Run the following command:

```

JspBatchCompiler.bat -enterpriseapp.name "instance_name_Commerce_Payments_App"
-cell.name "hostname" -node.name "hostname"
-server.name "instance_name_Commerce_Payments_Server"

```

where `instance_name` is the name of the WebSphere Commerce Payments instance for which you are compiling the JSP files.

9. Schedule the PayCleanup job.

The PayCleanup job cancels WebSphere Commerce orders with payment authorization requests rejected by the respective financial institutions for longer than a store configured period of time.

- a. Open the Administration Console:

▶ Developer

`https://host_name/webapp/wcs/admin/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogon`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

▶ 400 ▶ Linux ▶ Windows

`https://host_name:8002/adminconsole`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server.

- b. From the **Configuration** menu, click **Scheduler**. A list of jobs that are scheduled to run on your store displays.
- c. Click **New**. The Schedule Job window opens.
- d. From the **Job command** drop-down list, select **PayCleanup**.
- e. Complete the fields as follows:
 - In the **Start time** field enter 00:00.
 - In the **Interval** field enter 86400.
 - In the **Priority** field enter 1.
- f. Click **OK**.

Processing orders without enabling WebSphere Commerce Payments

If you do not want to enable WebSphere Commerce Payments, follow these steps:

1. Verify that information in the ORDPAYINFO and ORDPAYMTHD database tables is encrypted. To encrypt information, the PDI encrypt field must be checked during instance creation. This is the default value.
2. After the order is placed, go to the WebSphere Commerce Accelerator, and retrieve the order. The decrypted credit card number is shown on the Order Summary page.
3. Process the order using your own payment mechanism (offline payment).

Encrypting and decrypting credit card information

Even if the store is not connected to a payment manager, the credit card information can still be taken along with the order when the order is submitted for process. The credit card information will be validated with the Mod10 checking algorithm, and then the valid credit card information will be stored for the order, or the order will not be submitted for process.

1. Ensure that encryption is turned on, this is the default value. If you changed the value of the PDI encrypt field when you set up WebSphere Commerce follow these steps to enable encryption:
 - a. Launch the Configuration Manager and traverse to the PDI Encrypt node for your instance as follows: WebSphere Commerce > *host_name* > Instance List > **instance_name** > Instance Properties
 - b. Check the PDI encrypt check box.

- c. To apply your changes to Configuration Manager, click **Apply**.
 - d. Upon successfully updating the configuration for your instance, you will receive a message indicating a successful update.
 - e. From the WebSphere Application Server Administration Console, stop then restart the WebSphere Commerce server instance.
2. Open the following file:

```
Developer WCDE_workspacedir\CommerceAccelerator\Web
Content\tools\order\OrderSummaryDetailsB2C.jsp
```

3. Search for the section shown in bold.

```
<TR>
  <%
    String creditCardNumber = getPaymentCreditCardNumber(orderId);
    if (!creditCardNumber.equals("")) {
      StringBuffer displayCreditCardNumber = new StringBuffer();

      for (int i=0;i<creditCardNumber.length()-4;i++){
        if (creditCardNumber.charAt(i) != '-' && creditCardNumber.charAt(i) != ' '){
          displayCreditCardNumber.append("*");
        }
      displayCreditCardNumber.append(creditCardNumber.substring(creditCardNumber.length()-4));
    }
    <TD ALIGN=LEFT>
      <%= UIUtil.toHTML((String)orderLabels.get("orderSummaryDetCreditCardNumber")) %><%=
      UIUtil.toHTML((String)orderLabels.get("orderSummaryDetLabelTextSeparator")) %></TD>
    <TD><1><%= UIUtil.toHTML(displayCreditCardNumber
      .toString()) %></TD>
  <%
  }
  <%>
</TR>
</TR>
```

4. Comment and add a line as follows:

```
//for (int i=0; i<creditCardNumber.length()-4; i++) {
// if (creditCardNumber.charAt(i) != '-' && creditCardNumber.charAt(i) != ' ') {
// displayCreditCardNumber.append("*");
// }
//}
//displayCreditCardNumber.append(creditCardNumber.substring(creditCardNumber.length()-4));
displayCreditCardNumber.append(creditCardNumber);
```

Once this is done, you will be able to see the credit card number for the order on the Order Summary page in the WebSphere Commerce Accelerator. Any user who has authority to view the Order Summary page in the WebSphere Commerce Accelerator will be able to see the credit card information. This may have security implications.

Chapter 15. Using e-Marketing Spots

e-Marketing Spots reserve space on your store pages in which personalized marketing content for campaign initiatives displays. When a customer requests a page, any e-Marketing Spots on the page communicate with the rule server. The rule server processes the rule-based code associated with the spot and returns the content in the reserved space. Each e-Marketing Spot has one or more campaign initiatives associated with it. e-Marketing Spots use the e-MarketingSpot bean to return the results of currently scheduled campaign initiatives. You can customize your e-Marketing Spot and the corresponding campaign initiative by using different properties of the e-MarketingSpot bean. For more information about how e-Marketing Spots are used in campaigns refer to Chapter 19, "Marketing," on page 113.

The e-Marketing Spot code provided with the Express Store sample store supports two types of campaign initiatives:

- Suggestive selling
 - Product recommendation
 - Category recommendation
 - Cross-sell
 - Up-sell
- Awareness advertisement
 - General advertisement
 - Advertise discount

If your e-Marketing Spot is specific to a single subset of campaign initiative, for example, cross-sell; then modify the sample code to remove all code except the code required for the cross-sell initiative.

Adding an e-Marketing Spot to a store page

The Express Store contains an e-Marketing Spot on the Shopping cart page, `ShoppingArea/ShopcartSection/OrderItemDisplay.jsp`. The e-Marketing Spot code is in the file `include/emsContent.jsp`. This e-Marketing Spot shows a product recommendation.

To add an e-Marketing Spot to a store page, do the following:

1. Determine on which JSP files the e-Marketing Spot will display. You can add the same e-Marketing Spot to multiple JSP files; and a single JSP page may contain more than one e-Marketing Spot.
2. Determine where in the JSP file to place the e-Marketing Spot.
3. Add appropriate HTML tagging to display the content of the e-Marketing Spot. The following example shows an e-Marketing Spot added to the Shopping cart page in the Express Store sample store.

```
<tr>
  <!-- Display the heading for the e-Marketing spot-->
  <td width="10">&nbsp;</td>
  <td align="left" colspan="5" valign="middle" width="580">
    <font class="textCustomColor"><%=infashiontext.getString("SUGGESTION")%></font></td>
</tr>
<tr>
  <td width="10">&nbsp;</td>
```

```

<td align="left" colspan="5" valign="middle" class="categoryspace">
  <table cellpadding="5" border="0">
    <tbody>
      <!-- Add e-Marketing Spot to the page -->
    </tbody>
  </table>
</td>
</tr>

```

4. Add the e-Marketing Spot to the JSP file by using the Java dynamic include tag, `<jsp:include>` tag, for example:

```

<jsp:include page="../../include/emsContent.jsp" flush="true">
  <jsp:param name="emsName" value="ShoppingCartPage" />
</jsp:include>

```

where:

<jsp:include>

page The name of the JSP page containing the e-Marketing Spot code. In the Express Store sample store this code is in the file `include/emsContent.jsp`.

flush The flush attribute allows the Dynacache feature to be enabled. If `flush="true"`, then the content of the JSP, excluding the e-Marketing Spot, can be cached every time the page displays the e-Marketing Spot is refreshed.

<jsp:param>

name The value of the name attribute is a variable that is passed to the JSP file containing the e-Marketing Spot code, for example `include/emsContent.jsp`. If you are using the sample e-Marketing Spot, do not change this value. The default is `emsName`.

value The name of the page on which the e-Marketing Spot will display. The name should include the location, such as `HomePageAd`, or `CheckoutPageRecommendation`. This helps to reduce confusion about where it will appear, and what content it will contain. If necessary, add numbers to differentiate between two e-Marketing Spots that appear on the same page. You must use this same name when registering the e-Marketing Spot in the database by using the WebSphere Commerce Accelerator.

Note:

- a. If a merchandising association initiative is scheduled to the e-Marketing Spot, and the source of the association is based on the content of the page, it can be set using the `setSourceCatalogEntryId(String source)` and the `setMultipleCatalogEntryId(String source)` methods provided in `com.ibm.commerce.marketing.beans.EMarketingSpot`. For example, if the product displayed in the `ProductDisplay` page is the source of the association, the following method is invoked:

```
eSpot.setSourceCatalogEntryId(productId)
```

where:

eSpot is an instance of the `com.ibm.commerce.marketing.beans.EMarketingSpot` class.

productId is the identifier of the source product.

5. If you require more than one e-Marketing Spot per JSP file, repeat steps 2 on page 91 through 4.

6. Register the e-Marketing Spot in the database by using the WebSphere Commerce Accelerator. The e-Marketing Spot should be registered using the value of the `<jsp:param>` attribute, for example, `ShoppingCartPage`. To register the e-Marketing Spot in the database refer to the online help topic "Creating an e-Marketing Spot".
7. Schedule the campaign initiative for this e-Marketing Spot by using the WebSphere Commerce Accelerator. To schedule the campaign initiative refer to the online help topic "Scheduling a campaign initiative for an e-Marketing Spot".

Part 4. Running your store front

This part of the book introduces the main tasks involved in the day-to-day operation of your storefront. To perform the tasks described in this section, use the WebSphere Commerce Accelerator which is a browser-based tool.

The WebSphere Commerce Accelerator allows you to maintain your online store. You can create and update the store catalog; change the look-and-feel of the store, manage orders, and track store activities.

WebSphere Commerce Accelerator menus display tasks that you are authorized to perform. These tasks are based on user roles, authority levels, and the business model and type of store.

Chapter 16. Managing the catalog

This chapter describes the following tasks:

- Managing categories
- Managing catalog entries, such as products, bundles, prebuilt kits and SKUs

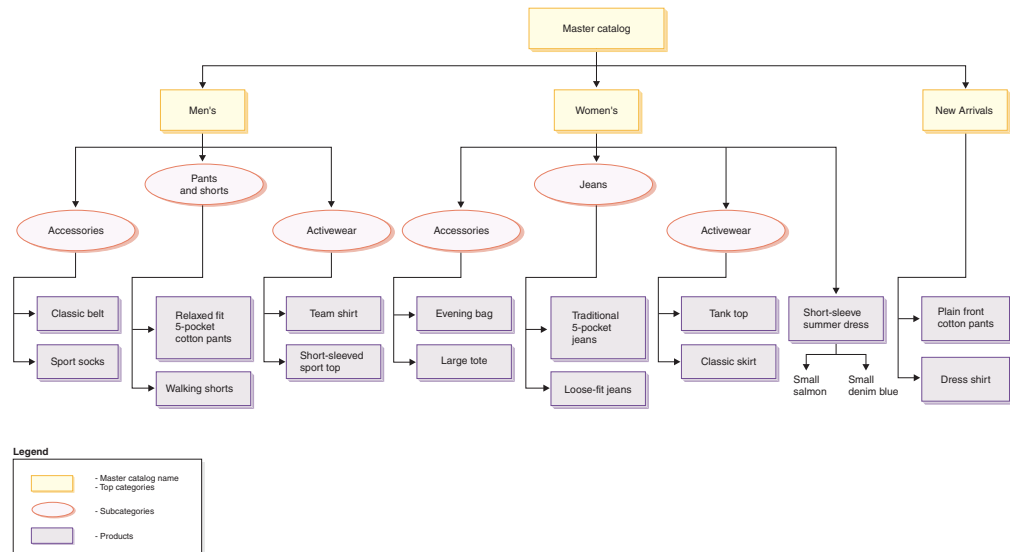
In WebSphere Commerce the primary tools to create and manage catalog information are the Loader package, and the WebSphere Commerce Accelerator product management tools. For more information about using the Loader package to create a catalog refer to Chapter 8, “Loading the catalog,” on page 49. This tool is run by the Site Administrator. The Product Manager and Category Manager can update products, by using the tools provided in the WebSphere Commerce Accelerator.

Category management

The Category Manager manages the category hierarchy by creating, changing and deleting categories.

The master catalog is the central tool for managing your store’s merchandise. The master catalog contains all the catalog information that the store requires. This is the single catalog containing all catalog entries. In WebSphere Commerce the catalog is multilingual, that is the descriptions, attributes and images can vary by language.

Using the Product Management tool to view your catalog provides a representation of the master catalog.

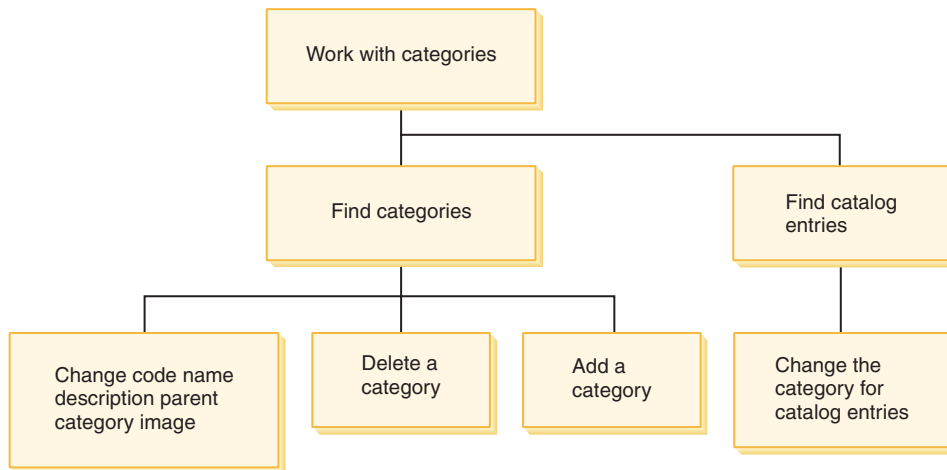


This part of the master catalog follows a basic navigation for a store containing clothing for Men and Women. Following the three top or parent catalog groups, are additional catalog groups (Accessories, Pants and shorts, and Activewear), going to products (Classic belt, and Sports socks) and their related SKUs. Note that products may be directly under a top category, for example short-sleeve summer dress is under the Women’s top category.

Managing categories

You can view the categories and the category hierarchy for the catalog. By viewing the parent and child associations between categories, you can determine where additional categories might be needed, and where some existing categories need to be either changed or deleted. For example, if you want to add a category for music-related videotapes, you can review the list of categories to determine whether or not the new category would be a parent category or if it should be associated with an existing category. In WebSphere Commerce you can have an unlimited number of hierarchical levels.

A category is a group of objects that have similar properties. Categories are used to organize products or services offered by the store. The Category Manager can create, find, list, change, and delete categories. You can classify products and SKUs under different parent categories.



Procedures for managing categories

Role	Task	Topic in the WebSphere Commerce online help
Category Manager	<ul style="list-style-type: none"> • Create a category • Change a category • Change the parent category of a child category • Delete a category 	"Working with the catalog"
Category Manager Product Manager	Change the catalog entry's category	"Working with the catalog"

Product management

Product management refers to the process of managing catalog entries. In WebSphere Commerce there are several kinds of catalog entries:

Products

Products act as templates for SKUs, the individual pieces of merchandise which are ultimately sold to a customer. The product wizard allows you to create new products for your catalog. Use the product notebook to view or change product details, such as names, descriptions, and images, or delete

products from your catalog. You can update product taxes, discounts, and shipping categories, and add shipping measurements for each product. You can also customize fulfillment options for your product, such as tracking inventory or backordering a product for an order.

SKUs Once you have created your product, you must create SKUs to represent each orderable item of merchandise for sale. All SKUs related to a particular product exhibit the same set of attributes and are distinguished by their attribute values. Create SKUs using the SKU wizard. Or you can automatically generate SKUs for a product, once you have created all required product attributes. The SKU notebook allows you to view or change SKU details, such as names, descriptions, and images, and delete SKUs from your catalog. You can update sales tax, discounts, and shipping categories, and add shipping measurements for each SKU.

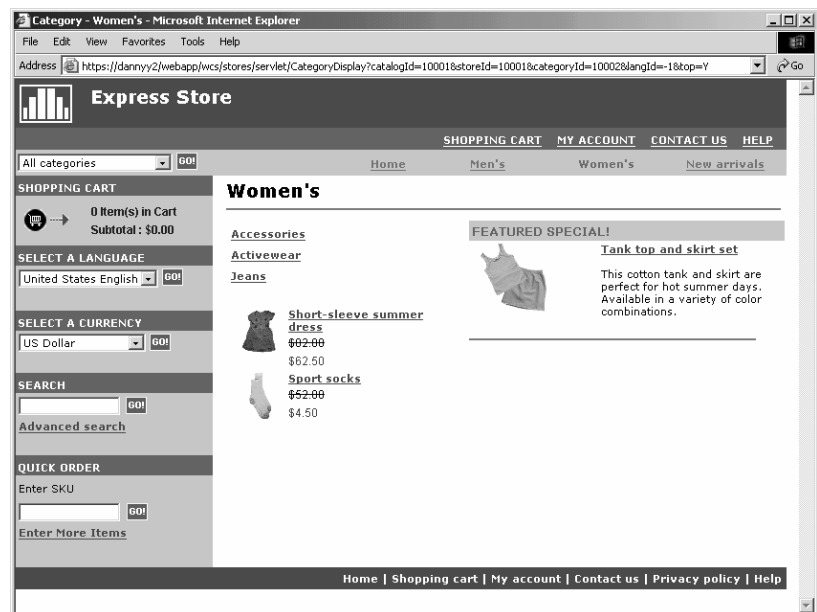
Bundles and kits

Bundles and kits are groupings of catalog entries. Used for promotional purposes, the catalog entries in bundles and kits are grouped in such a way to attract customers.

