

IBM WebSphere Commerce



ToolTech Sample Store: Online Help Files

Version 54

IBM WebSphere Commerce



ToolTech Sample Store: Online Help Files

Version 54

Note!

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

Contents

Chapter 1. Sample stores	1
Sample stores	1
ToolTech sample store (Business Edition)	1
Chapter 2. Setting up a sample store	3
Setting up a Sample Store	3
Chapter 3. Creating a store using a sample	5
Creating a store using a sample	5
Creating a sample store archive	5
Changing store database assets	5
Changing Web assets	14
Publishing a sample store archive from Store Services	15
Publishing a store archive using the command line (Windows)	16
Setting up Payment Manager for your store	18
Creating scheduled jobs for a sample store	19
Configuring e-mail notification for a sample store..	21
Creating an account-based campaign in ToolTech (Business Edition)	22
Adding your organization's logo to ToolTech (Business Edition)	24
Chapter 4. Sample store database assets	25
Store database assets	25
Sample store access control database assets	26
ToolTech Shipping database assets (Business Edition)	27
ToolTech Catalog database assets (Business Edition)	28
ToolTech contract, business account, and business policy database assets (Business Edition)	30
ToolTech Organization database assets (Business Edition)	31
ToolTech sample store organizations (Business Edition)	32
ToolTech sample store purchase orders (Business Edition)	32
ToolTech sample store contracts and accounts (Business Edition)	33
Chapter 5. ToolTech store pages	35
ToolTech store pages (Business Edition)	35
ToolTech Address book pages (Business Edition)	36
ToolTech search pages (Business Edition)	38
ToolTech Billing and Shipping Information page (Business Edition)	40
ToolTech Main Category page (Business Edition)	41
ToolTech Product Display page (Business Edition)	42
ToolTech Subcategory page (Business Edition)	43
ToolTech Home page (Business Edition)	43
ToolTech Confirmation page (Business Edition)	45
ToolTech Product Availability page (Business Edition)	45
ToolTech error pages (Business Edition)	46
ToolTech header (Business Edition)	47
ToolTech logon pages (Business Edition)	49
ToolTech left navigation frame (Business Edition)	51
ToolTech Order Summary page (Business Edition)	53
ToolTech Order Status page (Business Edition)	55
ToolTech order status codes (Business Edition)	57
ToolTech Your Order page (Business Edition)	58
ToolTech Quick Order page (Business Edition)	60
ToolTech registration pages (Business Edition)	60
ToolTech Add to Existing Requisition List page (Business Edition)	63

ToolTech New Requisition List page (Business Edition)	64
ToolTech View Requisition List page (Business Edition)	66
ToolTech Requisition List page (Business Edition)	66
ToolTech Edit Requisition List page (Business Edition)	68
Sample store pages: common implementation techniques	70
ToolTech Collaborative Workspaces page (Business Edition)	72
ToolTech collaborative workspaces Members page (Business Edition)	73
Chapter 6. ToolTech use cases	75
ToolTech sample store use cases (Business Edition)	75
Logon by a registered customer use case (Business Edition)	75
Register a new customer use case (Business Edition)	76
Approve a buyer use case (Business Edition)	76
Update personal information use case (Business Edition)	77
Add new address use case (Business Edition)	77
Edit an address use case (Business Edition)	78
Create a new requisition list use case (Business Edition)	79
Add an item to a requisition list use case (Business Edition)	80
Remove items from a requisition list use case (Business Edition)	80
Add items to an order use case (Business Edition)	81
Create an order using a SKU use case (Business Edition)	81
Chapter 7. ToolTech troubleshooting	83
ToolTech sample store troubleshooting (Business Edition)	83
Notices	87

Chapter 1. Sample stores

Sample stores

WebSphere Commerce provides you with several sample stores on which you can base your own store. Each store is designed to display the features found in today's top retail sites. The sample stores provided with WebSphere Commerce are the following:

- InFashion, an online business-to-consumer fashion store.
- WebFashion, an online business-to-consumer fashion store with additional enhancements.
- WebAuction, an online business-to-consumer auction store.
- NewFashion, an online business-to-consumer fashion store with new features.
-

Business

ToolTech, an online business-to-business hardware supply store.

Each sample store is packaged with WebSphere Commerce as a store archive, and as a result, no further installation is necessary. All that is required to view each sample store is to create a new store archive based on the sample store using the Store Services tools, then publish the archive to the WebSphere Commerce Server. For more information, see *Creating a store archive using Store Services*.

Since store creation in WebSphere Commerce is based upon selecting a sample store archive and modifying it, all of the sample stores are designed to act as the basis for your store. The sample stores are based on a typical shopping flow. All sample store pages can be customized.

For more information on each sample store, see the sample store's files listed in the following links.

ToolTech sample store (Business Edition)

Business

ToolTech is the business-to-business (B2B) online hardware store provided with WebSphere Commerce. ToolTech highlights the tasks a buyer can perform at a seller's e-commerce site. Some of the features included in the ToolTech sample store are:

- Contract-based purchasing
- Requisition lists
- Contract-based pricing
- RFQ creation
- Split orders and backorders.
- Multiple shipping addresses
- Weight-based shipping
- Product search capability
- Buyer registration
- Collaboration workspaces
- Customer care
- Adding your organization's logo to the store
- Account-based campaigns

The ToolTech store provides all the pages and features necessary for a functioning B2B online store. ToolTech is packaged with WebSphere Commerce as a store archive, and as a result, no further

installation is necessary. All that is required to view the sample store is to create a new store archive based on ToolTech using the Store Services tools, then publish the archive to the WebSphere Commerce Server. For more information, see the related links below.

Since store creation in WebSphere Commerce is based upon selecting a sample store archive and modifying it, ToolTech is designed to act as the basis for your store. It is based on a typical shopping flow. The sample store pages can be customized. All ToolTech sample store pages use JavaServer pages and XML to load the necessary data. ToolTech pages are cached based on user type, state, and role, in addition to the cache command functions available for all pages in WebSphere Commerce. For more information on the WebSphere Commerce cache command functions and parameters, see the related links below.