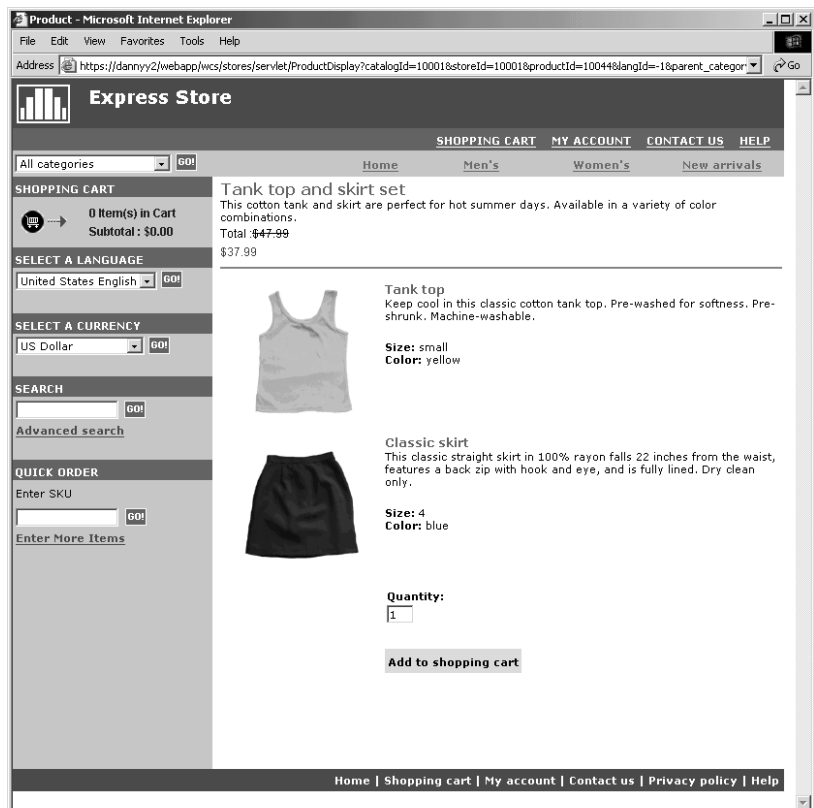
Prebuilt kit

A collection of catalog entries that cannot be sold separately. For example, a computer prebuilt kit might contain a specific central processing unit, monitor, and hard drive. A prebuilt kit has its own price and can be added to the shopping cart. Once added to the shopping cart, you cannot change the prebuilt kit contents. Viewed in similar ways as a product, a prebuilt kit has defining attributes.

The following image shows a kit in the Express Store sample store.



Notice that the customer cannot remove any SKUs from the kit.



Bundle

A collection of catalog entries which allow customers to buy multiple merchandise with one click. For example, when a bundle for a computer is a central processing unit, a monitor, a hard drive, and a CD-ROM drive, the components can be sold separately. A bundle's price is composed by its different catalog entries. A bundle cannot be purchased directly. The customer can remove one or more SKUs from the bundle.

Some catalog entries may also have attributes. There are two types of attributes: defining attributes and descriptive attributes.

Defining attributes

Defining attributes are properties of SKUs in an online store, such as color or size. Attribute values are the property of an attribute such as a specific color (blue or yellow) or size (medium). You must predefine attribute values before assigning them to SKUs. Attribute values are implicitly related to their attributes. Each possible combination of attributes and attribute values equals a new SKU. After creating attributes and their values, you can update information such as name, description, and type (text, whole numbers, or decimal numbers). Defining attributes are also used for SKU resolution.

Descriptive attributes

Descriptive attributes are simply additional descriptions. For example, some pieces of clothing should only be dry cleaned, never wash. A

descriptive attribute can specify the dry-clean-only condition. Or if electronics equipment uses AA or AAA batteries. Descriptive attributes are not used for SKU resolution.

Characteristics of products, SKUs, bundles and prebuilt kits

In the WebSphere Commerce Accelerator, catalog entries can contain the following characteristics:

Catalog entry characteristic	Product	SKU	Bundle	Prebuilt kit
General information Code Name Announcement date Withdrawal date Last updated Display to customers For purchase On auction	X	X	X	X
Description Short description Long description 1 Long description 2 Long description 3	X	X	X	X
Images to display on store pages Full size image Thumbnail image	X	X	X	X
Discount code	X	X		X
Manufacturer information Manufacturer part number Manufacturer name	X	X	X	X
Sales tax	X	X	X	X
Shipping tax	X	X	X	X
URL	X	X	X	X
Units of measure for shipping Weight Weight measure Sold in multiples of this amount Number of items per package Unit of Measure	X	X	X	X
Product fulfillment Track inventory Allow back order Force back order Release separately Returnable Creditable Inventory tracking amount Inventory tracking unit of measure	X			X
Category (Group the product and all associated SKUs under a category)	X		X	X

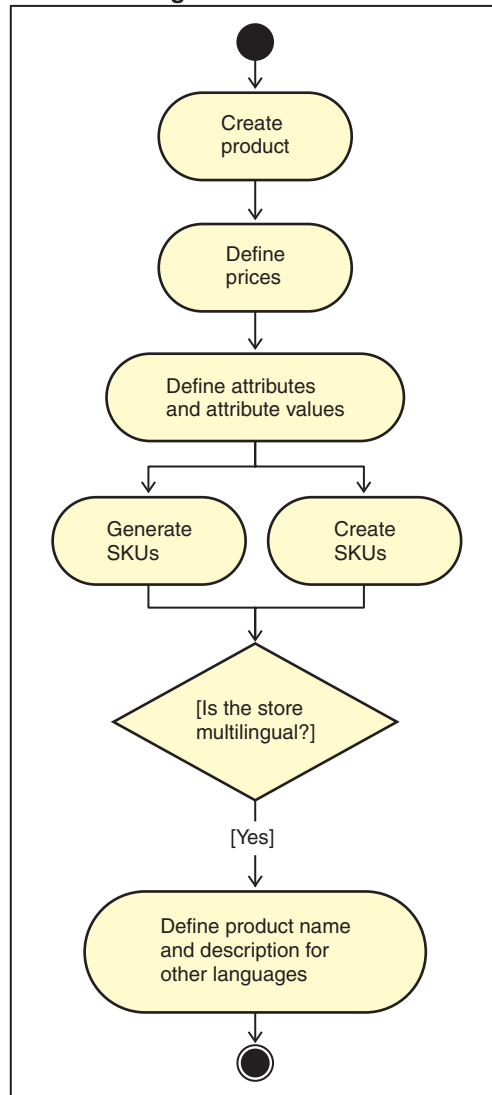
Catalog entry characteristic	Product	SKU	Bundle	Prebuilt kit
Attribute (Add or maintain defining attribute values for a SKU)		X		

Creating products, attributes, and SKUs

After the Category Manager has created categories in the catalog, the Product Manager can create catalog entries.

The following diagram shows the steps required to create a new product by using the WebSphere Commerce Accelerator.

Product Manager



Procedures for managing products

Role	Task	Topic in the WebSphere Commerce online help
Product Manager	Announce a product	"Adding a product"
	Discontinue a product	"Changing a product"
	Withdraw a product	"Changing a product"
	Specify whether a product is displayed to customers	"Hiding a product"
	Specify whether a product can be purchased within a specified period of time.	"Changing a product"
	Change the information for many products simultaneously (for example change the manufacturer name for all products within a given category)	"Finding and replacing text in the dynamic table"
	List all products containing a particular attribute value	"Finding a catalog entry"
	Pricing Set a price, or a price range, for a product Set a price, or a price range, for a SKU Set a price, or a price range, for a prebuilt kit	"Adding a price to a product" "Changing the price of a product"
	Add an additional description to a product, for example, "dry-clean only" on a shirt	"Adding a descriptive attribute"
	Create a grouping of products to enable customers to easily purchase several products at once, while still allowing customers to remove one or more items from the group.	"Creating a bundle"
	Create a collection of products that cannot be sold individually. Customers will not be able to remove items from the group.	"Creating a kit"
	Add an additional attribute to a product such as color or size	"Adding a defining attribute"
	Add an additional attribute value to a product, such as blue, or small.	"Adding an attribute value"
Delete catalog entries	"Deleting a product" "Deleting a SKU" "Deleting a bundle or kit"	

Role	Task	Topic in the WebSphere Commerce online help
Application developer	Create a catalog	<i>WebSphere Commerce Store Development Guide.</i>

Chapter 17. Managing customer information

This chapter describes the customer registration information that may be changed by a Customer Service Representative, (CSR). In addition to changing customer registration information the CSR may also register a customer at a store, reset the customer's password, and lock or unlock the customer's account.

Customer registration information

A Customer Service Representative can manage customer information for registered customers.

non-registered customer

A customer that does not have a customer logon ID and password for shopping at the store. The customer has not provided registration information, such as contact or demographic information, and any interests lists or shopping cart items will not be retained for future visits. A non-registered customer is considered a guest.

registered customer

A customer that has provided information to create a registration profile. This customer has a logon ID and password, which are required when submitting orders at the store. Any interest lists or shopping cart items are "saved" so that the customer can retrieve these items during return visits.

Changing customer registration information

Customer Service Representatives can manage customer information, such as a customer's address, password, or account status. The registration information that customers provide depends on the information collected on the store's Registration page. Some registration information may be optional. To view or update customer information use the Customer Information notebook. The notebook shows the following:

General

Security information, a challenge question and answer for identification purposes, and an X.509 security certificate status, if applicable. General information also includes the customer's full name, preferred shopping language and currency and the account status (enabled or disabled).

Address

Full address.

Contact

Contact information, such as e-mail addresses, phone numbers, and fax numbers. Customers can indicate their preferred method of communication and the best time to call. Customers can also specify whether or not they want to receive promotional material from the store.

Demographic

Demographic information, such as age, gender, marital status, annual income, number of household members, number of children, return customer, occupation, employer and hobbies.

Procedures for customer registration

Customer Service Representatives can perform the following tasks:

Role	Task	Topic in the WebSphere Commerce online help
Customer Service Representative	Find a customer Register a customer to a store View a customer summary Lock or unlock a customer's account Reset the customer's password Change the customer's address Change the customer's demographic information Change the customer's contact information Specify that the customer does not want to receive promotional e-mail from the store	"Customer information"

Chapter 18. Managing orders

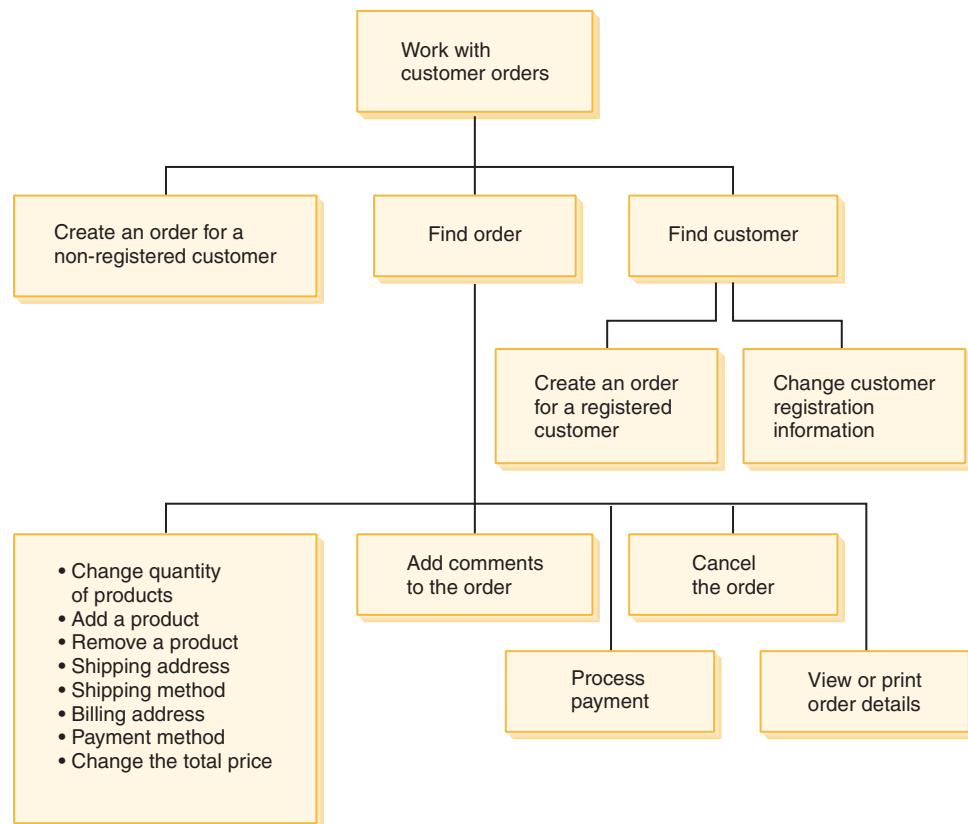
A Customer Service Representative can track and manage details about orders, including the customer, recipient, products and quantity, total cost (including tax and shipping charges), shipping specifications, payment method, and any comments.

In WebSphere Commerce an order is: one or more products, their prices, and the quantity specified, that a customer has selected to purchase or has purchased. A Customer Service Representative can also place an order on behalf of a customer. In addition to products, a customer order includes a billing address; shipping address (not applicable to downloadable purchases, such as software); shipping method, carrier, and service; payment information; tax and shipping charges; and any comments or price adjustments stipulated by the person placing the order.

This chapter provides an introduction to common order-related tasks.

Working with customer orders

The following diagram shows common order-related tasks.



Find customer orders

You can search for an order based on the order number, the customer logon ID, or order status. You can also sort orders to help find the information you are looking for. For example, you can sort orders by the order status or by the last name of the user.

Order status codes help you track the progress of orders placed by customers. Some commonly used order status codes are shown in the following table:

Code	Name	Description
I	Submitted	The customer has submitted the order, but has not yet initiated payment.
S	Shipped	All order items have been manifested.
X	Canceled	The order has been canceled.
R	Released	All order items have been released for fulfillment.
D	Deposited	Payment has been captured.

You can search for any order that is in the WebSphere Commerce database.

Change customer orders

Changes can be made to a submitted order so long as the order has not yet been released for fulfillment. Orders that have an order status of Shipped, Canceled, Released or Deposited cannot be changed.

Cancel an order

Orders that have an order status of Shipped, Canceled, Released or Deposited cannot be canceled. When you cancel a customer order, you can include a reason for the cancellation, such as the customer mistakenly ordered the wrong product, or the product selected is temporarily out of stock and the customer does not want to wait for the new stock. You can choose whether to send the customer an e-mail containing the text of this comment. **After an order has been canceled, it cannot be recovered.**

Add comments to an order

You can add a comment to an order for your own reference, or e-mail the comment to the customer. For example, if there is insufficient inventory to fulfill an order, you can indicate the reason in a comment and e-mail it to the customer. This way, customers are aware of delayed shipments.

Process payment for an order

Before an order can be shipped, payment must be approved. Payments may be handled using either of these methods:

- WebSphere Commerce Payments is used to process payments. To change the payment status of a customer order, Customer Services Representatives can link to WebSphere Commerce Payments from the WebSphere Commerce Accelerator.
- Re-enter payment information to another payment system. Use the method when WebSphere Commerce Payments is not the Payment method. View the order information; then re-enter payment information to the back-end or Point-of-Sale system. Setting up WebSphere Commerce Accelerator pages to see payment information is described in the section "Encrypting and decrypting credit card information" on page 88.

Create an order for a non-registered customer

Follow the Place Guest Order pages sequentially to provide the information required for an order, including the products, quantity, billing address, shipping address, shipping method, payment option, and a comment or special price adjustment, if desired.

Create an order for a registered customer

Follow the Place Order wizard pages sequentially to provide the

information required for an order, including the products, quantity, billing address, shipping address, shipping method, payment option, and a comment or special price adjustment, if desired.

Procedures for managing orders

Role	Task	Topic in the WebSphere Commerce online help
Customer Service Representative	Create an order	<ul style="list-style-type: none"> • Creating an order for a registered customer • Creating an order for a non-registered customer
	Change an order (quantity of products, add a product, remove a product, change shipping address, change shipping method, change total price, change payment option, change billing address, add a comment, process payment)	<ul style="list-style-type: none"> • Changing an order
	View order information (comments, order summary, payment status, invoices, personalized attributes)	<ul style="list-style-type: none"> • Viewing order information
	Process payments (using WebSphere Commerce Payments)	<ul style="list-style-type: none"> • Processing payment for an order • Changing payment processing for an order • Viewing the payment status for an order
application developer	Customize the Order Management subsystem	See the section on the <i>Order Management subsystem</i> WebSphere Commerce development environment online help.
Site Administrator	Delete stale orders from the database	See the section on the <i>Database cleanup utility</i> in the <i>WebSphere Commerce Administration Guide</i> .

Part 5. Marketing and merchandising

This part of the book introduces the main tasks involved in setting up marketing and merchandising activities in the Express Store. To perform the tasks described in this section, use the WebSphere Commerce Accelerator which is a browser-based tool.

The WebSphere Commerce Accelerator allows you to maintain your online store. You can create and update the store catalog; change the look-and-feel of the store, manage orders, and track store activities.

WebSphere Commerce Accelerator menus display tasks that you are authorized to perform. These tasks are based on user roles, authority levels, and the business model and type of store.

Chapter 19. Marketing

WebSphere Commerce facilitates marketing and personalization by enabling Marketing staff to interact with the site directly, instead of requesting that the Information Technology department make required changes. Marketing Managers create large-scale marketing campaigns that target various customer profiles. Campaigns drive site sales, increase revenue, or increase customer awareness.

This chapter provides the following information:

- Discounts
- e-Marketing Spots
- Customer profiles
- Marketing campaigns

Discounts

Discounts allow you to offer customers incentives to purchase. You can offer percentage discounts (such as 10% off), or fixed-amount discounts (such as \$15 off). Discounts can apply to specific products, specific SKUs, or to the total purchase. For example, you can offer a 20% reduction to senior citizens; or if you have many red baseball caps in stock, you can offer a 25% discount on the caps for a limited time.

Create discounts by using the Discount wizard in the WebSphere Commerce Accelerator. Users belonging to any of the following roles may create discounts: Product Manager, Buyer, Category Manager, and Sales Manager. Discounts can also be created by using the Loader package (by the Site Administrator), or imported during migration from previous version. These discounts will function correctly, but may not display properly in the WebSphere Commerce Accelerator.

Discounts can be either active, or inactive. Discounts are set as inactive by default when created, but can be activated at any time using the WebSphere Commerce Accelerator. For example, you might want to deactivate a discount before it expires if you notice that the inventory level for a discounted product is too low for the increased demand.

In WebSphere Commerce there are two ways in which you can implement discounts:

schema-based system

The discounts are defined using tables in the WebSphere Commerce database. The schema-based discount system is maintained for those sites being upgraded from previous versions.

rule-based system

The discounts are rules in the rule server in the WebSphere Commerce Server. If you are creating a new site, you should implement the rule-based discount system, since some discount types are not available using the schema-based system.

Discount types

The following list shows the different discount types available in WebSphere Commerce.

The following discounts are always available

- Percentage off total purchase
- Amount off total purchase
- Percentage off per product
- Amount off per product
- Amount off qualifying products

The following discounts are only available if you are using rules-based discounts

- Order level shipping discount
- Buy X of item A, give another A for free
- Buy product or item A, or B, or C, and give D as a gift

Assigning a discount to a product or SKU

You can assign a discount to either a product or a SKU. Assign a discount to a product if all orderable SKUs contained within the product are to be discounted. Assign the discount to a particular SKU if not all of the SKUs under a product are to be discounted.

Advertising discounts

You can advertise discounts using the marketing tools in the WebSphere Commerce Accelerator. Create ad copy which supports the discount, and select the ad copy when you are scheduling the advertisement to occupy a particular e-Marketing Spot.

Targeting a discount to a set of customers

To target a discount to a particular group of customers, you need to create a customer profile. The following table illustrates the differences between the calculation methods used (for product-level discounts), and the end results.

A customer buys 4 books which cost ten dollars each. The customer also purchases 3 DVDs which cost twenty dollars each. The total price without a discount is \$100. The discount is applied to determine the final price.

Discount type		Rules		Calculation	Result
Percentage off total purchase	Simple	25% off total purchase with no minimum qualifications.		$[(4 \times \$10) + (3 \times \$20)] \times (1 - 25\%)$	\$75
	Multi-range	0-\$39.99	No discount	$[(4 \times \$10) + (3 \times \$20)] \times (1-10\%)$	\$90
		\$40-\$109.99	10% off total purchase		
\$110 and up	20% off total purchase				
Amount off total purchase	Simple	Spend at least \$90 and get \$15 off total purchase		$[(4 \times \$10) + (3 \times \$20)] - \$15$	\$85
	Multi-range	0-\$99.99	No discount	$[(4 \times \$10) + (3 \times \$20)] - \$10$	\$90
		\$100-\$199.99	\$10 off total purchase		
\$200 and up	\$20 off total purchase				

Discount type		Rules		Calculation	Result
Percentage off per product	Simple	25% off each book when four or more books are purchased.		$[4 \times (\$10 \times \{1 - 25\%\})] + (3 \times \$20)$	\$90
	Multi-range	1-5 books	10% off each book	$[4 \times (\$10 \times \{1 - 10\%\})] + (3 \times \$20)$	\$96
		6 or more books	20% off each book		
Amount off for all qualifying products	Simple	When 3 or more books are purchased, get \$10 off the total order price		$[(4 \times \$10) - \$10] + (3 \times \$20)$	\$90
	Multi-range	1-5 books	\$10 off book sub-total	$[(4 \times \$10) - \$10] + (3 \times \$20)$	\$90
		6 or more books	\$20 off book sub-total		
Amount off per qualifying product	Multi-range	1-5 books	\$3 off each book	$[4 \times (\$10 - \$3)] + (3 \times \$20)$	\$88
		6 or more books	\$6 off each book		

Procedures for managing discounts

Role	Task	Topic in the WebSphere Commerce online help
Site Administrator	Enable rule-based marketing content discounts	Enabling WebSphere Commerce's personalized marketing
	Activating rules-based discounts	Checking the status of a rule service.
	Managing rules-based data	<ul style="list-style-type: none"> Propagating Rule-based discount data from a staging server to the production server Maintaining a staging server with rule-based discounts
Seller, Buyer (sell-side), Category Manager, Product Manager	Assign a discount code to a catalog entry (product, SKU, pre-built kit) Note that you cannot discount a bundle.	Discounts
	Create, activate, deactivate or delete a discount code	
	Create an order-level discount	
	Create a product-level discount	

Role	Task	Topic in the WebSphere Commerce online help
Application developer	Understand the discount data model	Calculation rule - discount data model. This topic is in the WebSphere Commerce development environment online help.
	Understand the discount implementation in a sample store	Configure and activate pre-defined discounts <i>WebSphere Commerce Sample Store Guide</i>

e-Marketing Spots

An e-Marketing Spot displays advertising content, category recommendations, or product recommendations.

Application developers place e-Marketing Spots on the appropriate pages. Each e-Marketing Spot has a unique name. The Marketing Manager selects the e-Marketing Spot that they want to target as part of the campaign creation process.

While the customer browses your store, the e-Marketing Spots communicate with the WebSphere Commerce Server to determine what content should be displayed in their place. A JSP page may contain any number of e-Marketing Spots. Any JSP page in your site may contain an e-Marketing Spot.

The following image shows a product recommendation on the shopping cart page of the Express Store. This e-Marketing Spot is created when the fashion sample catalog is loaded. e-Marketing Spots may also be created by using the WebSphere Commerce Accelerator.

Shopping Cart - Microsoft Internet Explorer

Address: <https://dannyy2/webapp/wcs/stores/servlet/OrderItemDisplay?langId=-1&krypto=c7MxXthuGMKX5v4rvpAujkMEMaEn%2BGsI8wFrRe1%2Fo>

Express Store

SHIPPING CART MY ACCOUNT CONTACT US HELP

All categories

Home Men's Women's New arrivals

Shopping cart

Your cart contains:



Quantity	Item	Price	Total	
1	Tank top and skirt set	\$47.99	\$37.99	Remove
	Tank top and skirt set	\$37.99		
1	Sport socks	\$5.50	Free	Remove
		\$4.50		

* opromo2 **-\$4.50**

Subtotal: \$37.99

or

We also suggest:

 Tank top \$31.89	 Team shirt \$25.99
---	--

Home | Shopping cart | My account | Contact us | Privacy policy | Help

If an e-Marketing Spot contains an advertisement; then you must define ad copy to be placed in the e-Marketing Spot.

ad copy

Ad copy refers to all of the supporting material created for your campaigns. Graphic artists and writers create ad copy in Flash, .gif or JPEG format. Ad copy includes product pictures, graphics, and marketing text.

Customer profiles

Customer profiles define groups of customers that have common sets of characteristics. These profiles identify targets for your marketing campaigns. You create profiles by selecting criteria and then assigning target values. When a customer visits your site, the system compares their personal information against the specified criteria. Depending on the customer, and the profiles that you have defined, profile-specific awareness advertisements or suggestive selling messages display. There are numerous possible criteria, including: data supplied during registration; purchase history information; and information about the frequency of a customer's visits.

For example, you can assign profiles for groups such as senior citizens, by creating a profile that targets customers who are at least sixty years old. You can further refine this profile to target senior women by specifying that in addition to the age requirement, the profile also targets customers who are female.

Profiles serve as dynamic targets for your marketing efforts. Depending on the criteria defined, your customers may or may not fit a particular profile at any given time. For instance, you might create a profile named "New Customers", which includes only customers who have registered in the last 30 days. While most of your customers would not belong to this profile, all new customers would fit this profile upon registration. New customers would fit this profile for a maximum of 30 days. This is dynamic because it requires no intervention on your part, applies to customers on an individual basis, and is determined in real-time. The last date for inclusion in this profile depends on the customer's date of registration and the maximum days limit defined during profile creation.

If you choose to use customer profiles, you must create them prior to creating marketing campaigns. During campaign creation, you select the target profile. You can create campaigns that are applicable to all customers; however, customer profiles can target a specific sub-set of customers.

Customer profile management

Use the Customer Profile page to view the defined customer profiles for the current store. You can sort the displayed profiles, create new profiles, create copies of profiles, delete profiles, see the customers who belong to a particular profile, and modify profiles.

Customer profile creation guidelines

To effectively use customer profiles, the store's pages must capture the information required by the profile. Marketing Managers must communicate their requirements to Store Developers. A well-designed registration page, that communicates how customers enter information, will eliminate potential problems. Note the following:

- Strings are matched exactly and are case-sensitive.

To create a customer profile based on state, you could enter "New York" on the State or Province page in the Customer Profile notebook. Any customer who enters "New York" on the store's registration page is included in the profile. However, a customer who enters "NY" is not included. To target all intended customers you could use a list on the registration page. Instead of typing the

state, customers would select the state from the list. Consider using lists for country or region of residence, city of residence, interests, phone number, and either zip or postal code.

- Any found strings will include the customer in the profile.

To create a customer profile based on an area code, you could enter 905 on the Phone Number page of the Customer Profile notebook. Any customer whose phone number has the three digits 905 is included in the profile. This is true whether they correspond to the customer's area code, exchange, or the end of the phone number. If you specify (905), but the customer types 905, that customer is excluded from the profile. To avoid this problem, you should specify the parentheses when creating the profile. The application developer should ensure that the Store Registration page validates that there are parentheses around the area code.

Campaigns

There are two primary components to any campaign, which work together to provide a complete, flexible marketing strategy.

1. The Campaign.

A campaign is a major marketing effort. A campaign has associated business objectives. They function as containers for initiatives.

2. The Campaign initiatives.

Campaign initiatives deliver the personalized content. Initiatives are coordinated activities that are designed to achieve the overall goal of a campaign. They represent the individual messages used to achieve a campaign's business objective. There are two types of initiative.

Suggestive selling

Used to provide a product or category recommendation. These recommendations suggest associated products in an attempt to up-sell or cross-sell products to a customer. Suggestive selling initiatives also remind customers of related products that they might want to consider purchasing. The recommendations are based on the customer's previous purchases, or the current contents of the customer's shopping cart.

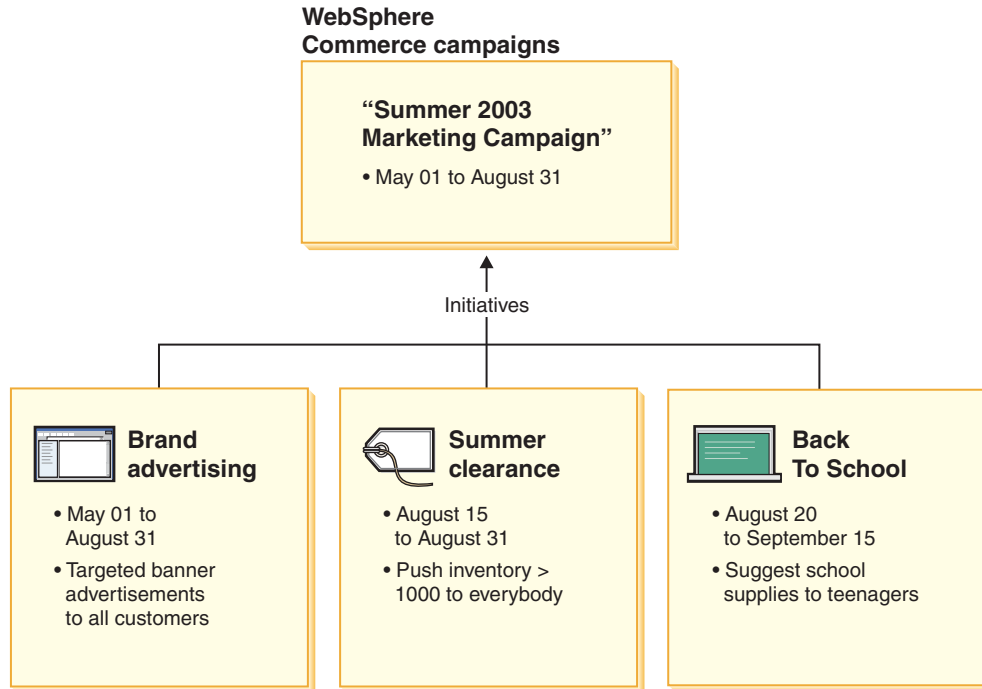
Awareness advertisement

Used to display advertisements that are designed to increase a customer's awareness. For instance, awareness advertisements inform customers about upcoming events, or increase brand awareness.

Campaign initiatives also have one or more target locations, namely e-Marketing Spots.

Suggestive selling initiative

The following diagram shows the relationship between a campaign and the initiatives contained within it.



The campaign contains the following three initiatives:

Back to school

This is a suggestive selling initiative, which targets a customer profile called teenagers, and recommends school supplies. The back to school initiative runs from August 20 until September 15.

Summer clearance

This is a suggestive selling initiative, which targets all customer profiles, and recommends all products that have an inventory count greater than 1000. The summer clearance initiative runs from August 15 until August 31.

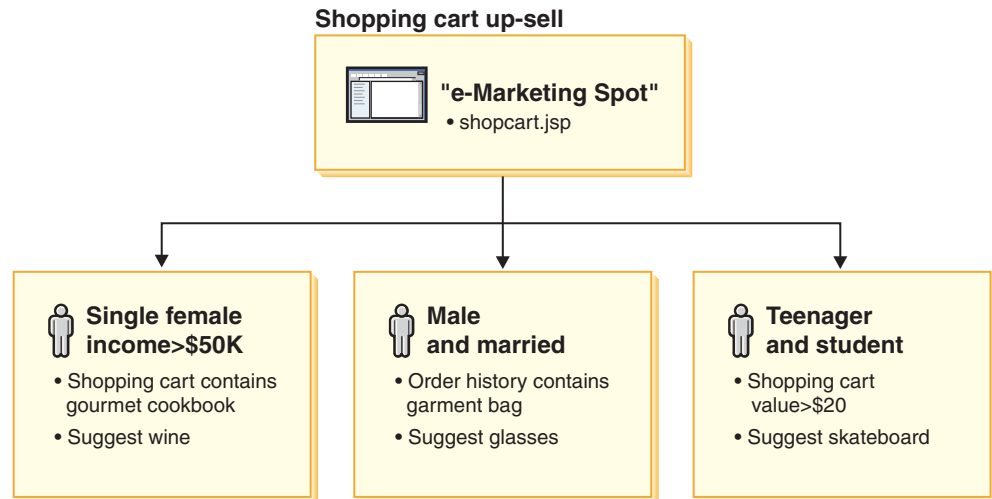
Brand advertising

This is an awareness advertisement initiative, which targets all customer profiles, and shows banner advertisements that are intended to increase brand awareness. The brand advertising initiative runs from May 1 until August 31.

Suggestive selling initiative with conditions

The following diagram shows a relationship between e-Marketing Spots, customer profiles, and suggestive selling initiatives. A suggestive selling initiative results in

dynamic content.



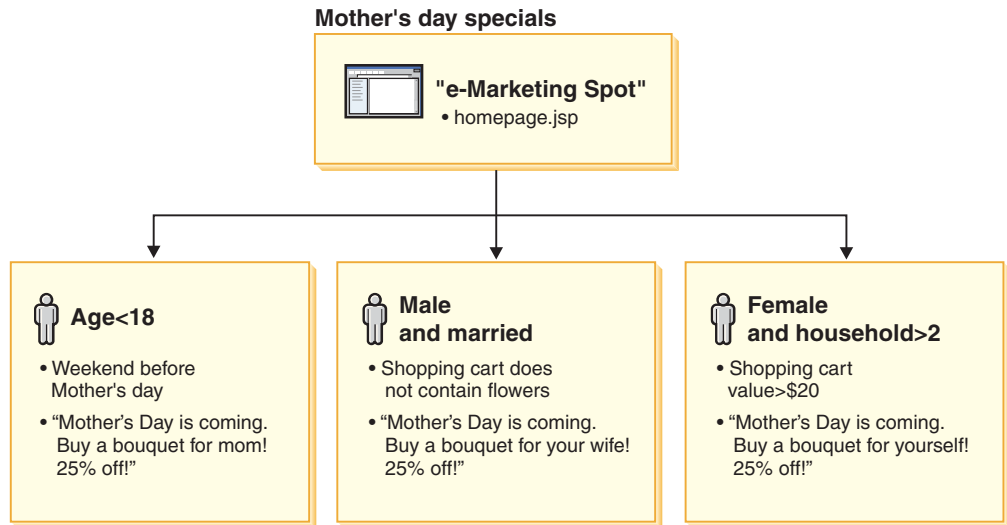
This page has a defined e-Marketing Spot. This e-Marketing spot corresponds to a suggestive selling initiative. The initiative associated with the e-Marketing Spot has three conditions. The conditions specify the target customer profile, the situation in which the condition is true, and the product recommendation to display when the condition is true. When this page displays, one of the following four things can happen:

- If the customer is single, female, earns more than fifty thousand dollars per year, and her shopping cart contains a gourmet cookbook, recommend a bottle of wine.
- If the customer is male, married, and his order history contains a garment bag, recommend a pair of sunglasses.
- If the customer is a teenager, and a student, and their shopping cart has a current value of greater than twenty dollars, recommend a skateboard.
- If the customer does not fit any of these profiles, or if the situation does not meet the specific criteria, no recommendations display.

Awareness advertisement

For each campaign initiative that you create, you must define a condition when the initiative is true. Associate each condition with an individual initiative. Conditions lay out the scenarios under which a particular initiative's content displays. To create conditions you answer questions; your answers define the condition. The questions include who the initiative should target, when (which day of the week) the initiative is considered, what content should be displayed, and which customer behaviors should be targeted. Your answers to these questions define the situation in which a particular condition will be true, and the personalized content will display. Conditions also provide the ability to display multiple messages in the same spot, under different situations.

The following diagram shows the relationship between e-Marketing Spots, customer profiles, and awareness advertisement initiatives. Imagine that an online store has a spring campaign, and an initiative designed to raise awareness for Mother's day.



The e-Marketing Spot corresponds to an awareness advertisement initiative for Mother's Day. The initiative associated with the e-Marketing Spot has three conditions. The conditions specify the target customer profile, the situation in which the conditions will be true, and the target ad copy to display when the condition is true. When this page displays, one of the following four things can happen:

- If the customer is under eighteen years old, and the current day is the weekend before Mother's Day, display an advertisement to remind the customer to buy flowers for their mother.
- If the customer is male, married, and his shopping cart does not contain flowers, display an advertisement to remind the customer to buy flowers for his wife.
- If the customer is female, has a household size of greater than two, and her shopping cart has a current value of more than twenty dollars, display an advertisement to remind the customer to buy flowers for herself.
- If the customer does not fit any of these profiles, or if the situation does not meet the specific criteria, no advertisements display.