Chapter 2. Setting up a sample store

Setting up a Sample Store

After publishing a sample store, you may be required to perform additional steps to get the store running at full functionality. This page lists the steps required to get each sample store set up.

Setting up NewFashion

After publishing, the following steps are required to set up the NewFashion store:

- Creating scheduled jobs for a sample store
- Configuring e-mail notification for a sample store

Note: If you want to use the Customer Care collaboration feature with the NewFashion store, you must first install Lotus Sametime. For more information on the installation of Lotus Sametime refer to the *IBM WebSphere Commerce Additional Software Guide*. For more information on configuring NewFashion to use Customer Care, see the related tasks below.

Setting up WebFashion

After publishing, the following steps are required to set up the WebFashion store:

- Creating scheduled jobs for a sample store
- Configuring e-mail notification for a sample store
- Creating discounts in WebFashion
- Creating a marketing campaign in WebFashion
- Creating a new store owner for WebFashion

Setting up InFashion

After publishing, the following steps are required to set up the InFashion store:

- Creating scheduled jobs for a sample store
- Configuring e-mail notification for a sample store

Setting up WebAuction

After publishing, the following steps are required to set up the WebAuction store:

- Setting up Payment Manager for your store
- Enabling auctions
- Creating scheduled jobs for a sample store
- Configuring e-mail notification for a sample store
- Creating discounts
- Creating a marketing campaign
- Creating a new store owner

Business

Setting up ToolTech

Before publishing the ToolTech store you must have Payment Manager installed and running. After publishing, the following steps are required to set up the ToolTech store:

- Setting up Payment Manager for your store (only if you forgot to have Payment Manager running at the time of publishing).
- Selecting approval types for an organization (Buyer and order approvals)
- Enabling and disabling the RFQ trading mechanism
Enabling RFQ notification messaging
- Using LDAP with WebSphere Commerce (If you are configured for LDAP)
- Creating an organization (if you wish to create a new one when publishing the second time)

Note: If Payment Manager was not installed or not running at the time of publishing, the brand types configured in Payment Manager should match the brand types specified for contract 3456 for Credit Card.

Note: If you want to use collaboration features with the ToolTech store, you must first install Lotus Sametime for the Customer Care feature, or Lotus QuickPlace for the Collaborative Workspaces feature. For more information on the installation of Lotus Sametime and Lotus QuickPlace refer to the *IBM WebSphere Commerce Additional Software Guide*. For more information on configuring ToolTech to use Customer Care or Collaborative Workspaces features, see the related tasks below.

Chapter 3. Creating a store using a sample

Creating a store using a sample

To create a store using one of the sample stores, do the following:

1. (Optional)

Business

Create an organization to act as the seller.

To determine if you want to create a new organization to act as the seller, see [Shared data assets](#).

2. Create users for the following roles:

- Site Administrator (if you are not using the default Site Administrator)
- Seller Administrator
- Store Administrator
- Store Developer

Important: In order to publish a store archive you must have Site Administrator, Store Administrator, or Store Developer of all stores access.

If you are creating a user with Store Administrator authority, ensure that the access group applies to all stores.

3. Create a store archive using Store Services
4. Change store database assets.
5. Change Web assets.
6. Publish a store archive.
7. (Optional) Configuring stores.
8. Set up Payment Manager for your store.

Important:

1. In some instances, the sample stores require some set up before all features will function properly. If you are basing your store on a sample store, you may need to complete the set up. For more information, see [Setting up a sample store](#).
2. Certain store data assets, like catalogs and fulfillment centers, are shared between stores. As a result, if you publish multiple stores based on the same sample store, the catalog and fulfillment center will be the same for each store. If you have made changes in one catalog, publishing another store on the same sample will overwrite these changes. For more information and how to avoid overwriting any changes, see [Shared data assets](#).

Creating a sample store archive

To create a store archive that can be used as a sample with Store Services, see the *IBM WebSphere Store Developer's Guide*.

Changing store database assets

If you create your store archive using the tools in Store Services, your new store archive will initially contain the same store database assets as the sample store archive on which you based it, for example, `infashion.sar`. In a store archive, the store database assets take the form of XML files.

In most cases, to change the store database assets, you must edit the XML files directly. In some cases, you can use the tools in Store Services to edit the database assets.

You also have the option of editing database assets directly, that is, once you have published your store archive to the commerce server, you can edit the database using the WebSphere Commerce Accelerator, the Loader package, or directly through SQL inserts. If you choose to edit the database rather than the assets in the store archive, you must either update your store archive to match your changes in the database, or discontinue using the store archive.

Your options are listed in the following table.

Important:

1. The tools in Store Services search for the asset names listed in the table below. As a result, if you want to edit your store archive using the tools in Store Services, you must use the same assets names in your store archive.
2. When a store archive is published to the WebSphere Commerce Server, the database information is loaded in the order specified in the Assets column below. As a result, The order of your assets, as specified in the sarinfo.xml file, should match the order of the assets specified below.
3. The order of the database information in each XML file does not necessarily have to match the order specified in the Database Tables column below. However information for a parent table must precede information for a child table.
4. The information marked optional is not required to create a functioning store.
5. For the Database Edit Options column, note that unless specified otherwise all database assets can be edited using SQL inserts or the Loader package. As a result, the column reflects which assets can be edited by the WebSphere Commerce Accelerator.