Campaign management

The Campaign notebooks facilitate marketing campaign creation and maintenance. These online campaigns organize marketing techniques, including suggestive selling and awareness advertisements. Campaigns represent large marketing efforts, typically associated with an occasion, or objective, such as a back-to-school campaign or a Christmas campaign. The initiatives and their conditions implement rule-based processing that generates personalized content on a customer-by-customer basis.

The content of a campaign displays on your store's JSP pages. Special data beans called e-Marketing Spots contain the content. Application developers create e-Marketing Spots, and use them as placeholders during page design.

Use the Campaigns page to show the marketing campaigns defined for the current store. You can also perform other actions from this list, including creating new campaigns, deleting campaigns, and modifying existing campaigns.

Chapter 20. Setting up an e-mail campaign

E-mail activities allow you to deliver news and promotions to customers using e-mail. This allows you to reach customers who may not have visited your site in some time, or to keep regular customers up-to-date regarding upcoming events or new products.

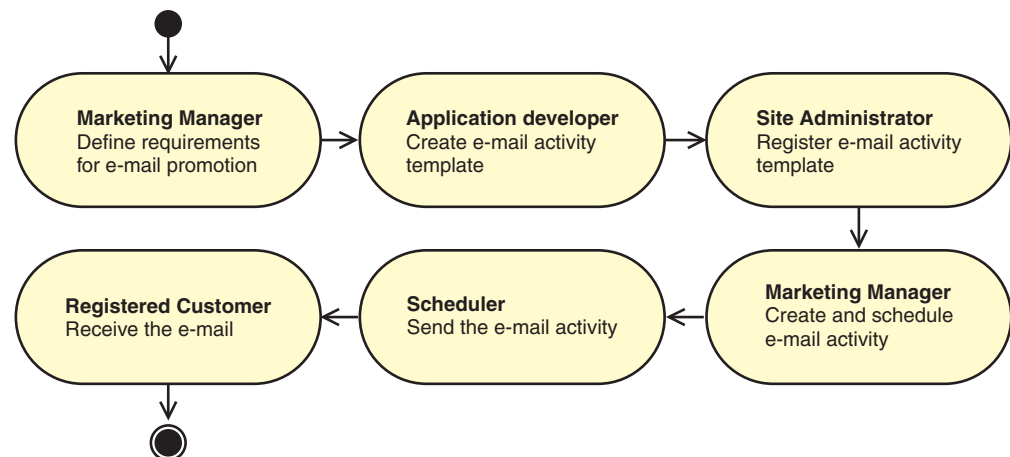
E-mail activities send a single dynamic e-mail message to multiple recipients. An e-mail activity is sent multiple times: once to every selected target e-mail address. It is not sent as a single e-mail with multiple target addresses. This eliminates the ability of a recipient to see the e-mail addresses of the other recipients, and should reduce privacy concerns.

Care should be taken in any endeavor to mass e-mail customers. There is growing concern, and in some places, laws forming, around what is allowable depending on the level of consent, whether there is no consent, implied consent, or full consent. Any customer e-mail activities should ensure compliance with the latest developments in this legal area.

E-mail activity templates

E-mail activity templates are JavaServer Pages, which define the address and subject of the e-mail, in addition to the content. The templates contain data beans, which access the commerce database to generate dynamic content. Data beans retrieve the addresses of the target customer profile, the subject, the return e-mail address, and finally the body of the e-mail.

The following diagram illustrates the workflow of creating an e-mail campaign.



Creating an e-mail activity template

This section describes how to create an e-mail activity template. The templates are created as JavaServer Pages which are capable of providing robust, dynamic, and personalized e-mail messages.

In order to receive e-mail activities, the customer must have a corresponding entry in the EMLUSRRECV table. If the customer does not have an entry in the

EMLUSRRECV table, or if the RECEIVEEML column in that table is set to '0', the customer will not receive e-mail. That is, the customer only receives e-mail if there is a corresponding '1' in the RECEIVEEML column. A customer profile is not store-specific, therefore, an entry of '0' in the STORE_ID column represents a customer being opted-out of all e-mail activities in all stores. The Registration and Change Registration pages in the Express Store contain a field that allows customer to opt in to receiving promotional e-mail. For instructions on how to enable this field see "Changing store flows" on page 17.

The EmailActivityUrlHelperDataBean class allows the customer a one click opt-out method, for e-mail activities.

The content of an e-mail message is separated from the address (both the sender and recipient) and subject elements of the message. A typical example of a MIME e-mail message header follows:

```
Received: from localhost (localhost)
  by some.address.com (1.12.256/1.12.256) id g8UG02Hj033414;
  Tue, 22 Oct 2002 14:43:22 -0400
Date: Tue, 22 Oct 2002 14:43:22 -0400
From: Mail Delivery Subsystem <MAILER-DAEMON@some.address.com>
Message-Id: <200209301843.g8UG02Hj033414@some.address.com>
To: <yourname@anotherAddress.com>
MIME-Version: 1.0
Content-Type: multipart/report; report-type=delivery-status;
  boundary="g8UG02Hj033414.1033411402/some.address.com"
Subject: Returned mail: see transcript for details
--g8UG02Hj033414.1033411402/some.address.com
```

The original message was received at Tue, 22 Oct 2002 13:57:55 -0400 from yet.another.address.com [256.256.256.256]

As can be seen from the areas marked in bold, the content of the e-mail is distinctly separate from the header information. This separation influences software APIs and products which are developed on top of these e-mail formats.

When creating an e-mail template, you must adhere to the following specification rules:

1. If the URL query string requesting the JSP contains the "returnSubject=true" parameter, then the JSP should render only what would be on the subject line of the e-mail message.
2. If the URL query string requesting the JSP does not contain the "returnSubject=true" parameter (that is, it is set to anything but "true", or does not exist), then the JSP should render the content of the e-mail message, except for the subject line data.
3. The URL query string registered as part of an e-mail template must have the following parameters added during the request:

recipientid

The userID of the user for whom to generate the e-mail.

storeid

The store entity ID of the store for this e-mail template.

activityid

The ID of the e-mail activity that made this request.

These parameters are populated during e-mail creation.

Conformance to the above rules ensures that the resources requested by the URL registered to the e-mail template will provide dynamic multi-part, e-mail messages. Since all parameter data is passed by way of the URL query string, the receiving URL need not be a JSP. In fact, any resources that can read and respond according to the above rules may be considered valid e-mail templates. In practice, however, it may be much more difficult to achieve the same rich content produced by JSP files running within the WebSphere Commerce product.

Store e-mail activity templates in a location that is accessible to the WebSphere Commerce Server. They must also be registered in the commerce database by a database administrator before they will be available to the WebSphere Commerce Accelerator.

Registering an e-mail activity template

E-mail templates must be registered in the commerce database using a command line script which executes SQL statements. These statements insert data into the EMLMSG table of the corresponding instance.

To register a new e-mail activity template, do the following:

1. Ensure that you know the *storeid*. The *storeid* is the value located in the STORE_ID column of the STORE database table. To determine the *storeid*, do the following:
 - a. Open a DB2 command window.
 - b. Connect to the WebSphere Commerce database.
 - c. Run the following command:

```
select * from storeent
```
2. To register an e-mail activity template on iSeries proceed to step 6 on page 126.

Windows and Linux

3. Log on as the database user.
4. At a command prompt, navigate to the *WCinstall_dir/CommerceServer55/bin* directory.
5. Type the following command:

Windows

```
createEmailTemplate.db2 db_name db_user JSP_path  
property_file template_name "template_description" storeId
```

Linux

```
createEmailTemplate.db2.sh db_name db_user JSP_path  
property_file template_name "template_description" storeId
```

When prompted, enter the user's database password.

where:

db_name

is the name of the WebSphere Commerce database.

db_user

is the database user name for the specified *db_name*.

JSP_path

The fully qualified URL of the JSP associated with the template on the WebSphere Commerce Server.

property_file

The property file associated with the template. If there is no property file, specify *noTemplate*.

template_name

The name of the template.

template_description

A short description for the template.

storeId The store ID of the target store.

For example,

```
RegisterJSP mall db2guy db2man 1 "http://host/webapp/wcs/stores/store1/template.jsp"
"http://host/webapp/wcs/stores/store1/template.jsp" "Male Summer Sports Promotion"
"This is going to help us sell more golf and water apparel."
```

This registers the template. The template is available to the Marketing Manager under the name Male Summer Sports Promotion.

iSeries

- ▶ 400 Using the Operations Navigator, or some other comparable method, run the following SQL on your instance:

```
insert into schemaname.emlmsg
(emlmsg_id, jspath, propertyfile, name, description, storeent_id)
select coalesce(max(emlmsg_id), 0) + 1, 'JSP_path', 'property_file',
'template_name', 'template_description', storeId from emlmsg ;
```

where:

schemaname

is the name of your instance.

JSP_path

The fully qualified URL of the JSP associated with the template on the WebSphere Commerce Server.

property_file

The property file associated with the template. If there is no property file, specify *noTemplate*.

template_name

The name of the template.

template_description

A short description for the template.

storeId The store ID of the target store.

Configuring e-mail activity accounts

1. Open the Administration Console:

▶ Developer

https://host_name/webapp/wcs/admin/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogon

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

▶ 400

▶ Linux

▶ Windows

https://host_name:8002/adminconsole

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server.

2. Select **Store** on the Administration Console Site/Store Selection page.
3. From the Configuration menu, click **E-mail Activities**. The E-mail Activity wizard opens.

4. Complete the fields on the Outbound E-mail Activity Account Configuration page with the information for an outbound SMTP account, and click **Next**.

Name Type a name for the account.

Description

Type a brief description for the account.

Host Type the host name of the SMTP server to connect with.

Account

Type the e-mail account to use on the SMTP server. This acts as a user ID when connecting to the server.

Requires Authentication

Select this option if the e-mail account requires authentication prior to connecting to the server. Enter the password in the **Password** field.

Port Enter the port number on the SMTP host which accepts SMTP requests. The default is 25.

Commerce Host

Enter the host name of the WebSphere Commerce Server. This is the machine from which the e-mail activity will be sent.

Start Retrieval at

Enter the time at which the account should begin mailing pending e-mail activities. The date and time are based on the current system time for the machine where WebSphere Commerce Server is installed.

5. Complete the fields on the Inbound E-mail Activity Account Configuration page with the information for an inbound POP3 account. This information is used by the inbound POP3 e-mail account to receive returned mail associated with your e-mail activities.

Name Type a name for the account.

Description

Type a brief description for the account.

Host Type the host name of the POP3 server to connect with.

E-mail Address

Type the e-mail address of the inbound account.

Account

Type the e-mail account to use on the POP3 server. This acts as a user ID when connecting to the server.

Requires Authentication

Select this option if the e-mail account requires authentication prior to connecting to the server. Enter the password in the **Password** field.

Port Enter the port number on the POP3 host which accepts POP3 requests. The default is 110.

Commerce Host

Enter the host name of the WebSphere Commerce Server. This is the machine that will retrieve e-mail associated with the e-mail activity.

Time to start retrieval

Enter the time at which the account should begin mailing pending e-mail activities. The date and time are based on the current system time for the machine where WebSphere Commerce Server is installed.

6. Click **OK** to save the e-mail activity configuration.

Creating a new e-mail activity

To create a new e-mail activity, do the following:

1. Open the WebSphere Commerce Accelerator.

▶ Developer

`https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=common.mcLogon`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce - Express development environment machine.

▶ 400 ▶ Linux ▶ Windows

`https://host_name:8000/accelerator`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server machine.

2. From the Marketing menu, click **E-mail Activity**. The E-mail Activity window opens, containing a list of the currently defined e-mail activities.
3. Click **New**. The E-mail Activity dialog displays.
4. Complete the page as required selecting the values appropriate for the e-mail activity you want to create.

Name Type a name for the e-mail activity. This is displayed in the E-mail Activity list.

Description

Type a description for the e-mail activity.

Specify a customer profile

Select a customer profile for sending the e-mail activity. This list is populated with the customer profiles defined for the store.

Specify an e-mail template

Select an e-mail template to send using the e-mail activity. This list is populated with the e-mail templates.

Start delivery on

Specify the date that the e-mail activity should be sent.

5. Click **Finish** to save the e-mail activity. The e-mail activity is displayed in the E-mail Activity list.

Chapter 21. Reporting and business intelligence

This chapter describes the operational reporting and business intelligence capabilities of WebSphere Commerce. In WebSphere Commerce there are two types of reports:

Business intelligence reports

More robust business intelligence reporting capabilities are provided by WebSphere Commerce Analyzer, an optional component of WebSphere Commerce.

Operational reports

Operational reports include inventory, product, return and Product Advisor usage.

Analytical reports and business intelligence

WebSphere Commerce Analyzerr is a comprehensive set of tools for analyzing e-commerce results. In particular, the WebSphere Commerce Analyzer data mart contains extractions from the WebSphere Commerce operational data, resulting in extensive information about the business. Using the delivered reports for analysis, business users can gain insight into trends and customer characteristics, make better decisions, form strategies, and learn how to improve their business.

Features include:

- A data mart, installed on a separate server, where you can run detailed data analysis and reporting without affecting the performance of the WebSphere Commerce operational systems. The data mart is designed with predefined and expandable reporting tables to provide historical data for comparison purposes.
- A set of tools to extract, transform, and load data from the WebSphere Commerce operational database into the data mart. These extraction tools come with predefined extractions that are easily expanded to address your custom data requirements.
- The Reporting Framework delivers an extensive list of analytical reports. Over 250 reports in 16 categories ensure the user will have the data needed specific to the area of interest. This framework is delivered with the WebSphere Commerce Analyzer, so companies can expand the reports delivered with the system by changing existing reports, or creating new ones. This Reporting Framework replaces the Brio Broadcast Server included in previous releases. If you want more robust reporting capability, reporting integration kits will be available for several of the commercial reporting vendors.

Closed-loops analytics

Closed-loop analytics provide rapid 'analytics to actions' technology. Closed-loop analytics integrate advanced data mining analysis results so that a Marketing Manager can create an explicit customer profile based on the segments generated from WebSphere Commerce Analyzer's data mining function.

This closed-loop function eliminates the manual steps between WebSphere Commerce Analyzer and WebSphere Commerce by automating the creation of an

explicitly targeted customer list. More accurate customer profiles are created by using the segment (member list) generated by WebSphere Commerce Analyzer (explicit profile).

WebSphere Commerce operational reports

Expected inventory management

These reports depend on ATP (Available to Promise) data. ATP is not enabled in the Express Store.

Inventory receipts through expected inventory records

This report reviews expected inventory record receipts.

Open expected inventory records

This report reviews expected receipts and outstanding expected inventory records.

Inventory reports

These reports depend on ATP (Available to Promise) data. ATP is not enabled in the Express Store.

The WebSphere Commerce Accelerator provides the following reports to track problems regarding inventory.

Inventory receipts by product

This report lists all receipts: receipts received through expected inventory records, ad hoc receipts, and receipts added as a result of product disposition. Report options include the start date, end date, SKUs, and fulfillment centers. The report can be sorted by either the SKU, or the date received.

Inventory adjustments

This report shows inventory adjustments by product. Report options include the start date, end date, SKUs, fulfillment centers, and adjustment reasons. The report can be sorted by the adjustment date, or SKU. The report is grouped by the product, fulfillment center, and adjustment date.

Inventory status

This report shows the current inventory allocation and receipt status of products in stock. Report options include SKUs, and fulfillment centers. The report is grouped by the fulfillment center and SKU.

Inventory receipts through expected inventory records

This report shows expected inventory record receipts.

Shipping exceptions

This report shows which releases are missing manifest information.

Products on backorder

This report shows cumulative backorder information by product. Use this report to review the current inventory allocation and receipt status of products in stock. Report options include SKUs and fulfillment centers.

Overdue backorders

This report shows orders that have products on backorder that are overdue.

Open expected inventory records

This report shows open expected inventory records. Use this report to review expected receipts and outstanding expected inventory records.

Marketing reports

Before using the Marketing reports, the Site Administrator must enable the Marketing Events monitor.

Campaign initiatives generate statistics about their use. These statistics can be viewed by Marketing Managers, using the WebSphere Commerce Accelerator. The statistics illustrate an initiative's clickthrough rate for each e-Marketing Spot where it is implemented. These statistics provide feedback on the effectiveness of the initiative, as well as comparative success rates among the various locations in which it displays.

Campaign initiative statistics

For each campaign initiative the following information is displayed: e-Marketing Spot, visits, clicks, click ratio.

Product management reports

The WebSphere Commerce Accelerator provides the following reports to manage products.

Order summary

This report summarizes orders received by the store for a user-defined sales period.

Discounts

This report displays information about the sales revenues and sales targets associated with defined discounts.

Outstanding returns

Determine which returned inventory has not yet been received.

Products on backorder

Determine which products, and how many of each, are on backorder.

Inventory receipts by product

Review inventory receipts.

Inventory status

Review current inventory allocation and receipt status of products in stock.

Return management reports

These reports depend on ATP (Available to Promise) data. ATP is not enabled in the Express Store.

Returns with partial receipts

This report shows returns for which at least one product has been received from the customer and at least one product has not been received from the customer.

Return receipts not disposed

This report shows returns by product for those products that have been received but not yet disposed.

Outstanding returns

This report shows returns that have been recorded, but for which none of

the products have been received from the customer. This report does not depict returns that have been logged where some of the products have been received from the customer; these returns can be found in the Returns with Partial Receipts report.

Shipping exceptions

This report depend on ATP (Available to Promise) data. ATP is not enabled in the Express Store.

Shipping exceptions

Determine which releases are missing manifest information.