Store Database Assets	Database Tables	Store Archive Edit Options	Database Edit Options
fulfillment	FulfillmentCENTER (0...1)	<ul style="list-style-type: none"> • Edit store archive XML files. 	<ul style="list-style-type: none"> • New Fulfillment Center and Changing Fulfillment Center pages in WebSphere Commerce Accelerator. • See Creating a fulfillment center and Changing a fulfillment center

Store	Database Tables Database Assets	Store Archive Edit Options	Database Edit Options
store	STOREENT	<ul style="list-style-type: none"> • Edit store archive XML files. • Changing store information using the Store Profile notebook. Store Profile notebook edits the following database columns: <ul style="list-style-type: none"> – IDENTIFIER – DIRECTORY – SETCURR 	
	STADDRESS	<ul style="list-style-type: none"> • Changing store information using the Store Profile notebook. 	<ul style="list-style-type: none"> • New Fulfillment Center and Changing Fulfillment Center pages in WebSphere Commerce Accelerator. • See Creating a fulfillment center and Changing a fulfillment center
	STOREENTDS	<ul style="list-style-type: none"> • Changing store information using the Store Profile notebook. Store Profile notebook edits the following database columns: <ul style="list-style-type: none"> – DESCRIPTION – DISPLAYNAME 	
	STORELANG	<ul style="list-style-type: none"> • Changing store information using the Store Profile notebook. Store Profile notebook edits the following database columns: <ul style="list-style-type: none"> – LANGUAGE_ID 	
	STORELANGDS	<ul style="list-style-type: none"> • Edit store archive XML files. 	
	STORE	<ul style="list-style-type: none"> • Changing store information using the Store Profile notebook. Store Profile notebook edits the following database columns: <ul style="list-style-type: none"> – PHONE1 – CITY – STORE_ID – COUNTRY – STATE – EMAIL1 – ADDRESS1 – ADDRESS2 – FAX – ZIPCODE 	
	DISPENTREL (default CATENTRY template, CATENTRY_ID = -1)	<ul style="list-style-type: none"> • Changing store information using the Store Profile notebook. Store Profile notebook edits the PAGENAME database column. 	
	DISPCGPREL (default CATEGORY template, CATGROUP_ID = -1)	<ul style="list-style-type: none"> • Changing store information using the Store Profile notebook. Store Profile notebook edits the PAGENAME database column. 	
	VENDOR	<ul style="list-style-type: none"> • Edit store archive XML files. 	

	Store Database Tables Database Assets	Store Archive Edit Options	Database Edit Options
catalog	CATGROUP	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	CATGRPATTR	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	
	CATGRPDESC	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	CATALOG	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	
	CATALOGDSC	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	
	CATTOGRP	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	
	CATGRPREL	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	CATENTRY	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	CATENTDESC	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	ATTRIBUTE	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	ATTRVALUE	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	CATGPENREL	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	CATENTREL	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	
	BASEITEM	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	ITEMSPC	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	VERSIONSPC	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	
	DISTARRANG	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	
8	RECEIPT ToolTech Sample Store: Online Samples	<ul style="list-style-type: none"> Changing catalog information Edit store archive XML files. 	
	RCPTAVAIL	<ul style="list-style-type: none"> Changing catalog information 	

	Store Database Tables Database Assets	Store Archive Edit Options	Database Edit Options
tax	JURSTGROUP (common between tax and shipping)	<ul style="list-style-type: none"> • Edit store archive XML files. • Changing tax settings using the Tax notebook. Tax notebook edits the following database columns: <ul style="list-style-type: none"> – JURSTGROUP_ID – DESCRIPTION – SUBCLASS – STOREENT_ID – CODE 	
	JURST (common between tax and shipping)	<ul style="list-style-type: none"> • Changing tax settings using the Tax notebook. Tax notebook edits the following database columns: <ul style="list-style-type: none"> – JURST_ID – COUNTRY – STOREENT_ID – CODE – SUBCLASS – STATE 	
	JURSTGPREL (common between tax and shipping)	<ul style="list-style-type: none"> • Changing tax settings using the Tax notebook. Tax notebook edits the following database columns: <ul style="list-style-type: none"> – JURST_ID – JURSTGROUP_ID – SUBCLASS 	
	CALMETHOD (common between tax and shipping)	<ul style="list-style-type: none"> • Edit store archive XML files. 	
	TXCDCLASS (optional, only for categorizing CALCODE, used by the WebSphere Commerce Accelerator)	<ul style="list-style-type: none"> • Edit store archive XML files. 	
	TAXCGRY	<ul style="list-style-type: none"> • Changing tax settings using the Tax notebook. Tax notebook edits the following database columns: <ul style="list-style-type: none"> – TAXCGRY_ID – STOREENT_ID – NAME 	
	CALCODE (common between tax and shipping)	<ul style="list-style-type: none"> • Changing tax settings using the Tax notebook. Tax notebook edits the following database columns: <ul style="list-style-type: none"> – CALCODE_ID – CODE – CALUSAGE_ID – STOREENT_ID – GROUPBY – CALMETHOD_id – CALMETHOD_id_app – CALMETHOD_id_qfy 	
	CALCODEDSC (common between tax and shipping, optional)	<ul style="list-style-type: none"> • Edit store archive XML files. 	

Store	Database Tables Database Assets	Store Archive Edit Options	Database Edit Options
taxfulfillment	TAXRULE	<ul style="list-style-type: none"> • Edit store archive XML files. • Changing tax settings using the Tax notebook. Tax notebook edits the following database columns: <ul style="list-style-type: none"> – CALRULE_ID – FFMCENTER_ID – JURSTGROUP_ID 	
storecatalog	CATALCD (optional)	<ul style="list-style-type: none"> • Edit store archive XML files. 	<ul style="list-style-type: none"> • Product Management tools in WebSphere Commerce Accelerator
	CATGPCALCD (optional)	<ul style="list-style-type: none"> • Edit store archive XML files. 	