Appendix A. Use cases

Adding a new address

Actor	Registered Customer
Main flow	<p>The customer clicks Add a new address. The system displays a page with the following fields:</p> <ul style="list-style-type: none">• Nickname• First name• Last name• Street address (composed of two text boxes)• City• State or Province• ZIP code or Postal code• Country/Region• Phone number <p>The customer types the information in the fields and clicks Submit. The system adds a new address to the address book (E1).</p>
Alternate flow	None
Exception flow	<p>E1: Missing Mandatory Field</p> <p>If any of the following fields are missing, system issues an error message.</p> <ul style="list-style-type: none">• First name• Last name• Street address• City• State or Province• ZIP code or Postal Code• Country/Region <p>If the nickname entered already exists in the customers address book, the system reports an error message.</p>

Implementation details

Adding a new address uses the following JavaServer Pages (JSP) files:

- AddressBookForm.jsp
- AddressForm.jsp

Adding an item to a requisition list

Actor	Customer
Main flow	<ol style="list-style-type: none"> 1. The customer browses to the product display page and from the Requisition list drop down list, selects New and then Go. 2. The Edit Requisition List page displays with the SKU added in the first item field and a quantity of 1 (or more if specified). 3. The customer types a Requisition List name and clicks Change Name. 4. The Edit Requisition List page displays with the newly created list.
Main flow of adding to an existing requisition list	<ol style="list-style-type: none"> 1. The customer browses to the product display page and from the Requisition list drop down list, selects an existing requisition list name and then Go. 2. The Edit Requisition List page displays with the new item added to it (it should also contain all old items).
Main flow of adding and modifying an existing requisition list	<ol style="list-style-type: none"> 1. The customer clicks on My Account in the header. 2. The My Account page displays. 3. The customer clicks View Requisition List. 4. The Requisition list page displays with all the actor's created lists (assuming the actor has a list). 5. The customer clicks on a requisition list name. 6. The Edit Requisition List page displays showing all items in it. 7. The customer modifies the Requisition list name and clicks Change Name. 8. The Edit Requisition List page displays showing all items in it with a new name (or with an error message if the name is not appropriate). 9. The customer changes the quantity of an item in the list and clicks Update Quantity. 10. The Edit Requisition List page displays showing the updated quantity (or an error message if the quantity is entered incorrectly) 11. The customer types a SKU into the SKU text box and a quantity into the Quantity text box and clicks Add. 12. The Edit Requisition List page displays with the new SKU added to it (or an error message if the SKU is not found in the store).
Alternate flow	None
Exception flow	<ol style="list-style-type: none"> 1. The customer clicks on My Account in the header. 2. The My Account page displays. 3. The customer clicks Requisition List. 4. The Requisition list page displays and tells the customer they do not have any requisition lists created. A link displays to Create New Requisition List.

Implementation details

Supporting a requisition list uses the following JavaServer Pages (JSP) files: (this does not include all of the catalog pages used to get to the Product display page.)

- CachedProductOnlyDisplay.jsp
- MyAccountDisplay.jsp
- ProductOnlyDisplay.jsp
- RequisitionListAddToListForm.jsp
- RequisitionListCreateForm.jsp
- RequisitionListDisplay.jsp
- RequisitionListUpdateForm.jsp

Adding an item to the wish list

Customers can view and edit the products they have selected to purchase through the shopping cart.

In Express Store, a wish list allows Registered Customers to add products to a list that they would like to order in the future. A customer can send the wish list by e-mail to family or friends, who can then purchase the items as gifts for the customer. A wish list differs from a shopping cart in that customers plan to purchase the products in the shopping cart during the current shopping session.

Precondition	The wish list feature is enabled in the store, for more information see Chapter 2, “Configuring the Express Store,” on page 11.
Actor	Registered Customer
Main flow	The customer views a product, package, or bundle. For details see “Displaying the product page” on page 145, “Displaying the bundle page” on page 143, and “Displaying the package page” on page 144 and then clicks Add to wish list . The system adds the product, bundle, or package to the wish list (E1) and then displays the wish list page, as described in “Viewing a wish list” on page 167.
Exception flow	E1: Guest customer tries to add an item to the wish list If a guest customer tries to add an item to the wish list, the system displays the following message: To save items to your wish list, register, and then try again.

Implementation details

Adding items to a wish list uses the following JavaServer Pages (JSP) files:

- CachedItemDisplay.jsp
- CachedProductItemDisplay.jsp
- CachedProductOnlyDisplay.jsp
- ItemDisplay.jsp
- OrderItemDisplay.jsp
- ProductDisplay.jsp
- ProductItemDisplay.jsp
- ProductOnlyDisplay.jsp
- WishListDisplay.jsp

Adding items to an order

Actor	Customer
Precondition	The customer is logged on to the store
Basic flow	<ol style="list-style-type: none">1. The customer clicks Browse Catalog from the Home page. The Main Category page displays.2. The customer selects a category. The category display page displays.3. The system displays a page with the top category chosen and the main categories beneath it.4. The customer selects a product. The Product Display page displays.5. The system displays a page with items within the product line chosen.6. The customer selects an item. The Item Display page displays.7. The system displays a page with the item and all of its attributes.8. The customer selects a contract and clicks Add to Order. If the customer does not select a contract, the item is added under the contract with the lowest price.9. The Your Order page displays.
Postcondition	The system displays a page with the new item added to the order. If the item is the first one chosen, a new order is created with the item in it.

Implementation details

Adding items to an order uses the following JavaServer Pages (JSP) files:

- CachedItemDisplay.jsp
- CachedProductItemDisplay.jsp
- CachedProductOnlyDisplay.jsp
- CachedStoreCatalogDisplay.jsp
- CachedSubCategoriesDisplay.jsp
- CachedTopCategoriesDisplay.jsp
- CategoriesSetup.jsp
- ItemDisplay.jsp
- OrderItemDisplay.jsp
- ProductDisplay.jsp
- ProductItemDisplay.jsp
- ProductOnlyDisplay.jsp
- StoreCatalogDisplay.jsp
- SubCategoriesDisplay.jsp
- TopCategoriesDisplay.jsp

Changing personal information

Customers can change personal information, such as e-mail address and password.

Actor	Customer
Main flow	<p>The customer clicks Change Personal Information (E1). The system displays the Change personal information page, which includes the following fields:</p> <ul style="list-style-type: none">• First name, pre-filled with the current first name• Last name, pre-filled with the current last name• E-mail , pre-filled with the current e-mail address• Password <p>The password field can be left empty. If the customer chooses to leave the password field empty, the system does not modify the current password and the system leaves the previous password unchanged.</p> <ul style="list-style-type: none">• Verify password• Preferred language• Preferred currency• Age, pre-filled with the age• Gender, pre-filled with the gender• Interest e-mail <p>The customer alters the fields as needed.</p> <p>The customer then selects Submit, and the system updates the e-mail address and password (E2).</p>
Alternate flow	None
Exception flows	<p>The password entered must conform to the following rules:</p> <ul style="list-style-type: none">be at least 6 characterscontain at least one digitcontain at least 1 lettermust not contain the same character 5 times in total, or 4 consecutive times <p>or an error message is displayed. An error message is also displayed if the password is the same as the e-mail address, or if the password is the same as the previous password.</p> <p>E1: Customer not logged in Customers must be logged in to change their e-mail addresses and passwords. If customers are not logged in, the system prompts the customers to log in using the procedure described in the “Logging on by a Registered Customer” on page 148 use case. This use case then resumes from the beginning.</p> <p>E2: E-mail already exists for a different customer The entered e-mail address cannot match the e-mail address of any other existing customer. If the system finds a match it prompts the customer to select a different e-mail address. The use case resumes from the beginning.</p> <p>E3: Can not verify password The password entered must match the password specified as the verify password. Otherwise, the system displays the following error message: Password does not match the verify password, please try again.</p> <p>The use case resumes from the beginning.</p>

Implementation details

Changing personal information uses the following JavaServer Pages (JSP) files:

- MyAccountDisplay.jsp
- UserRegistrationSetup.jsp
- UserRegistrationUpdateForm.jsp

Checking out the shopping cart

Check out means to submit payment information for merchandise in the shopping cart.

Actor	Customer
Main flow	<ol style="list-style-type: none"> 1. The customer clicks Checkout to initiate the use case. The system displays the Billing Address page. If the customer has at least one address in the address book, the system prompts the customer to select one of the addresses from the address book as the billing address. The customer can select an address or click Create new address. If the customer clicks Create new address, a new address is added using the "Adding a new address" on page 134 use case. 2. The customer selects an address. The system sets up the selected address as the billing address for the order. If the customer does not have any addresses in the address book, A2 Enter Billing Address is performed. 3. The system displays the Shipping Address page. A list of addresses in the address book displays. If the customer finds the appropriate shipping address in the list, the customer selects the address. Otherwise, the customer clicks Create New Address and adds a new address using the "Adding a new address" on page 134 use case. The system sets up the selected address as the shipping address of the order. 4. The system displays a list of shipping methods appropriate for the shipping address. The following information displays for each shipping method: <ul style="list-style-type: none"> • Short description • Cost of shipping: <ul style="list-style-type: none"> – Fixed shipping cost per order – Shipping cost per item ordered • Approximate delivery time 5. The customer selects the shipping method. The system sets the selected method as the shipping method of the order. The shipping address and shipping method are specified for each order item. For each product in the order the system displays: <ul style="list-style-type: none"> • Quantity • Short product description • Attribute values • Remove button (to remove the item from the order) 6. The customer then selects the shipping preference for the order. As well as displaying the product information, the system also displays the details of the order information. For each item in the order the system displays: <ul style="list-style-type: none"> • Short product description • Attribute values • Quantity • Unit price • Total price • Shipping address • Shipping method • The system also displays: <ul style="list-style-type: none"> – Billing address – Subtotal (total cost of all products ordered) – Discounts, if any – Total tax, if any (multiple taxes should be displayed separately) – Duties, if any – Shipping cost including the shipping method – Grand total (what the customer will be charged) 7. The system prompts the user to enter the following credit card information. <ul style="list-style-type: none"> • Credit card type (for example, Visa(R) or MasterCard(R)) • Card number • Expiration month • Expiration year 8. The customer types credit card information and clicks Order Now. The system stores the payment information (E1) and displays a confirmation page with the following information about the order: <ul style="list-style-type: none"> • Order number • Sub total (total cost of all products ordered) • Total tax • Shipping • Discounts, if any • Grand total 9. The customer prints the page for future reference.
Alternate flow	<p>A1: Add billing address The customer is prompted to add a new address. The customer types an address using the "Adding a new address" on page 134 use case. The system sets up the new address as the billing address for the order. The use case continues.</p>

Exception flow	<p>E1: Invalid credit card number The system checks the validity of the credit card number. If the check fails, the system displays an error message stating: Invalid credit card number</p> <p>and prompts the customer to re-enter the information. The use case continues.</p> <p>E2: Credit card expired If the expiry date of the credit card is earlier than the current date, the system displays an error message. The use case resumes from the beginning.</p>
----------------	--

Implementation details

Checking out a shopping cart uses the following JavaServer Pages (JSP) files:

- AddressBookForm.jsp
- AddressForm.jsp
- BillingAddressDisplay.jsp
- MultiOrderProcess.jsp
- MultipleShippingAddressDisplay.jsp
- MultipleShippingMethodDisplay.jsp
- OrderConfirmationDisplay.jsp
- OrderItemDisplay.jsp
- OrderOKSetup.jsp
- ProductAvailabilityDisplay.jsp
- ProductAvailabilitySetup.jsp
- StandardCreditCard.jsp

Creating a quick checkout profile

In order for Registered Customers to use the quick checkout feature, they must first create a quick checkout profile.

Actor	Registered Customer
Main flow	<p>The customer clicks My Account, then Create or update profile to initiate the use case.</p> <p>The system displays the quick checkout profile page.</p> <ol style="list-style-type: none">1. Billing address, which includes the following fields:<ul style="list-style-type: none">• First name• Last name• Street address• City• State/Province• ZIP code/Postal code• Country/Region• Phone number<p>Note: If a default billing address already exists, the system pre-fills the page.</p>2. Shipping address, which includes the following fields:<ul style="list-style-type: none">• First name• Last name• Street address• City• State/Province• ZIP code/Postal code• Country/Region• Phone number<p>Note: If a default shipping address already exists, the system pre-fills the page. The shipping address also includes a same as billing address above check box. If the check box is selected, the shipping address will be the same as the billing address.</p>3. Available shipping methods. Each shipping method includes the following:<ul style="list-style-type: none">• Name of the shipping method• Details of the shipping cost• Delivery time<p>Note: The last shipping method selected by the customer is selected by default.</p>4. Payment information, including:<ul style="list-style-type: none">• Credit card type• Card number• Expiration month• Expiration year <p>The customer types or edits the information, then clicks Submit. The system saves the information, adding the address information as described in “Adding a new address” on page 134. The system then displays the My Account page.</p>

Implementation details

Creating a quick checkout profile uses the following JavaServer Pages (JSP) files:

- MyAccountDisplay.jsp
- QuickCheckoutProfileForm.jsp
- QuickCheckoutUpdateProfile.jsp

Displaying cross-sells in a category page

Purpose	Category page displays a list of subcategories and products. If cross-sell is setup, it also shows the cross-sell category.
Actor	Customer
Precondition	Cross-sell is setup for that category. A subcategory page displays.
Postcondition	Cross-Sell displays in the subcategory page
Main flow	<ol style="list-style-type: none">1. The customer clicks on a category link.2. System retrieves from the database the sub-categories and products within the selected category, and displays the information of those categories and products using the Category display use case. If Cross-Sell is setup for that category, in the category page, the name of each cross-sell category is listed. Also, there is a sentence indicating that these are cross-sell category. The name will be a link to the corresponding sub-category display page.
Alternate flow	The customer clicks on a sub-category with no Cross-Sell being setup
Exception flow	None

Implementation details

Displaying cross-sells in a category page uses the following JavaServer Pages (JSP) files:

- CachedProductOnlyDisplay.jsp
- CachedStoreCatalogDisplay.jsp
- CachedSubCategoriesDisplay.jsp
- CachedTopCategoriesDisplay.jsp
- CategoriesSetup.jsp
- ProductDisplay.jsp
- ProductOnlyDisplay.jsp
- StoreCatalogDisplay.jsp
- SubCategoriesDisplay.jsp
- TopCategoriesDisplay.jsp

Displaying the bundle page

Purpose	The bundle page displays more details about the products or items that compose a bundle.
Actor	Customer
Main flow	<p>The customer selects a bundle in the category page. The system then retrieves the bundle information from the database and then displays a page with detailed information about the selected bundle. The bundle page displays the following information:</p> <ul style="list-style-type: none">• Short description• Long description• Full-sized image of the bundle• Images of the products that compose the bundle• Price for each product in the bundle• A list of attributes if the products that compose the bundle have variations, and a list of values for each attribute. <p>The customer then selects the appropriate values for each attribute, selects the quantity, and clicks Add to shopping cart. The system adds the bundle to the shopping cart (E-1), and displays the Shopping Cart page as described in the “Displaying the shopping cart” on page 146.</p> <p>Note: The default quantity is 1. Customers can modify the quantity after the bundle has been added to the shopping cart.</p> <p>If the customer clicks Add to wish list the system adds the bundle to the wish list and displays the wish list as described in the “Adding an item to the wish list” on page 136 use case.</p> <p>Note: The Add to wish list button is only available if the wish list feature is enabled in the store, for more information see Chapter 2, “Configuring the Express Store,” on page 11.</p>
Alternate flow	None
Exception flow	None

Implementation details

Displaying bundle pages uses the following JavaServer Pages (JSP) files:

- BundleDisplay.jsp
- CachedBundleDisplay.jsp
- CachedSubCategoriesDisplay.jsp
- CategoriesSetup.jsp
- OrderItemDisplay.jsp
- SubCategoriesDisplay.jsp
- WishListDisplay.jsp

Displaying the package page

Purpose	The package page displays more details about the products or items that compose the package. Note: Within the Product Management tools, a Package object is referred to as a prebuilt kit. Both terms are considered synonymous.
Actor	Customer
Main flow	<p>The customer selects a package in the category page. The system retrieves the package information from the database and then displays a page with detailed information about the selected package. The package page displays the following information:</p> <ul style="list-style-type: none">• Short description• Long description• Images of the products that compose the package• Price for the package• A list of attributes, and a list of values for each attribute <p>The customer then selects the appropriate values for each attribute, and the quantity, and clicks Add to shopping cart. The system adds the package to the shopping cart (E-1), and displays the Shopping Cart page as described in the “Displaying the shopping cart” on page 146. The default quantity is one.</p> <p>Note: Since all items in a package cannot be sold separately, a package counts as one item in the shopping cart.</p> <p>The customer also has the option of clicking Add to wish list. If the customer does so, the system adds the package to the wish list and displays the wish list.</p> <p>Note: The Add to wish list button is only available if the wish list feature is enabled in the store, for more information see Chapter 2, “Configuring the Express Store,” on page 11.</p>
Alternate flow	None
Exception flow	<p>E1: Attribute is not selected</p> <p>For products with a variety of attributes, the customer must select a value for each attribute. If the customer does not select a value, the system selects a default value and adds the package to the shopping cart or wish list.</p>

Implementation details

Displaying package pages uses the following JavaServer Pages (JSP) files:

- CachedPackageDisplay.jsp
- CachedSubCategoriesDisplay.jsp
- CategoriesSetup.jsp
- OrderItemDisplay.jsp
- PackageDisplay.jsp
- SubCategoriesDisplay.jsp
- WishListDisplay.jsp

Displaying the product page

Purpose	The product page displays more details about a product.
Actor	Customer
Main flow	<p>The customer selects a product in the category page. The system then retrieves the product information from the database and displays a page with detailed information about the selected product. The Product page displays the following information:</p> <ul style="list-style-type: none">• Product name• Detailed description• Price• Full-sized image of the product• A list of attributes and a list of values for each attribute <p>The customer then selects the appropriate values for each attribute and clicks Add to shopping cart. The system adds the selected item in the shop cart (E1), and displays the shopping cart page as described in the “Displaying the shopping cart” on page 146. In the Express Store, the customer can specify the quantity to add to the shopping cart. The default quantity is 1.</p> <p>If the customer clicks Add to wish list the system adds the package to the wish list and displays the wish list as described in “Viewing a wish list” on page 167.</p> <p>Note: The Add to wish list button is only available if the wish list feature is enabled in the store, for more information see Chapter 2, “Configuring the Express Store,” on page 11.</p>
Alternate flow	None
Exception flow	None

Implementation details

Displaying product pages uses the following JavaServer Pages (JSP) files:

- CachedProductOnlyDisplay.jsp
- CachedSubCategoriesDisplay.jsp
- CategoriesSetup.jsp
- OrderItemDisplay.jsp
- ProductDisplay.jsp
- ProductOnlyDisplay.jsp
- SubCategoriesDisplay.jsp
- WishListDisplay.jsp

Displaying the shopping cart

Purpose	Customers can view and edit the products they have selected to purchase through the shopping cart.
Actor	Customer
Main flow	<p>The customer clicks Shopping cart. The system then displays a page showing the contents of the shopping cart. A list of products in the cart displays (E1). For each such product the following information displays.</p> <ul style="list-style-type: none"> • Quantity • Short product description • Attribute name and attribute value pairs of the item, such as size and large. • Unit price • Total price <p>The system also displays the total cost of the order before tax and shipping.</p> <p>If the customer clicks:</p> <p>Update totals A1: Update totals is performed.</p> <p>Remove A2: Remove an item from the cart is performed.</p> <p>Checkout A3: Update cart and checkout is performed.</p> <p>Return to shopping The system displays the same page from where the customer came to the shopping cart page. The use case terminates.</p> <p>Quick checkout The system displays the Quick checkout: Order summary page as described in “Using quick checkout” on page 162.</p>
Sub flows	<p>A1: Update totals To update the quantity of any item, the customer types the new quantity in the text box, then clicks Update totals. The system updates the quantity of the items in the shop cart. If the new quantity of an item is 0, the item is removed from the cart.</p> <p>A2: Remove an item from the cart To delete an item from the cart, the customer clicks Remove item for the item. The system removes the item from the cart.</p> <p>A3: Update cart and checkout The system updates the quantity of the items in the shop cart and initiates the checkout process. Refer to “Checking out the shopping cart” on page 139 use case.</p>
Exception flow	<p>E1: Empty shopping cart If there are no items in the shopping cart, the system displays the following message: Your cart is empty.</p> <p>If an invalid number is entered into the quantity field such as a character, the shopping cart page will be re-displayed with an error message.</p>

Implementation details

Displaying a shopping cart uses the following JavaServer Pages (JSP) files:

- EmptyShopCartDisplay.jsp
- OrderItemDisplay.jsp

Editing an address book

Purpose	Customers can edit addresses in their address book.
Actor	Customer
Main flow	<p>The customer selects an address from address book to edit. The system retrieves and displays the selected address details:</p> <ul style="list-style-type: none">• Nickname• First name• Last name• Street address• City• State or Province• ZIP code or Postal Code• Country or Region• Phone number <p>The customer makes the desired changes and clicks Submit. The system updates the address (E1).</p>
Alternate flow	None
Error flow	<p>E1: The system notifies the customer of missing mandatory information, and requests the missing information which can include the following:</p> <ul style="list-style-type: none">• First name• Last name• Street address• City• State or Province• ZIP code or Postal Code• Country or Region <p>If the nickname already exists in the address book, an error message displays.</p> <p>Customer types missing mandatory information.</p>

Implementation details

Editing an address book uses the following JavaServer Pages (JSP) files:

- AddressBookForm.jsp
- AddressForm.jsp

Logging on by a Registered Customer

Purpose	This use case describes the functional flow of events are when a Registered Customer logs on to a store.
Actor	Registered Customer
Preconditions	The customer has previously registered with the store, and accesses the store using a supported browser.
Main flow	<ol style="list-style-type: none">1. The system displays the Logon page.2. The customer selects a shopping language and types their user ID and password in the User ID and Password fields.3. The customer clicks Submit and the information is submitted. The store Home page displays.
Alternate flow	<p>The customer forgets their password and would like to reset their password.</p> <ol style="list-style-type: none">1. The customer selects the Forgot Your Password? link.2. The system displays the Forgot Password page.3. The customer types their User ID in the appropriate field.4. The customer clicks Send My Password and the information is submitted.5. The system sends the password to the customers e-mail address.6. A confirmation message displays.7. The customer can transfer to the Logon page after they have received their password.
Exception flow	The customers e-mail address does not match with those in the system. The system displays an error message and prompts the customer to enter another user ID.

Implementation details

Logging on by a Registered Customer uses the following JavaServer Pages (JSP) files:

- AccountDisplay.jsp
- LogonSetup.jsp
- MyAccountDisplay.jsp
- PasswordResetDisplay.jsp
- PasswordResetForm.jsp
- PasswordUpdateForm.jsp

Logon

Actor	Customer
Main flow	<ol style="list-style-type: none">1. The customer types their correct e-mail address (username) and password on the logon page and clicks Go.2. The customer is logged into the store and is on the home page (StoreCatalogDisplay).
Alternate flow	<ol style="list-style-type: none">1. The customer is on the logon page but forgets their password. The Actor clicks Forgot my password.2. The Forgot your password page displays.3. The customer types their User ID (e-mail address) in the field and clicks Send my password4. The Forgot your password page is displayed again, and a message appears telling the user their password was sent to them. A link appears so the user can go back to the logon page.
Exception flow	<ol style="list-style-type: none">1. The customer types an incorrect e-mail address (username) and password on the login page and clicks Go.2. The logon page is displayed again, with an error message stating that their username or password is incorrect (it should not say which one was incorrect)
Exception flow2	<ol style="list-style-type: none">1. The customer arrives at the logon page but forgets their password. The Actor clicks Forgot my password .2. The forgot your password page displays.3. The customer types an incorrect username (e-mail address) in the field and clicks Send my password.4. The forgot your password page displays again and tells the user that there is nobody registered with that username.

Implementation details

Supporting login uses the following JavaServer Pages (JSP) files:

- RestrictedSiteAccountDisplay.jsp
- RestrictedSitePasswordResetForm.jsp
- UseLogonRouter.jsp

Managing an account

Purpose	Customers manage their account through the account pages.
Actor	Registered Customer
Main flow	<p>The customer selects My Account. The system then displays the My Account page with the following options:</p> <ul style="list-style-type: none"> • Change personal information • Edit my address book • Create or update Quick Checkout Profile • View wish list • View orders <p>If the customer selects:</p> <p>Change personal information alternate flow A1: Change e-mail and password is performed.</p> <p>Edit my address book alternate flow A2: Edit address book is performed</p> <p>Create or update profile The Quick Checkout Profile page displays, as described in the “Creating a quick checkout profile” on page 141.</p> <p>View wish list the Wish list page displays, as described in the “Viewing a wish list” on page 167. Note: The Add to wish list button is only available if the wish list feature is enabled in the store, for more information see Chapter 2, “Configuring the Express Store,” on page 11.</p> <p>View orders The Order Status page displays, as described in the “Viewing orders” on page 169.</p>
Alternate flow	<p>A1: Edit e-mail and password The system prompts the customer to modify the e-mail address and password using the procedure described in the “Changing personal information” on page 138 use case. The use case resumes from the beginning.</p> <p>A2: Edit address book The system displays a page that lists all the address already added to the address book. Next to each address are two buttons: Edit and Delete. Below the list of addresses is an Add a new address button.</p> <ul style="list-style-type: none"> • If the customer clicks Delete, the system deletes the corresponding address from the database and then displays a message indicating that the address was deleted properly. The alternate flow resumes from the beginning. • If the customer clicks Edit, the system prompts the customer to edit the address using the procedure described in the Edit Address use case. The system then displays a message confirming that the address was updated properly, and the alternate flow resumes from the beginning. • If the customer clicks Add a new address, the system prompts the customer to enter a new address using the procedure described in the “Adding a new address” on page 134 use case. This alternate flow resumes from the beginning.
Exception flow	None

Implementation details

Managing an account uses the following JavaServer Pages (JSP) files:

- MyAccountDisplay.jsp

Placing a quick order

Actor	Customer
Main flow	<ol style="list-style-type: none">1. The customer types a product SKU in the Quick Orders entry box on the left navigation and clicks Go.2. The shopping cart page displays with the corresponding SKU added to the shopping cart.
Alternate flow	<ol style="list-style-type: none">1. The customer clicks Add multiple items to cart.2. The Add multiple items to cart page displays.3. The customer types multiple SKUs and quantities on this page and clicks Add to cart.4. The shopping cart page displays with the corresponding SKU(s) added to the shopping cart.
Exception flow	<ol style="list-style-type: none">1. The customer types a product SKU that does not exist in the catalog in the Quick Orders entry box on the left navigation and clicks Go.2. The shopping cart page displays with an error message stating that the SKU does not exist in the catalog.
Exception flow 2	<ol style="list-style-type: none">1. The customer clicks Add multiple items to cart.2. The Add multiple items to cart page displays.3. The customer types multiple SKUs (some may exist and others may not exist) and quantities on this page and clicks Add to cart.4. The quick orders page displays again showing the SKU(s) that are not found in the catalog.

Implementation details

Supporting quick orders uses the following JavaServer Pages (JSP) files:

- OrderItemDisplay.jsp
- QuickOrderForm.jsp
- SidebarDisplay.jsp

Registering

Purpose	The registration process allows customers to enter personal information in the database.
Actor	Customer
Main flow	<p>The customer selects Register. The system then displays a page with the following fields:</p> <ul style="list-style-type: none">• E-mail• Password• Verify password• First name• Last name• Preferred language• Preferred currency• Age• Gender <p>If the customer would like to receive e-mail about specials and features, he must check this box to indicate an interest, and then select the interest items listed.</p> <p>The customer types the appropriate information in the above fields, and selects Submit. The system creates a new customer in the system and saves the customers information (E1, E2, E3).</p> <p>The system prompts customers to manage their accounts following the process in the “Managing an account” on page 150 use case.</p>
Alternate flow	None
Exception flow	<p>E1: E-mail address already exists If the e-mail address already exists in the system, the system displays an error message asking the user to enter another e-mail address. The use case resumes from beginning.</p> <p>E2: Missing mandatory fields An error message displays if the following fields are incomplete: E-mail, Password, Verify password, First name, Last name. The use case resumes from beginning.</p> <p>E3: Invalid password If the password does not match the verification password, the system issues a warning.</p> <p>Note: The password entered must be at least six characters, contain at least one digit and one letter, and must not contain the same character five times in total, or four consecutive times, or an error message displays. An error message is also displayed if the password is the same as the e-mail address.</p>

Implementation details

Registration uses the following JavaServer Pages (JSP) files:

- AccountDisplay.jsp
- MyAccountDisplay.jsp
- UserRegistrationAddForm.jsp
- UserRegistrationSetup.jsp

Searching for a specific product

Actor	Customer
Preconditions	The store has enabled search.
Main flow	<ol style="list-style-type: none">1. The customer types in the name of a product in the Search field of the left navigation frame, and clicks Go.2. The system displays the Search Results page.
Alternate flow	<ol style="list-style-type: none">1. The customer clicks Advanced Search on the left navigation frame and the system displays the Advanced Search page.2. The customer searches for products based on words, categories, manufacturer, part number, or SKU by entering information in the appropriate fields. The customer can also select the number of results per page.3. The customer clicks Submit.4. The system displays the Search Results page.

Implementation details

Searching for a specific account uses the following JavaServer Pages (JSP) files:

- AdvancedCatalogSearchForm.jsp
- CatalogSearchResultDisplay.jsp

Setting up an e-mail address to receive e-mail option

Actor	Guest customer
Preconditions	<ul style="list-style-type: none">• The store has enabled the promotional e-mail field.• One default e-mail campaign is set up for the store.• E-mail accounts for inbound and outbound messages are set up in Administration Console.• A guest customer is on the new registration page.
Postconditions	The customer provides an e-mail address.
Main flow	<ol style="list-style-type: none">1. The customer fills in the e-mail field in the new registration page in a@b.c format.2. E-mail address appeared in the form.3. The customer checks "Send me e-mails about your stores promotion's" check box.4. Check box checked.5. The customer fills in other fields in the registration page and click submit.6. Registration is successful.
Alternate flow	<ol style="list-style-type: none">1. Customer fills in the e-mail field in the new registration page in a@b.c format.2. Email address appeared in the form.3. The customer does not check "Send me e-mails about your stores promotion's" check box.4. Check box unchecked.5. The customer does not check "Send me e-mails about your stores promotion's" check box.6. The customer fills in other fields in the registration page and clicks Submit.7. Registration is successful and customer will not receive e-mail newsletters.
Exception flow	None

Implementation details

Setting up an e-mail address to receive e-mail options uses the following JavaServer Pages (JSP) files:

- UserRegistrationAddForm.jsp
- UserRegistrationSetup.jsp

Setting up cross-sells, up-sells, and accessories

Purpose	Customer clicks on Product in either the Home Page or Category pages. It shows the cross-sell, up-sell, or accessory products if they are set up.
Actor	Customer
Preconditions	Cross-sell, up-sell or accessory is setup for that product. A product page displays.
Post condition	Cross-Sell, Up-Sell or Accessory displays in the product page
Main flow	<ol style="list-style-type: none">1. The customer clicks on a product.2. A Product page displays with more details about a product. If Cross-Sell, Up-Sell or Accessory is setup for that product, in the product page, the cross-sell, up-sell or accessory products are listed. The following information displays for each cross-sell, up-sell or accessory product:<ul style="list-style-type: none">• a thumbnail image The thumbnail image and the name will be a link to the corresponding product page• name• offer price• A sentence displays indicating that these products are cross-sell, or up-sell, or accessory.3. The customer clicks on the cross-sell, or up-sell, or accessory product4. If a product is clicked, a product page displays
Alternate flow	Customer clicks on a product with no Cross-Sell, Up-Sell or Accessory being setup
Exception Flow	None

Implementation details

Cross-sells, up-sells, and accessories use the following JavaServer Pages (JSP) files:

- CachedProductOnlyDisplay.jsp
- CachedStoreCatalogDisplay.jsp
- CachedSubCategoriesDisplay.jsp
- CachedTopCategoriesDisplay.jsp
- CategoriesSetup.jsp
- ProductDisplay.jsp
- ProductOnlyDisplay.jsp
- StoreCatalogDisplay.jsp
- SubCategoriesDisplay.jsp
- TopCategoriesDisplay.jsp

Submitting your order

Purpose	This use case describes the flow of events when a Registered Customer submits their order.
Actor	Registered Customer
Preconditions	The customer has at least one item added to the current order.
Postconditions	The order is placed.
Main flow	<ol style="list-style-type: none">1. The system displays the Your Order page.2. The customer selects a payment method and clicks Next.3. The system displays the Billing and Shipping Information page.4. The customer selects a billing address from the Billing address drop-down box and clicks Next.5. The system displays the Order Summary page.6. The customer types their payment information under the Payment information heading, clicks Order Now. The order is submitted.7. The system displays the Order Confirmation page.
Alternate flow	<p>Alternate flow 1 The customer would like to remove an item from their order.</p> <ol style="list-style-type: none">1. The system displays the Your Order page.2. The customer clicks Remove beside the item they would like removed. <p>Alternate flow 2 The customer has clicked Order Now on the Order Summary page and some of the items in the order are not available.</p> <ol style="list-style-type: none">1. The system displays the Product Availability page with two shipping preference options:<ol style="list-style-type: none">a. Option 1: Ship the entire order when all items are available. (date available).b. Option 2: Ship the available items now and ship the rest when they become available. (I understand that I may have to pay another shipping charge at this time, and some discounts may not be available.) (date available)2. The customer chooses a shipping preference option <p>Alternate flow 2a The customer selects Option 1 and clicks Next. The order is not split and all items are shown in the Order Summary page.</p> <p>Alternate flow 2b The customer selects Option 2 and clicks Next. The order is split into available items and not available items, and the information of the two orders is shown on the Order Summary page. The customer can choose to process the available order and continues with the submit your order flow.</p>

Implementation details

Submitting your order uses the following JavaServer Pages (JSP) files:

- AddressBookForm.jsp
- AddressForm.jsp
- BillingAddressDisplay.jsp
- MultiOrderProcess.jsp
- MultipleShippingAddressDisplay.jsp
- MultipleShippingMethodDisplay.jsp
- OrderConfirmationDisplay.jsp
- OrderItemDisplay.jsp
- OrderOKSetup.jsp
- OrderSubmitForm.jsp

- ProductAvailabilityDisplay.jsp
- ProductAvailabilitySetup.jsp
- StandardCreditCard.jsp

Updating personal information

Actor	Registered Customer
Main flow	<ol style="list-style-type: none">1. The system displays the Account page.2. The customer clicks Change Personal Information. The Update Registration page displays.3. The system displays the fields with pre-filled information.4. The customer alters the fields as needed. (If the Password field is left empty the current password is maintained.)5. The customer clicks Update and the information is submitted. The system updates the edited fields.

Implementation details

Updating personal information uses the following JavaServer Pages (JSP) files:

- MyAccountDisplay.jsp
- UserRegistrationSetup.jsp
- UserRegistrationUpdateForm.jsp

Updating receive promotional e-mail option

Purpose	Update personal e-mail address or change interests or unsubscribe.
Actor	Registered Customer
Preconditions	<ul style="list-style-type: none"> • One default e-mail campaign is set up for the store. • A Registered Customer logged in and is in the Change Personal Information page. • The promotional e-mail field is enabled.
Postconditions	A customer updates the e-mail address. If the customer clears the "Send me e-mails about your stores promotion's." box, no e-mail will be sent to this customer when the e-mail activity is scheduled next time.
Main flow	<ol style="list-style-type: none"> 1. The customer checks "Send me e-mails about your stores promotion's" check box. 2. The customer fills in other fields in the page and clicks Submit. 3. Registration update is successful. The customer will receive promotional e-mail.
Alternate flow	<ol style="list-style-type: none"> 1. The customer updates the e-mail field in the Update Personal Information page in a@b.c format. 2. E-mail address appeared in the form. 3. The customer checks "Send me e-mails about your stores promotion's" check box. 4. Check box checked. 5. The customer fills in other fields in the registration page and click submit. 6. Registration update is successful and customer will receive the following e-mail to the new e-mail address. <ol style="list-style-type: none"> 1. Alternate flow 2: customer updates the e-mail field in the Update Personal Information page in a@b.c format. 2. E-mail address appeared in the form. 3. The customer clears "Send me e-mails about your stores promotion's" check box. 4. Check box unchecked. 5. The customer fills in other fields in the page and click submit. 6. Registration update is successful and customer will not receive e-mail when the e-mail activity is scheduled next time.
Exception flow	None

Limitation: Currently, the registration command does not check whether the e-mail format is valid. So, if the e-mail is incorrect, the customer will not receive the e-mail.