Store Database Assets	Database Tables	Store Archive Edit Options	Database Edit Options
shipping	JURSTGROUP (common between tax and shipping)	<ul style="list-style-type: none"> Edit store archive XML files. Changing shipping settings using the Shipping notebook. Shipping notebook edits the following database columns: <ul style="list-style-type: none"> JURSTGROUP_ID DESCRIPTION SUBCLASS STOREENT_ID CODE 	
	JURST (common between tax and shipping)	<ul style="list-style-type: none"> Changing shipping settings using the Shipping notebook. Shipping notebook edits the following database columns: <ul style="list-style-type: none"> JURST_ID COUNTRY STOREENT_ID CODE SUBCLASS STATE 	
	JURSTGPREL (common between tax and shipping)	<ul style="list-style-type: none"> Changing shipping settings using the Shipping notebook. Shipping notebook edits the following database columns: <ul style="list-style-type: none"> JURST_ID JURSTGROUP_ID SUBCLASS 	
	SHIPMODE	<ul style="list-style-type: none"> Changing shipping settings using the Shipping notebook. Shipping notebook edits the following database columns: <ul style="list-style-type: none"> CODE CARRIER SHIPMODE_ID 	
	SHPMODEDSC	<ul style="list-style-type: none"> Changing shipping settings using the Shipping notebook. Shipping notebook edits the following database columns: <ul style="list-style-type: none"> SHIPMODE_ID LANGUAGE_ID 	
	CALMETHOD (common between tax and shipping)	<ul style="list-style-type: none"> Edit store archive XML files. 	
	CALCODE (common between tax and shipping)	<ul style="list-style-type: none"> Changing shipping settings using the Shipping notebook. Shipping notebook edits the following database columns: <ul style="list-style-type: none"> CALCODE_ID CODE CALUSAGE_ID STOREENT_ID GROUPBY CALMETHOD_ID 	
	CALCODEDSC (optional)	<ul style="list-style-type: none"> Changing shipping settings using the Shipping notebook. 	

Store	Database Tables Database Assets	Store Archive Edit Options	Database Edit Options
ship fulfillment	SHPJCRULE (at least one default rule for store)	<ul style="list-style-type: none"> Changing shipping settings using the Shipping notebook. Shipping notebook edits the following database columns: <ul style="list-style-type: none"> – CALRULE_ID – SHPARRANGE_ID – JURSTGROUP_ID 	
	SHPARRANGE	<ul style="list-style-type: none"> Changing shipping settings using the Shipping notebook. Shipping notebook edits the following database columns: <ul style="list-style-type: none"> – SHARRAND_ID – STORE_ID – FFMCENTER_ID – SHIPMODE_ID 	
store catalog	STORECAT	<ul style="list-style-type: none"> Edit store archive XML files. 	
	STORECENT	<ul style="list-style-type: none"> Edit store archive XML files. 	
	STORECGRP	<ul style="list-style-type: none"> Edit store archive XML files. 	
	DISPENTREL	<ul style="list-style-type: none"> Edit store archive XML files. 	
	DISPCGPREL	<ul style="list-style-type: none"> Edit store archive XML files. 	
store fulfillment	INVENTORY	<ul style="list-style-type: none"> Edit store archive XML files. 	
offerings	TRADEPOSCN (1)	<ul style="list-style-type: none"> Edit store archive XML files. 	
	MGTRDPSCN (optional, for customer group)	<ul style="list-style-type: none"> Edit store archive XML files. 	
	OFFER	<ul style="list-style-type: none"> Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	OFFERDESC	<ul style="list-style-type: none"> Edit store archive XML files. 	
	OFFERPRICE	<ul style="list-style-type: none"> Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	LISTPRICE	<ul style="list-style-type: none"> Edit store archive XML files. 	
command	URLREG	<ul style="list-style-type: none"> Edit store archive XML files. 	
	CMDREG	<ul style="list-style-type: none"> Edit store archive XML files. 	
	VIEWREG	<ul style="list-style-type: none"> Edit store archive XML files. 	

Store Database Assets	Database Tables	Store Archive Edit Options	Database Edit Options
currency	CURCONVERT	<ul style="list-style-type: none"> Edit store archive XML files. 	
	CURFORMAT	<ul style="list-style-type: none"> Edit store archive XML files. 	
	CURFMTDESC	<ul style="list-style-type: none"> Edit store archive XML files. 	
	CURCVLIST	<ul style="list-style-type: none"> Edit store archive XML files. 	
	CURLIST	<ul style="list-style-type: none"> Changing store information using the Store Profile notebook. Store Profile notebook edits CURRSTR. 	
campaign	EMSPOT	<ul style="list-style-type: none"> Edit store archive XML files. 	<ul style="list-style-type: none"> e-Marketing Spot Management tools in WebSphere Commerce Accelerator
	CAMPAIGN	<ul style="list-style-type: none"> Edit store archive XML files. 	<ul style="list-style-type: none"> Campaign Management tools in WebSphere Commerce Accelerator
store catalog	CATENCLAPPID	<ul style="list-style-type: none"> Edit store archive XML files. 	<ul style="list-style-type: none"> Product Management tools in WebSphere Commerce Accelerator
	CATGPCALCD	<ul style="list-style-type: none"> Edit store archive XML files. 	
store	STOREDEF	<ul style="list-style-type: none"> Edit store archive XML files. 	
consistency check		<ul style="list-style-type: none"> Edit store archive XML files. 	
payment	CMMDREG	<ul style="list-style-type: none"> Changing payment settings. Edit store archive XML files. 	
	VIEWREG	<ul style="list-style-type: none"> Changing payment settings. Edit store archive XML files. 	
business policy	POLICY	<ul style="list-style-type: none"> Edit store archive XML files. 	
	POLICYCMD	<ul style="list-style-type: none"> Edit store archive XML files. 	
organization	ORGENTITY	<ul style="list-style-type: none"> Edit store archive XML files. 	
	MPRREL	<ul style="list-style-type: none"> Edit store archive XML files. 	
	ADDRBOOK	<ul style="list-style-type: none"> Edit store archive XML files. 	
	ADDRESS	<ul style="list-style-type: none"> Edit store archive XML files. 	