Updating the receive personal information uses the following JavaServer Pages (JSP) files:

- UserRegistrationSetup.jsp
- UserRegistrationUpdateForm.jsp

Using a private requisition list

Actor	Customer
Main flow	<ol style="list-style-type: none"> 1. The customer arrives at the store home page. 2. The home page displays. 3. The customer clicks on a Top Category. 4. The top category sub categories are displayed. 5. The customer clicks on a subcategory link. 6. Products in the subcategory are displayed. 7. The customer clicks on a product. 8. The products and product items are displayed. 9. From the Requisition list drop down list, the customer selects New and then Go. 10. The New Requisition List page displays. This page has a text box for the Actor to type a name for the list. Then click Change Name. The first SKU box contains the SKU of the item the used just clicked on. Quantity of this item is set to 1 by default. 11. The Edit Requisition List page displays listing the list that was just created. 12. The customer clicks Place Order. 13. The Shopping cart page displays with the Requisition list items added to the customer's cart. 14. The customer can complete checkout as usual.
Alternate flow	<ol style="list-style-type: none"> 1. The customer arrives at the stores home page. 2. The home page displays. 3. The customer clicks on a Top Category. 4. The top category sub categories are displayed. 5. The customer clicks on a subcategory link. 6. Products in the subcategory are displayed. 7. The customer clicks on a product. 8. The products and product items are displayed. 9. From the drop down list, the customer selects an existing requisition list name and clicksGo. 10. The Edit Requisition List page displays, displaying all lists this user owns. 11. The customer select a list so add to and clicks Update Quantity 12. The Edit Requisition List page displays showing all previous items on this list plus the one just added.

Alternate flow 2	<ol style="list-style-type: none"> 1. The customer arrives at the Edit Requisition List page. 2. The Edit Requisition List page displays. 3. The customer modified the List name and clicks Change Name. 4. The Edit Requisition List page displays. The List name is updated. 5. The customer types a new SKU with no quantity and clicks Add. 6. The new SKU is added to the list with quantity of 1. 7. The customer types a new SKU with quantity 10 and clicks Add. 8. The Edit Requisition List page displays. The new SKU is added to the list with quantity of 10. 9. The customer types a new SKU that does not exist and clicks Add. 10. The Edit Requisition List page displays. A message displays telling the user that the SKU does not exist. 11. The customer changes the quantity of all the items on the list and clicks Update Quantity. 12. The Edit Requisition List page displays. All quantities are updated properly. 13. The customer removes all items on the list by clicking the Remove link beside each item. 14. The Edit Requisition List page displays. It shows the Requisition list name and allows you to enter new SKUs.
Exception flow	<p>One requisition list pages where multiple SKUs can be entered:</p> <ol style="list-style-type: none"> 1. The system should check to make sure all SKUs have a quantity assigned to them. 2. If not, we will assume a quantity of 1. All SKUs must be valid. If an invalid SKU is entered, the list page should be redirected back to and all invalid SKUs should be displayed. <p>If the Edit Requisition List page displays and the user has no lists to add to, a message displays stating: There are no existing requisition lists owned by you. Please go back to the Item Display page and choose New, to add the item to a new requisition list.</p>

Implementation details

Private requisition lists uses the following JavaServer Pages (JSP) files:

- CachedProductOnlyDisplay.jsp
- CachedStoreCatalogDisplay.jsp
- CachedSubCategoriesDisplay.jsp
- CachedTopCategoriesDisplay.jsp
- MyAccountDisplay.jsp
- OrderItemDisplay.jsp
- ProductOnlyDisplay.jsp
- RequisitionListAddToListForm.jsp
- RequisitionListCreateForm.jsp
- RequisitionListDisplay.jsp
- RequisitionListUpdateForm.jsp
- StoreCatalogDisplay.jsp
- SubCategoriesDisplay.jsp
- TopCategoriesDisplay.jsp

Using quick checkout

Purpose	Customers can complete an order without entering shipping and billing information, if they have previously created a quick checkout profile.
Actor	Customer
Main flow	<p>Once a customer has added a product to the shopping cart, they click Quick checkout to initiate the use case (E1, E2). The system then displays the Quick Checkout: Order summary page. For every item in the shopping cart, the following information displays:</p> <ul style="list-style-type: none"> • Quantity • Short description of the product • Unit price • Total price <p>The Quick Checkout: Order summary page also includes the following information about the order:</p> <ul style="list-style-type: none"> • Subtotal • Discounts if any • Total taxes. Multiple taxes will be displayed separately. • Duties if any. • Shipping cost including the shipping method • Grand total • Billing address • Shipping address <p>The system prompts the customer to enter the following payment information:</p> <ul style="list-style-type: none"> • Credit card type • Card number • Expiration month • Expiration year <p>Note: If the customer created credit card information in the Quick checkout profile, as described in “Creating a quick checkout profile” on page 141, the payment information field will be pre-filled.</p> <p>The customer reviews the order summary and clicks Order Now. The system completes the order (E3) and displays an Order confirmation page.</p>
Alternate flow	None
Exception flow	<p>E1: Customer does not have a quick checkout profile The customer must create a quick checkout profile as described in “Creating a quick checkout profile” on page 141, before completing a quick checkout. If the system cannot find the billing address, shipping address, or shipping method, it displays an error message telling the customer to create a quick checkout profile.</p> <p>E2: Customer is not registered If a guest customer tries to use quick checkout, the system displays an error message telling the customer to register and create a quick checkout profile before using the quick checkout option.</p> <p>E3: Invalid credit card information If the credit card number entered in the quick checkout profile is invalid or expired, the system displays an error message.</p>

Implementation details

Quick checkout uses the following JavaServer Pages (JSP) files:

- OrderConfirmationDisplay.jsp
- OrderItemDisplay.jsp
- OrderOKSetup.jsp

- QuickCheckoutForm.jsp
- QuickCheckoutPrepare.jsp
- StandardCreditCard.jsp

Using the Home page

Purpose	The home page acts as the storefront, and brings customers into your store.
Actor	Customer
Main flow	<p>A customer types the store URL in the Web browser. The system then displays the home page.</p> <p>The home page includes the following:</p> <ul style="list-style-type: none">• A navigation bar with links to the following pages:<ul style="list-style-type: none">– Home page– Advanced search– Shopping cart page– Registration page– Account page– Contact information page– Security and privacy page– Help page• Links to primary or top-level categories. For each category the following information displays:<ul style="list-style-type: none">– An image– Links to targeted productsIf customers have specified their gender when registering, products targeted to the customers gender display. If not, a generic set of products displays. For each product the following information displays:<ul style="list-style-type: none">- An image- A short description <p>The customer clicks the image. Then, the system displays the corresponding page (A1, A2).</p>
Alternate flow	<p>A1: Customer selects category When the customer selects a category, the system displays a category page as described in “Viewing a product category” on page 166.</p> <p>A2: Customer selects product When the customer selects a product, the product page displays as described in “Displaying the product page” on page 145.</p>

Implementation details

Displaying a home page uses the following JavaServer Pages (JSP) files:

- CachedStoreCatalogDisplay.jsp
- CachedTopCategoriesDisplay.jsp
- CategoriesSetup.jsp
- StoreCatalogDisplay.jsp
- TopCategoriesDisplay.jsp

Using the shopping total

Actor	Customer
Main flow	<ol style="list-style-type: none">1. The customer arrives at the stores home page as a guest.2. The Shopping total shows zero items in cart the subtotal is 0.3. The customer goes through the catalog pages and adds an item to their cart.4. The shopping cart shows 1 item in the shopping cart and the subtotal should match the item price.5. The customer returns to the home page.6. The home page displays with the correct shopping totals.7. The customer goes through the catalog pages again, following the same path as before.8. The shopping total should have the correct totals on all the catalog pages.9. The customer adds the same item to the shopping cart.10. The shopping total shows 2 items in the shopping cart and the subtotal matches the subtotal on the shopping cart page.
Alternate flow	<ol style="list-style-type: none">1. The customer arrives at the store's home page as a guest.2. The shopping total should show zero items in the shopping cart. The subtotal is 0.3. The customer clicks My Account.4. The Login or Register page displays.5. The customer logs in.6. The My Account page displays and the shopping total shows the number of items in the customer's cart from the past session.
Exception flow	None

Implementation details

Mini shopping uses the following JavaServer Pages (JSP) files:

- CachedStoreCatalogDisplay.jsp
- MiniShopCart.jsp
- MyAccountDisplay.jsp
- SidebarDisplay.jsp
- StoreCatalogDisplay.jsp

Viewing a product category

Purpose	<p>Catalog group pages display a list of subcategories and products. Catalog group pages help customers navigate through the products. The beginning catalog group pages lead to broad areas, and subsequent catalog group pages narrow the search.</p> <p>There are usually three types of catalog group pages:</p> <ul style="list-style-type: none">• Pages that display subcategories within a parent category• Pages that display products within a subcategory• Pages that display both subcategories and products
Actor	Customer
Main flow	<p>Top-level product categories are listed in the Home page. The following information displays for each category and product:</p> <ul style="list-style-type: none">• A thumbnail image (products only)• The name of each category or product• A brief description to identify product.• The price of the product. <p>The customer clicks the thumbnail image or the name. Then, the system displays the corresponding page (A1, A2, A3, A4).</p>
Alternate flow	<p>A1: Customer selects category When the customer selects a category, the system displays another category page by resuming this use case from the beginning.</p> <p>A2: Customer selects product When the customer selects a product, the product page displays as described in “Displaying the product page” on page 145.</p> <p>A3: Customer selects bundle When the customer selects a bundle, the bundle page displays as described in “Displaying the bundle page” on page 143.</p> <p>A4 Customer selects package When the customer selects a package, the package page displays as described in “Displaying the package page” on page 144.</p>
Exception flow	None

Implementation details

Viewing product category uses the following JavaServer Pages (JSP) files:

- CachedStoreCatalogDisplay.jsp
- CachedSubCategoriesDisplay.jsp
- CachedTopCategoriesDisplay.jsp
- CategoriesSetup.jsp
- StoreCatalogDisplay.jsp
- SubCategoriesDisplay.jsp
- TopCategoriesDisplay.jsp

Viewing a wish list

Precondition	The Add to wish list feature is enabled for the store, see Chapter 2, “Configuring the Express Store,” on page 11.
Purpose	The customers can view and edit the products they have added to their wish list.
Actor	Customer
Main flow	<p>Customers click My Account, and then View wish list. The system then displays a page showing the contents of the wish list. A list of products in the wish list displays (E1). For each product, bundle, or package in the wish list, the following information displays.</p> <ul style="list-style-type: none"> • Short product description • Attribute and attribute values of the item • Unit price • Remove link <p>The system also displays the following items in the page:</p> <ul style="list-style-type: none"> • Return to shopping button • Add selected items to shopping cart button. • Send wish list button • E-mail address field (To: / recipient) • E-mail message field • From:/sender Name • E-mail address <p>To delete an item from the wish list, the customer, clicks Remove. The system completes A1, Remove item.</p> <p>To send the wish list to friends and family via e-mail, customers enter the recipients E-mail addresses in the E-mail address field and click Send wish list. The system completes A2, Send wish list. In the Express Store, customers can send an e-mail message using the E-mail message field. Customers need to specify their name and can optionally specify their e-mail address. Registered Customers have the Name field pre-filled.</p> <p>To add items in the wish list to the shopping cart, customers select each item to add to the shopping cart and click Add selected items to shopping cart. The system completes A3, Add to cart.</p>

Alternate flow	<p>A1: Remove item To delete an item from the cart, customers click Remove for the item. The system removes the item from the wish list. The use case resumes from the beginning.</p> <p>A2: Send wish list To send the wish list to friends and family via e-mail, customers enter the recipients e-mail addresses in the E-mail address field. To send the wish list to more than one friend, each e-mail address must be separated with a comma, for example: info@infashion.com, wcs@infashion.com. The customers then click Send wish list. The system composes an e-mail message containing the following information:</p> <ul style="list-style-type: none"> • instructions on how to purchase the wish list items • a link to the wish list, which contains the following information: <ul style="list-style-type: none"> – short description of each product, package or bundle in the wish list – a link to the product, bundle, or package display page – price – Add selected items to shopping cart button <p>Customers can send a personalized e-mail message along with the system message by entering the message in the E-mail message field.</p> <p>A3: Add to cart To add an item in the wish list to the shopping cart, customers select each item to add, then click Add selected items to shopping cart. The system adds all selected items to the shopping cart, then displays the shopping cart, as described in the “Displaying the shopping cart” on page 146</p> <p>Exception flows</p> <p>E1: Empty wish list If there are no items in the wish list, the system displays the following message: Empty wish list. The use case terminates.</p>
-----------------------	--

Implementation details

Viewing a wish list uses the following JavaServer Pages (JSP) files:

- MyAccountDisplay.jsp
- OrderItemDisplay.jsp
- WishListDisplay.jsp
- WishListMessageConfirmationDisplay.jsp

Viewing orders

Purpose	Customers can track and view the status of their order.
Actor	Customer
Main flow	<p>The customer clicks My Account and then View orders. The system displays the Order Status page, which lists all of the orders placed by the customer. If the customer has not placed any orders, E1: No orders found is performed.</p> <p>The following information displays for each order:</p> <ul style="list-style-type: none">• Order number (linked to the Order Details page)• Order date• Order status• Total amount <p>To find out more information about the order, the customer can click the order number. The system displays the Order Details page. The Order Details page displays the following information about each product in the order:</p> <ul style="list-style-type: none">• Quantity• Short product description• The attributes and values for each attribute• Unit price• Total price• Shipping date for the whole order• Tracking ID numbers
Exception flow	<p>E1: No orders found</p> <p>If the system cannot find any orders associated with this customer, the system displays an error message stating:</p> <p>There are no orders found</p> <p>The customer can then click the Home , My Account, or another category.</p>

Implementation details

Viewing orders uses the following JavaServer Pages (JSP) files:

- MyAccountDisplay.jsp
- OrderDetailDisplay.jsp
- OrderStatusDisplay.jsp

Appendix B. Data beans, access beans and commands used in the Express Store

API information such as the data beans, access beans, and commands is in the WebSphere Commerce — Express development environment help. To access this help system see the direction is the section “WebSphere Commerce - Express development environment online help” on page vi.

Note: Before using the help system ensure that you download the latest updates. Refreshed versions of the help are available from the WebSphere Commerce Technical Library page.

Data beans

AddressBookDataBean

The bean representing the address book information of a member. The AddressBookAccessBean will automatically get userId from commandContext. Users can use getAddressList() method to return a list of all addresses belong to this user getAddressBookProperties() method returns a hashtable containing the information in the AddressBook property file.

AddressDataBean

The addressDataBean will try to get the addressId and nickName parameter from request. If addressId is not null, it will find this address using the addressId. If addressId is null, it will find this address using the nickName. Users can also directly call setAddressId(String) and setNickName(String) methods to set addressId and NickName directly, overwriting the values get from request.

AttributeDataBean

This bean represents an attribute of a catalog entry.

AttributeValueDataBean

This bean represents an attribute value of a catalog entry.

BundleDataBean

This bean represents a Bundle.

CalculationCodeDataBean

IBM internal use only.

CalculationCodeListDataBean

Use this DataBean to retrieve the list of CalculationCodeDataBeans that matches the specified set of criteria. Null valued parameters are ignored unless specified otherwise

CatalogDataBean

This data bean represents a Catalog.

CatalogEntryDataBean

This data bean represents a Generic Catalog Entry. Use this databean in place of a more specific databean (for example ProductDataBean, ItemDataBean) when the type of Catalog Entry is not known.

CategoryDataBean

This data bean represents a Category.

CatEntrySearchListDataBean

This bean is used to retrieve product/item information from catalog based on input search criteria. The search criteria may be input using a JSP.

CompositeItemDataBean

This bean represents a composite Item.

CompositeProductDataBean

This bean represents a composite Product.

ControllerCommandInvokerDataBean

This is a SmartDataBean that allows a JSP to invoke a controller command.

CurrencyDescriptionDataBean

IBM internal use only.

DataBeanManager

The DataBean manager is responsible for populating databeans.

EmailUserReceiveDataBean

Used to determine if the user has opted out of receiving e-mail.

ErrorDataBean

The bean is for displaying error information on a JSP.

FormattedMonetaryAmountDataBean

A formatter bean used to format monetary amounts.

InterestItemDataBean

Represents an interest item within a interest list. This bean maps to the table IITEM.

InterestItemListDataBean

The bean represents an interest list for a user. This bean maps to the table IITEMLIST.

ItemDataBean

This data bean represents a catalog entry of type Item.

LanguageDescriptionDataBean

IBM internal use only.

OrderAdjustmentDataBean

IBM internal use only.

orderBean

This is an Entity Bean class with CMP fields

OrderDataBean

This data bean represents an order.

OrderItemAdjustmentDataBean

IBM internal use only.

OrderItemDataBean

This data bean represents an order item.

OrderListDataBean

This DataBean when populated will give a list of Order access beans/databeans for those orders that are of the specified status,user, and/or the specified stores.

OrderPaymentInfoDataBean

IBM internal use only.

PackageDataBean

This data bean represents a Package.

PaymentPolicyListDataBean

This DataBean, when populated, gives a list of PaymentPolicyInfo beans representing Payment Business Policies that the Store supports.

PayStatusListPMDDataBean

This DataBean, when populated, provides the payment state of the specified orders as known by WebSphere Commerce Payments. No information will be returned if the order is not known to the Payments.

PayStatusPMDDataBean

This DataBean, when populated, provides the order state and payment state of the specified order as known by WebSphere Commerce Payments. The order must be one that has been sent to the Payments for processing. No information will be returned if the order is not known to WebSphere Commerce Payments. This DataBean is designed for the use of JavaServer Pages that may be invoked during shopping flow to display the payment status of an order by shopper.

PriceDataBean

A formatter bean used to format prices. This bean also contains support for displaying taxes with the price.

ProductDataBean

This data bean represents a product.

RelatedCategoryDataBean

This bean represents a related Category.

RelatedItemDataBean

This bean represents a related Item.

RelatedProductDataBean

This bean represents a related Product.

RequisitionDataBean

IBM internal use only.