Store Database Assets	Database Tables	Store Archive Edit Options	Database Edit Options
business account	TERMCOND	The account assets cannot be updated and republished through the store archive. If you need to change the account assets you must edit the assets in the database using the Business Relationship Management tools in the WebSphere Commerce Accelerator.	Business Relationship Management tools in WebSphere Commerce Accelerator Important: Accounts cannot be loaded using the Loader Package.
	ACCOUNT		
	TRADING		
	TCDESC		
	PATTRVALUE		
	CREDITLINE		
	TRDDESC		
	POLICYTC		
	BUYERPO		
	PARTICIPNT		
	ATTACHMENT		
	TRDATTACH		
contract	CONTRACT	<ul style="list-style-type: none"> Editing contract XML files in the store archive 	Business Relationship Management tools in WebSphere Commerce Accelerator Important: Contracts cannot be loaded using the Loader Package.
	TERMCOND		
	PRODUCTSET		
	TRADING		
	TCDESC		
	PATTRVALUE		
	TRDDESC		
	POLICYTC		
	PARTICIPNT		
	TRADEPOSCN		
	ATTACHMENT		
	OFFER		
	TRDATTACH		
	OFFERPRICE		
	STORECNTR		
	PURCHASELT		
	STOREDEF (used by Store Services)		
APRVSTATUS			
FLINSTANCE			

Changing Web assets

To change the Web assets in your store archive, you have the following options:

- Import the store archive into WebSphere Studio and make the necessary changes to the Web assets using Page Designer or a tool of your choice. If necessary, create new store pages. Then export the Web assets back to the store archive or the running store.

- Download the Web assets from the store archive to a location of your choice using the Web Asset dialog in Store Services, then change them using a tool of your choice. Or replace the Web assets in the store archive with your existing Web assets using the Web Asset dialog in Store Services. If necessary, create new store pages.
- Manually open the Web assets compressed archived file in the store archive and make changes to existing files or add new files.

Publishing a sample store archive from Store Services

Publishing a sample store archive to a WebSphere Commerce Server allows you to create a running store. To publish a store archive, complete the following steps:

1. Ensure that you have Site Administrator or Store Administrator access. If you have Store Administrator access, ensure that the access is for all stores.
2. Ensure that you have completed the tasks in Ensure appropriate services and servers are running.
3. Open Store Services.
4. From the **Store Archive** list, select the store archive you wish to publish.
5. Click **Publish**.
The Publish Store Archive page displays.
6. Select your desired publishing options. For more information on publishing options, click **Help**.
Tip: To create a fully functional store, select all publishing options, including the product data option, the first time you publish a store archive.
7. Select **OK**.
While the store publishes you are returned to the Store Archive list page. The publishing state is reflected in the Publish status column. Click **Refresh** to update the status.
8. Select the store archive from the list and click **Publish Summary** to see the results of the publish.
9. When publishing is complete, click **Launch Store** to view and test your store. When you have finished, bookmark the site, and close the browser.

Important:

1. If you change the Web application Web path or the Web application document root, you must ensure that they match the paths defined in the WebSphere Commerce Server.
2. Only one store archive at a time can be published. Concurrent publishing is not supported and causes the publish of both stores to fail.
3. During publish, the consistency checker confirms that the files referenced by the store archive exist. If the consistency check finds an error, the error will be written to the log. Publishing continues as normal.
4. Before republishing a store, delete the files from following directory:

▶ NT

drive:\WebSphere\CommerceServer\instances*instancename*\cache

▶ 2000

drive:\Program Files\WebSphere\CommerceServer*instancename*\cache

▶ AIX

/usr/WebSphere/CommerceServer/instances/instancename/cache

▶ Solaris

/opt/WebSphere/CommerceServer/instances/instancename/cache

▶ 400

/QIBM/UserData/WebCommerce/instances/*instancename*/cache

While in the store development phase, you may want to disable caching. For more information, see [Configure caching](#).

5. When you launch the store from Store Services, you are logged into the store with the same user name and password that you used to log into Store Services. If you change your password in the store, you are also changing it for Store Services. Instead, to test the features in the store, including changing your password, save the site address, close all browser windows, then log on to the store again.
6. If you are logged in as the default administrator, you cannot browse a store based on the business to business sample store. Instead create a new user that belongs to the default organization, then browse the store.
7. Certain store data assets, like catalogs and fulfillment centers, are shared between stores. As a result, if you publish multiple stores based on the same sample store, the catalog and fulfillment center will be the same for each store. If you have made changes in one catalog, publishing another store on the same sample will overwrite these changes. For more information and how to avoid overwriting any changes, see [Shared data assets](#).

Publishing a store archive using the command line (Windows)

Although the primary method of publishing a store archive is through Store Archive Services, you can also publish a store archive using the command line. To publish using the command line, do the following:

1. Ensure that you have Site Administrator or Store Administrator access. If you have Store Administrator access, ensure that the access is for all stores.
2. Type the following command, using valid parameters for your store archive:

```
publishstore sarName hostname logonId logonPwd {insert|update}  
destination1=webapp.zip,destination2=properties.zip
```

where:

- *sarName* is the store archive name. *sarName* is case sensitive. Ensure you use the correct case.
- *hostname* where *hostname* is the fully qualified TCP/IP name of your WebSphere Commerce Server the tools port number for your instance. You can find the tools port number in the Configuration Manager, under **Instance properties > WebSphere**. By default this is `hostname:8000`.
- *logonId* is the WebSphere Commerce user ID.
- *logonPwd* is the user logon password for WebSphere Commerce
- *insert|update* determines whether the store is being created (insert) or updated (update).
- *{ALL|NOCATLG}* determines which XML files in the SAR should be published. To publish all, use ALL. To publish everything except for catalog, use NOCATLG.
- *destination1=webapp.zip,destination2=properties.zip*, is the list of the file asset files in the SAR, for example `webapp.zip`, and the paths to which they will be published, for example, *destination1* is

▶ NT

```
drive:\Websphere\AppServer\installedApps\  
WC_Enterprise_App_instancename.ear\wcstores.war
```

▶ 2000

```
drive:\Program Files\Websphere\AppServer\installedApps\  
WC_Enterprise_App_instancename.ear\wcstores.war
```

destination2 is

▶ NT

```
drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\
wcstores.war\WEB-INF\classes
```

▶ 2000

```
drive:\ProgramFiles\WebSphere\AppServer\installedApps\
WC_Enterprise_App_demo.ear\wcstores.war\WEB-INF\classes
```

The following is an example of the command:

```
publishstore mysar.sar myhost wcsadmin wcsadmin insert ALL
"d:\websphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\
wcstores.war=webapp.zip,d:\websphere\AppServer\installedApps\
WC_Enterprise_App_demo.ear\wcstores.war\WEB-INF\classes=properties.zip"
```

3. Open Internet Explorer. Go to the following Web address:
http://hostname/webapp/wcs/stores/store_directory/index.jsp, where the *store directory* is the directory of the store you just published.
Your store displays.