ShippingDataBean

The ShippingDataBean is a special purpose databean designed to estimate the shipping charge of a catalog entry for each of the available shipping modes in the absence of an order. Note: to simply retrieve the list of available shipping modes, it is strongly recommended that the ShippingModeDataBean or the ShippingHelper be used instead. The following input parameters are mandatory: commandContext, catalogEntryId. Other input parameters (e.g. addressId) are optional, but will improve the accuracy of the shipping charge estimates. The methods getShipModeIds() and getShipModes() will return the list of shipping modes that return non-null shipping charge estimates after the databean has been populated. The method getShippingCharges() will return the list of shipping charge estimates corresponding to the list of shipping modes returned by getShippingModeIds() or getShippingModes().

ShippingModeDataBean

IBM internal use only.

ShippingModeDescriptionDataBean

IBM internal use only.

StoreDataBean

This data bean represents a Store

SupportedLanguageDataBean

Gets the supported language data from the database.

UsablePaymentTCListDataBean

This DataBean, when populated, gives a list of PaymentTCInfo bean representing Payment Terms and Conditions that could be used to pay for the order specified as input parameter before the DataBean is activated.

UserRegistrationDataBean

The bean representing the registration information of a member.

Access beans

AddressAccessBean

This table stores the addresses of members in the WebSphere Commerce system. The addresses can be the members own addresses or for their friends, associates, or clients, and so on. Some columns here replace columns used in previous versions.

AttributeAccessBean

This table holds product Attributes which are used for descriptive or SKU resolution purposes.

CatalogAccessBean

This table holds the information related to a Catalog.

CatalogEntryAccessBean

This table hold the information related to a CatalogEntry. Examples of CatalogEntries include Products, Items, Packages, and Bundles.

CatalogEntryDescriptionAccessBean

This table holds language-dependent information related to a CatalogEntry.

DemographicsAccessBean

This table stores demographics information for users.

ItemAccessBean

This table hold the information related to a CatalogEntry. Examples of CatalogEntries include Products, Items, Packages, and Bundles.

OrderAccessBean

Each row in this table represents an Order in a Store.

OrderItemAccessBean

Each row of this table represents an OrderItem in an Order.

OrderPaymentInfoAccessBean

Each row in this table holds a name-value pair representing payment information for a particular Order.

PolicyAccountAccessBean

This table stores the account policies.

PolicyPasswordAccessBean

This table stores password policies.

StoreAccessBean

Each row of this table represents a Store. A Store is a StoreEntity.

StoreAddressAccessBean

Each row of this table represents a static address. StoreEntities,

FulfillmentCenters, and Vendors can have static addresses. Static addresses can be staged using the staging server.

StoreEntityDescriptionAccessBean

This table contains language-dependent information about a StoreEntity.

UserAccessBean

This table contains all users of the WebSphere Commerce system: registered users, guest users, and generic users.

Commands

Most commands used in the Express Store have a URL interface. For detailed information about these commands refer to the topic "Commands issued from the URL line" in the online help.

AddressAdd

This command adds a new address entry for a user or organization.

AddressDelete

This command deletes an address for a user.

CategoryDisplay

This command displays a category within a catalog.

InterestItemAdd

This site level command adds one or more products, or items, or both, to one or more interest item lists.

InterestItemDelete

InterestItemDisplay

This site level command deletes catalog entries from one or more interest item lists.

Logoff Log off a registered user.

Logon Log a registered user on to a store or site.

OrderDisplay

This command displays the contents of the specified orders. If order reference numbers are not specified, this command will display the list of orders that match input parameters.

OrderItemAdd

This command adds items or products to the list of items that are to be shipped.

OrderItemDelete

OrderItemDisplay

This command deletes an order item or product from a pending order.

OrderItemMove

This command moves OrderItems in one order to another order.

OrderItemUpdate

This command can do all that OrderItemAdd command can do (that is, add products and items into one or more orders into the order list). It can also update OrderItems in an existing order.

OrderList

This command displays a list of the customer's orders whether they have

been prepared or not. This command provides a summary of the customer's orders. It does not show tax, shipping, or discounts.

OrderPrepare

This command prepares an order by determining prices, discounts, shipping charges, and taxes for an order. If an order reference number is not specified, all current pending orders will be prepared for the current customer at the given store.

OrderProcess

This command submits an order. The order must have been locked by OrderPrepare. Once the OrderProcess command begins running, the order cannot be cancelled with OrderCancel.

OrderProfileUpdate

This command creates or updates a customer's default billing and shipping addresses, shipping mode and payment information. Order profiles can be stored at store level, to provide defaults for use within a store, and at store group level, to provide defaults for use within all stores in the store group.

ProductDisplay

This command displays a catalog entry, which consists of a single item or all the items contained within a product, package or bundle.

RequisitionListCopy

This command copies an existing order or requisition list to a new requisition list.

RequisitionListCreate

This command creates an empty requisition list.

RequisitionListDelete

This command deletes one or more requisition lists.

RequisitionListDisplay

This command displays one or more requisition lists.

RequisitionListItemUpdate

This command adds, updates, and deletes items in a requisition list.

RequisitionListUpdate

This command changes a requisition list type from private to shared, or from shared to private. It also updates the billing and payment information.

ResetPassword

Resets or updates passwords in the database. This command can be used by:

- Registered users and administrators, to update their own passwords.
- Registered users who wish to log on but have forgotten their password, to use this command to reset their password without logging in. Subsequently, a randomly generated password is e-mailed to the guest user.

StoreCatalogDisplay

This command displays all the catalogs for a given store.

TopCategoriesDisplay

This command displays the root categories for a given catalog.

UserRegistrationAdd

This command registers a guest user. After the user is registered, the user

automatically logs onto the WebSphere Commerce system. Information for new users are stored in the MEMBER, USERS, USERREG, MBRREL, USERPROF, BUSPROF, USERDEMO, and ADDRESS database tables.

UserRegistrationUpdate

This command updates the registration record for a registered user. To run this command, the user must first log onto the WebSphere Commerce system as registered user. If the current user is a guest user, this command will call the UserRegistrationAdd command to register a new user. Information for users are stored in the MEMBER, USERS, USERREG, MBRREL, USERPROF, BUSPROF, USERDEMO, and ADDRESS database tables.

Appendix C. Publishing a store to a production environment

Publishing the Express Store to a WebSphere Commerce Server

- ___ Step 1. Ensure that you have Site Administrator access.
- ___ Step 2. Ensure that the database is started..

► Linux

- a. Ensure that the database is started.
 - 1) Start a command line session .
 - 2) Issue the following command: `ps -ef | grep db2`
 - 3) Review the output of the command and ensure the following processes are listed:

```
db2tpcm  
db2srv1st  
db2ipccm  
db2spmr  
db2sysc  
db2wdog  
db2tcpdm  
db2gds  
db2tpcm  
db2sysc  
db2ipccm  
db2resyn  
db2gds  
db2wdog  
db2spm1w
```

If any of these processes are not listed after you issue the command, stop and restart DB2 Universal Database according to the instructions provided in the DB2 documentation.

► Windows

- a. Select **Start > Settings > Control Panel > Administrative Tools > Computer Management**.
- b. In the Computer Management console, expand **Services and Applications** and select **Services**.
- c. Ensure that the following services have a status of **Started**:
 - DB2 - DB2
 - DB2 - DB2CTLSV
 - DB2 - DB2DAS00
 - DB2 - JDBC Applet Server
 - DB2 License Server
 - DB2 Security Server
 - Warehouse logger
 - Warehouse server
- d. If any of these services are not started, right-click the service name and select **Start** from the pop-up menu to start the service. Ensure that you start the services in the order they are listed above. Starting the services in a different order may prevent DB2 Universal Database from starting correctly.

___ Step 3. Ensure that the Web server is started. Open a Web browser and go the following URLs:

```
http://web_server_host_name
https://web_server_host_name
http://web_server_host_name:5432
https://web_server_host_name:5433
https://web_server_host_name:8000
https://web_server_host_name:8002
https://web_server_host_name:8004
```

___ Step 4. Ensure that WebSphere Commerce is started.

a. Open the following log file in a text editor:

```
WAS_installdir/logs/WC_instance_name/SystemOut.log
```

where *instance_name* is the name of the WebSphere Commerce instance.

b. Ensure that the log file does not contain any Java exceptions.

c. Ensure that the following text appears near the end of the log file:

```
WSVR0221I: Application started: WC_instance_name
.
.
WSVR0001I: Server WC_instance_name open for e-business
```

where *instance_name* is the name of the WebSphere Commerce instance.

Note: Some text that appears in the log file has been omitted to indicate the relevant messages clearly.

___ Step 5. If you will use WebSphere Commerce Payments, ensure that WebSphere Commerce Payments is started.

a. Open the following log file in a text editor:

```
WAS_installdir/logs/instance_name_Commerce_Payments_Server/SystemOut.log
```

where *instance_name* is the name of the WebSphere Commerce Payments application server.

b. Ensure that the following text appears near the end of the log file:

```
CEPFW0702: IBM WebSphere Commerce Payments has started successfully.
.
WSVR0221I: Application started: instance_name_Commerce_Payments_App
.
WSVR0001I: Server instance_name_Commerce_Payments_Server
open for e-business
```

where *instance_name* is the name of the WebSphere Commerce Payments instance.

___ Step 6. Prepare the database for publish.

a. Open a DB2 command window.

b. To set attribute values to volatile, run the following statements:

```
db2 connect to databasename
db2 alter table catentdesc volatile
db2 alter table attribute volatile
db2 alter table attrvalue volatile
```

where *databasename* is the name of the database instance.

c. To increase the number of primary logs, run the following statement:

db2 update db cfg for *dbname* using LOGPRIMARY 20

where 20 is the number of primary logs (this number may be different for your site). Increasing the value of the LOGPRIMARY parameter increases your space requirement. For more information, see the *DB2 Administration Guide*.

- d. Close the DB2 command window.
- e. Disable summary tables.

Summary tables are updated during publish. If summary tables are not disabled, a database transaction log overflow could occur. To disable the summary tables, do the following:

- 1) Open Configuration Manager.
 - a) Ensure that the IBM WC Configuration Manager server process is running by doing the following:
 - i. Open the Computer Management console by selecting **Start > Settings > Control Panel > Administrative Tools > Computer Management**.
 - ii. Expand **Services and Applications** and select **Services**.
 - iii. Check that the *IBM WC Configuration Manager server* service has a status of Started.
 - b) Select **Start > IBM WebSphere Commerce > Configuration**.
 - c) In the Configuration Authentication window, enter the Configuration Manager user ID and password and click **OK**.
- 2) In Configuration Manager, select **WebSphere Commerce > Node > Instance List > instance_name > Search Configuration**.
- 3) Select the **Schema Optimization** tab.
- 4) Clear all check-boxes; then click **Apply**.

- f. If you will use WebSphere Commerce Payments, go to Step 7; otherwise close Configuration Manager and go to Step 8.

- ___ Step 7. If you will use WebSphere Commerce Payments, ensure that appropriate payment methods are available.
 - a. Select WebSphere Commerce, then the WebSphere Commerce node; then select **Payments, Instance list**, then the *instancename*.
 - b. From the payment *instancename* folder, select **Instance Properties**, then **Cassettes**. The *Cassettes-instancename* page displays, listing the Available and Selected Cassettes.
 - c. Ensure that the cassette for the payment method is selected. For example, ensure that the OfflineCard Cassette is in the Selected list.
 - d. If you have completed an action, click **Apply**.

- ___ Step 8. (Optional) Back up your WebSphere Commerce database. This will allow you to restore your pre-published version, if necessary.

- ___ Step 9. Publish the Express Store.

- a. Open the Administration Console:



`https://host_name:8002/adminconsole`

where *host_name* is the fully qualified TCP/IP name of your WebSphere Commerce Server.

- b. From the Logon page, type your Administration Console logon user name and password and click **Log On**.

- c. From the Administration Console Site/Store Selection page, select the **site** radio button and click **OK**.
 - d. From the **Store Archives** menu, select **Publish**. The Store Archives page displays. This page lists the store archives files available for publish.
 - e. Select Express Store from the list. Click **Next** The Parameters page displays.
 - f. Accept the default values in the parameters field; then click **Next**. The Summary page displays, listing the store archive and parameters selected, as well as the location to which the store archive will be published.
 - g. Review the summary information; then click **Finish**. A confirmation dialog displays, telling you the number of the publishing job. Note the job number for future reference. Click **Ok**. The Publish Job Status page displays. The Publish status column shows the status of the publish job. Click **Refresh** to update the status.
 - h. Review the publishing details.
 - i. If the store archive published successfully, you can launch the store. To view and test your store, Click **Launch Store**. A dialog box displays, asking you for the Web Application context. Accept the default. Click **Ok**.
- ___ Step 10. Compile WebSphere Commerce JSP pages. Pre-compiling the JavaServer Pages (JSP) files in your store significantly reduce the time needed to load each store page.

- a. Start a command line session.
- b. Change to the following directory:

WAS_installdir/bin

- c. Run the following command:

```
JspBatchCompiler.bat -enterpriseapp.name "WC_instance_name"
-cell.name "hostname" -node.name "hostname"
-server.name "WC_instance_name"
```

where:

instance_name

The name of the WebSphere Commerce instance for which you are compiling the JSP files.

hostname

The short host name of the machine on which you are compiling the JSP files.

Several errors may be logged when you perform these compiles. These errors may be safely ignored.

- ___ Step 11. If you will use WebSphere Commerce Payments, compile all of the WebSphere Commerce Payments JSP files for your site. run the following command:

```
JspBatchCompiler.bat -enterpriseapp.name
"instance_name_Commerce_Payments_App"
-cell.name "hostname" -node.name "hostname"
-server.name "instance_name_Commerce_Payments_Server"
```

where *instance_name* is the name of the WebSphere Commerce Payments instance for which you are compiling the JSP files.

___ Step 12. Enable caching.

- a. Copy the file

`WC_installdir/samples/dynacache/ExpressStore/cachespec.xml`

to the directory:

`WAS_installdir/installedApps/host_name/WC_instanceName.ear/
Stores.war/WEB-INF`

This file contains the caching rules for shopping events at the Express Store, for example, the file specifies that a requisition list is deleted from the cache when a customer creates a new requisition list.

- b. Open the file

`WC_installdir/samples/dynacache/invalidation/catalog/cachespec.xml`

Copy the invalidation policies from this file to the `cachespec.xml` file for your store. You can place the invalidation policies at the end of your store's `cachespec.xml` file before the last element. This file describes the non-shopping events that invalidate cached data, for example, when the product description is updated in the WebSphere Commerce Accelerator the cached pages containing that product are invalidated.

- c. If necessary, change the name and directory of JSP files in the sections copied from the sample invalidation files to match your store's information. You will need to do this if the name of the store is not `ExpressStore`, or if you did not publish the store to the `ExpressStore` directory.

- d. Save and close the file,

`WAS_installdir/installedApps/host_name/WC_instanceName.ear/Stores.war/WEB-INF`

- e. Restart the WebSphere Commerce Server. Directions for restarting the WebSphere Commerce Server are described in the *WebSphere Commerce Installation Guide* for your platform and edition.

+

+ Appendix D. Deploying the Express Store

+
+
+
+

The extent and nature of the customization of the Express Store will affect your deployment strategy. The following table describes a typical deployment scenario. Except where otherwise noted the tasks shown in this table are described in this guide.

±

Machine	Task
Development machine	<ul style="list-style-type: none"> • Publish the Express Store sample store. • Load a sample catalog to learn about the catalog structure. • Create and modify; and test JSPs, Java code and EJBs according to your business requirements. • Create access control and registration scripts for any development assets. <p>These scripts will be needed to quickly register your new views and commands when you move your development assets to another machine. Creating scripts to register new commands and views is described in the section “Set access control for a page” on page 70. You may also want to refer to the <i>WebSphere Commerce Programming Guide and Tutorials</i> and the <i>WebSphere Commerce Security Guide</i>.</p> <ul style="list-style-type: none"> • Use the WebSphere Commerce Accelerator to learn styles, options and other features.
Staging server and production server	<ul style="list-style-type: none"> • Prepare the store’s catalog in CSV or XML format to load into the database. • Install and configure your WebSphere Commerce instance including, if necessary, payments configuration.
Staging server	<ul style="list-style-type: none"> • Publish the Express Store. • Remove unused languages, and currencies from the store. • Set exchange rates. • Use the WebSphere Commerce Accelerator to configure the store. For example, changing the store logo, changing the shopping flow, changing the text on store pages, and changing the default text in e-mail messages. • Load the catalog data from XML or CSV files. • Use the WebSphere Commerce Accelerator to set up taxes, shipping and discounts. • Deploy JSP, XML, Java code, and EJB assets developed on the development system. • Use the staging tools to move database from staging server to the production server. To configure a staging server refer to the <i>WebSphere Commerce Installation Guide</i> for your platform. For information about how to use the staging server refer to the <i>WebSphere Commerce Administration Guide</i>. • Copy store file assets. For a list of the file assets that you need to copy refer to the section “Location structure and directory structure” on page 61. If you have changed the <code>MemberRegistrationAttributes.xml</code> file, you will also need to copy this file to the production machine.

+

+
+
+
+
+
+
+
+
+
+
+
+
+
+
+
+
+
+
+
+
+

Stale JSP files display

You will need to set a JSP reload interval in the WebSphere Commerce Server environment. If you do not set the JSP reload interval, then you will not see changes to JSP pages when a page is reloaded. Changes to JSP pages will be visible after 30 minutes, or when the server is restarted.

1. Open the file `ibm-web-ext.xml` in an editor. The file is located in the following directory:
`WAS_installdir/config/cells/hostname/applications/WC_instancename.ear/deployments/WC_instancename/Stores.war/WEB-INF`
2. Locate the `webappext` element. This element is near the beginning of the file.
3. Set the `reloadingEnabled` attribute to `true`. If the attribute does not exist, add it.
4. Set the `reloadingInterval` attribute value to `0`. If the attribute does not exist, add it.
5. Save the file.
6. Restart the server.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM product, program, or service may be used. Any functionality equivalent to product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering the subject matter in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chrome, Minato-ku
Tokyo 1061, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independent created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Lab Director
IBM Canada Ltd. Laboratory
8200 Warden Avenue
Markham, Ontario
L6G 1C7
Canada

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

This document may contain information about other companies' products, including references to such companies' Internet sites. IBM has no responsibility for the accuracy, completeness, or use of such information.

Trademarks

The IBM logo and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both:

- DB2
- DB2 Universal Database
- IBM iSeries
- WebSphere
- 400
- OS/400

Pentium and Intel are registered trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.