Note: If you are publishing a store archive created with WebSphere Commerce Suite, version 5.1, you will need to complete the following steps before launching the store through the URL:

1. If you are using DB2, do the following. If you are using Oracle, go to step 2.
 - a. When the command has finished running, from the **Start** menu, select **Programs, DB2 for Windows NT**, then **Command Window**.
 - b. In the **DB2 CLP** window, type `db2 connect to dbname`, where `dbname` is the database to which you are publishing your store. Press **Enter**.
 - c. In the command line, type `db2 select * from store`. Press **Enter**. A list of stores displays. Take note of the number of the store you created.
 - d. In the command line, type `db2 select * from catalog`. Press **Enter**. A list of catalogs displays. Take note of the number of the sample store catalog.
 - e. Go to step 3.
2. If you are using Oracle, do the following:
 - a. When the command has finished running, from the **Start** menu, select **Programs, Oracle - HomeOra81, Application Development**, then **SQL Plus**.
 - b. In the window, type your user name and password and host string.
 - c. In the SQL Plus window, type `select * from store;`. Press **Enter**. A list of stores displays. Take note of the number of the store you created.
 - d. In the SQL Plus window, type `select * from catalog;`. Press **Enter**. A list of catalogs displays. Take note of the number of the sample store catalog.
 - e. Go to step 3.
3. Open Internet Explorer. Go to the following URL:
<http://hostname/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=storeId from step1c or 2c&langId=-1&catalogId=catalogId from step1d or 2d>
Your store displays.
If you have problems displaying your store, see Troubleshooting publishing.

Important:

1. During publish, the consistency checker confirms that the files referenced by the store archive exist. If the consistency check finds an error, the error will be written to the log. Publishing continues as normal.
2. Before republishing a store, delete the files from following directory:

▶ NT

drive:\WebSphere\CommerceServer\instances*instancename*\cache

▶ 2000

drive:\Program Files\WebSphere\CommerceServer\instances*instance name*\cache

3. While in the store development phase, disable caching triggers and the cache. Leaving the cache on may result in the following:
 - Changes made to JSP files may not display in the browser.
 - Caching triggers will be invoked during publish, when the database is updated. Caching triggers may generate unnecessary database activity that could result in a database transaction log overflow. For more information, see *Configure caching*.
4. If you are logged in as the default administrator, you cannot browse a store based on the business to business sample store. Instead create a new user that belongs to the default organization, then browse the store.

Setting up Payment Manager for your store

You can complete the Payment Manager setup for your store using the Administration Console or the Payment Manager user interface. If you use the Administration Console, menu items appear on the **Payment Manager** menu. If you use the Payment Manager user interface, menu items appear under **Administration** in the navigation frame.

If you create your store using the sample store archive (which is recommended), Payment Manager will be partially configured.

To complete the setup of Payment Manager for your store, do the following:

1. Open the Administration Console or Payment Manager user interface.
2. Assign Payment Manager user roles to WebSphere Commerce users as required. To assign Payment Manager user roles, select **Users**.

The default WebSphere Commerce Site Administrator, `wcsadmin`, is assigned the Payment Manager administrator role by default. You may wish to assign other WebSphere Commerce users various Payment Manager roles.

3. Authorize cassettes for your store by doing the following:
 - a. Select **Merchant Settings**.
 - b. Click your store name in the **Merchant name** column.
 - c. Select the cassettes you wish to authorize for your store.
 - d. Click **Update**.

Important: If you created your store manually, you must add a new merchant (your store) in order to authorize cassettes for your store. When creating a new merchant, the merchant number specified must match the WebSphere Commerce Store ID. You can create a new merchant by selecting **Merchant Settings** then clicking **Add a Merchant**.

4. Configure cassettes for your store by doing the following:
 - a. Select **Merchant Settings**.
 - b. Select a cassette to configure by clicking the icon appearing in the row for your store and the column for the cassette you want to configure.
 - c. Click **Accounts** on the cassette's page for your store and do one of the following:
 - To change existing accounts, click the account name.
 - To create a new account, click **Add an Account**.

For more information on configuring the Cassette for BankServACH, refer to the *IBM WebSphere Payment Manager for Multiplatforms, Cassette for BankServACH Supplement, Version 3.1*.

For information on configuring the OfflineCard and CustomOffline cassettes, refer to the *IBM WebSphere Payment Manager for Multiplatforms, Administrator's Guide, Version 3.1*.

For information on configuring the Cassette for SET, refer to the *IBM WebSphere Payment Manager for Multiplatforms, Cassette for SET Supplement, Version 3.1*.

For information on configuring the Cassette for CyberCash, refer to the *IBM WebSphere Payment Manager for Multiplatforms, Cassette for CyberCash Supplement, Version 3.1*.

For information on configuring the Cassette for VisaNet, refer to the *IBM WebSphere Payment Manager for Multiplatforms, Cassette for VisaNet Supplement, Version 3.1*.

For help when using Payment Manager from the Administration Console or the Payment Manager user interface, click



in the upper right corner of the Payment Manager page you are working on.

For more information on the above or any other Payment Manager administration tasks, refer to the *IBM WebSphere Payment Manager for Multiplatforms, Administrator's Guide, Version 3.1*.


For details on installing WebSphere Payment Manager, refer to the following:

- *IBM WebSphere Commerce Business Edition, Installation Guide, Version 5.4*
- *IBM WebSphere Payment Manager for Multiplatforms, Install Guide, Version 3.1*.

Creating scheduled jobs for a sample store

After publishing a sample store, scheduled jobs needs to be created for the store. The following table shows the scheduled jobs that must be created for each sample store:

Sample store	Scheduled jobs required
InFashion	<ul style="list-style-type: none">• BalancePayment• PayCleanup• ReturnCreditAndCloseScan
NewFashion	<ul style="list-style-type: none">• BalancePayment• PayCleanup• ProcessBackorders• RAReallocate• ReleaseExpiredAllocations• ReleaseToFulfillment• ReturnCreditAndCloseScan
WebFashion	<ul style="list-style-type: none">• BalancePayment• PayCleanup• ReturnCreditAndCloseScan

Sample store	Scheduled jobs required
 ToolTech	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ProcessBackorders • RAReallocate • ReleaseExpiredAllocations • ReleaseToFulfillment • ReturnCreditAndCloseScan
WebAuction	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ReturnCreditAndCloseScan

Here is a brief descriptions of the jobs:

BalancePayment

This job calls the DoDepositCmd task command to capture payment once the order has been shipped. This command implements the automatic payment capture function for WebSphere Commerce.

PayCleanup

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

ProcessBackorders

This job allocates inventory to backorders which were created when inventory was not available.

RAReallocate

(Redistribute allocations against expected inventory) This job redistributes open Expected Inventory Records (EIR) against existing backorders. This is required to more accurately predict when backordered order items will be available as EIR information is added or modified, and as previously backordered items are deleted or allocated.

ReleaseExpiredAllocations

This job returns allocated inventory back into the receipt table from a previously allocated order line item which has exceeded its expiration time limit.

ReleaseToFulfillment

This job releases allocated items on an order to fulfillment.

ReturnCreditAndCloseScan

This job scans for return merchandise authorizations that are eligible to be credited and marked closed.

To create a scheduled job, follow the instructions found in [Scheduling a store level job](#) for each job. The following table lists the recommended parameters for each job:

Scheduled job name	Recommended start time	Recommended interval (seconds)	Recommended priority
BalancePayment	00:00	86400	1
PayCleanup	00:00	86400	1
ProcessBackorders	00:00	43200	8
RAReallocate	00:00	86400	1
ReleaseExpiredAllocations	00:00	3600	8
ReleaseToFulfillment	00:00	600	10

Scheduled job name	Recommended start time	Recommended interval (seconds)	Recommended priority
ReturnCreditAndCloseScan	00:00	86400	1

Setting the start time to 00:00 starts the scheduled job immediately.

Note: The **Job Parameters** field in the Schedule Job window does not need to be filled in for these jobs.

Configuring e-mail notification for a sample store.

The following procedure enables customer e-mail notification when payment is authorized, when an order is authorized, and when an order is cancelled. Note that you must have a mail server set up in order to e-mail customers.

Note: If you do not have a mail server set up, you will not be able to send e-mail notifications from your store, but the rest of the features of the sample store will still work.

The different sample stores support different e-mail notifications. The following table shows the e-mail notifications supported for each store:

Sample Store	E-mail notifications supported	Message type
InFashion	Password reset	Notification message for password reset
NewFashion	Authorized order	Notification message for a authorized order
	Password reset	Notification message for password reset
	Submission order	Notification message for a received order
	Canceled order	Notification message for a canceled order
	Shipping notification	Message for notifying the customer of an order release manifestation
WebFashion	Authorized order	Notification message for a authorized order
	Password reset	Notification message for password reset

Note: The WebAuction sample store is based on WebFashion. To set up e-mail notification you need to do all of the steps for WebFashion plus additional auction-related steps. For more information on the e-mail notification steps for WebAuction, see the Related Tasks below.

To enable e-mail notifications, do the following:

1. Ensure that the IBM WebSphere Application Server administration server is started.
2. Open the Administration Console using a Site Administrator ID.
3. On the Administration Console Site/Store Selection page select **Store**. The Select Store and Language section displays.
4. From the **Name** drop-down, select the store.
5. From the **Language** drop-down list, select the language. Click **OK**. The Store Administration Console home page displays.

6. From the **Configuration** menu, click **Transports**. The Transport Configuration page displays.
 - a. Verify that the **E-mail** transport has a status of 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, for example, `myserver.ibm.com`.
 - d. In the **Protocol** field, type `smtp` in lowercase letters, or the protocol of your choice. Click **OK**.
7. From the **Configuration** menu, click **Message Types**. The Message Type Configuration page displays.
8. 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**.
 - 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, for example, <code>example.ibm.com</code>
Protocol	Type <code>smtp</code> (you must use lowercase letters), or 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 email address that you want to use as the sender of the message, for example, <code>orders@example.ibm.com</code> . The address must be an email 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, <code>Your order has been accepted</code> .

- h. Click **Finish**. The Message Type Configuration page displays.
9. For each message type in your sample store, repeat step 8.

Notes:

- It will often take a long time before canceled order notification e-mail is sent. The time can be shortened by:
 1. Setting the value of the `REJECTEDORDEREXPIRY` column in the `STORE` database table to a smaller value.
 2. Change the `PayCleanup` scheduled job to have a smaller schedule interval.
- For more information about shipping notifications, see `ReleaseShipNotify` message

Creating an account-based campaign in ToolTech (Business Edition)

Business

In the ToolTech store you can create campaigns based on the accounts to which you belong. To create an account based campaign you must do the following:

- Create customer profiles.
- Create the campaign.
- Create campaign initiatives.
- Create e-marketing spots.

Create customer profiles

To create customer profiles do the following:

1. Logon to the WebSphere Commerce Accelerator.
2. Click **Marketing > Customer Profiles**. The Customer Profiles page displays.
3. Click **New**. The General page displays.
4. Type in a name for your customer profile in the text box, such as Woodworking Companies.
5. On the left navigation frame, click on **Registration > Registration Status**. The Registration Status page displays.
6. Select **Registered**.
7. On the left navigation frame, click on **Account > Accounts**. The Accounts page displays.
8. Select **Customer must belong to one of the selected accounts**.
9. Add the available accounts to the “Selected accounts” box by clicking **Add**. **Note:** Account 1 is for Organization A, and Account 2 is for Organization B.
10. Click **OK**, at the bottom of the page. You will be brought back to the Customer Profile page where you can see the profile you just created.

Create the campaign

To create the campaign do the following:

1. Click **Marketing > Campaigns**. The Campaigns page displays.
2. Click **New**. The General Definition page displays.
3. Type in a campaign name and description in the text boxes provided. For example, Woodworking Tools, Discount on woodworking tools.
4. Click **Next**, at the bottom of the page. The Business Objectives Definition page displays.
5. Type in your campaign sponsor and objectives in the text boxes if desired.
6. Click **Finish**, at the bottom of the page. Your campaign has been successfully created.

Create campaign initiatives

To create campaign initiatives do the following:

1. Select the campaign you just created and click **Initiatives** on the right side-menu.
2. Click **New**. The General Definition page displays.
3. Type in a name for your initiative in the textbox.
4. Select your campaign in the Campaign drop-down box.
5. On the left navigation frame, select **What**, then click **Next**. The Defining displayed dynamic content page displays.
6. Select **Suggest specific product(s)**.
7. Click **Find** or **Browse** to choose products.
8. Select **Who**, then click **Next**. The Defining target customers page displays.
9. Select **Target specific customer profiles**.
10. Add the available customer profiles to the “Selected Customer profiles” box by clicking **Add**.
11. Click **Next** at the bottom of the page. The Defining dates for targeted customers page displays.
12. Select a defining date option if desired.
13. Click **Next**. The Defining target customer behaviors page displays. To create a customer behavior click **Add** and follow the online help instructions.
14. Click **Finish**. Your campaign initiative has been successfully created.

Create e-marketing spots

To create e-marketing spots do the following:

1. Click **Marketing > E-Marketing Spots**. The e-Marketing spots page displays.
2. In the list of items select "StoreHomePage" and click **Schedule** on the right side-bar menu. The Campaign initiative scheduling page displays.
3. Click **New**. The Schedule Campaign Initiatives page displays.
4. In the "Selected campaign" drop-down box select your campaign.
5. Add the campaign initiatives you created earlier to the "Selected Campaign Initiatives" box by clicking **Add**.
6. Add a starting date for your campaign.
7. Choose an end date or select **Run this campaign initiative indefinitely**.
8. Click **OK** at the bottom of the page.

Logon to the ToolTech store using the Organization you selected when creating customer profiles. You should now be able to see the specific products you selected in the ToolTech Home page.

Adding your organization's logo to ToolTech (Business Edition)

Business

You can add your organization's logo to the ToolTech header by replacing your logo's URL in the `businessaccount.xml` file before publishing your store.

To add your organization's logo, replace the URL with your logo's URL in the `businessaccount.xml` file:

```
<TermCondition>
<DisplayCustomizationTC sequenceNumber = "1">
<DisplayLogo>
<Attachment URL = "images/logoAccountA.gif" />
</DisplayLogo>
</DisplayCustomizationTC>
</TermCondition>
```

Where `images/logoAccountA.gif` represents where your logo's URL should be placed.

For more information on the information stored in the `businessaccount.xml` file, see the related concepts below.

Chapter 4. Sample store database assets

Store database assets

Store data is the information loaded into the WebSphere Commerce Server database, which allows your store to function. In order to operate properly, a store must have the data in place to support all customer activities. For example, in order for a customer to make a purchase, your store must contain a catalog of goods for sale (catalog data), the data associated with processing orders (tax and shipping data), and the inventory to fulfill the request (inventory and fulfillment data).

Data may be exclusive to a store, or shared between stores. For more information, see *Shared data assets*.

The store database assets in the sample store archives provided with WebSphere Commerce are well-formed, XML files valid for the Loader package, with the following exceptions: the store archive XML files are intended to be portable and should not contain generated primary keys that are specific to a particular instance of the database. Instead they use internal-aliases (outlined in the Store archive Loader conventions) which are resolved by the IDResolver at the time of publish. The store archives also use a set of DTD macros (known in XML as entities). The macros act as place holders for values you select in Store Services during store creation. The use of these two conventions allows the sample store archives to be copied and published multiple times.

The sample store archive includes all the database assets necessary to create a functional store. You can modify these files for use in your own store archive, or use them as a guide for creating your own XML files. WebSphere Commerce requires that certain data be loaded into the WebSphere Commerce database to create a functional store, and that this data must be loaded in the order determined by the schema. For example, the FFMCENTER table must be populated before the STOREENT table. Since the sample stores include all the mandatory data in the order and structure that WebSphere Commerce requires, using the database assets as a base, or guide, for your own store saves you a substantial amount of time during the initial creation period.

For a list of the database asset files used in the sample store archive, see *Sample store archive database assets*. For more detailed information on store data, see the *IBM WebSphere Commerce Store Developer's Guide*.

Note: The DTDs for the sample store database asset XML files are not in the store archive files. They are located in the following directory:

▶ NT

`drive:\WebSphere\CommerceServer\xml\sar`

▶ 2000

`drive:\Program Files\WebSphere\CommerceServer\xml\sar`

▶ AIX

`/usr/WebSphere/CommerceServer/xml/sar`

▶ Solaris

`/opt/WebSphere/CommerceServer/xml/sar`

▶ 400

`/QIBM/ProdData/WebCommerce/xml/sar`

Sample store access control database assets

In WebSphere Commerce all stores have access control policies. An access control policy authorizes users or a group of users to perform particular actions. There are two access control policy files for each store:

- *samplestorename*AccessPolicies.xml
- *samplestorename*AccessPolicies_locale.xml

Both the AccessPolicies.xml and AccessPolicies_locale.xml, are high level native access control files. AccessPolicies.xml is national language independent, whereas AccessPolicies_locale.xml is national language dependant. Each file includes the possible actions, action groups, resources, and policy definitions used in the sample stores. These files are transformed into the AccessPoliciesOut.xml, and AccessPoliciesOut_locale.xml files respectively. Each transformed file populates the database. For more information on transforming the access policy files, refer to the related links below.

Note: Only the transformed files can be mass loaded or used directly in the SAR file, the pre-transformed files cannot.

The database asset information for the sample stores can be divided into the following sections:

- Actions
- Resource Categories
- Resource Groups
- Action Groups
- Policy Definitions

Actions

Actions that can be performed under the access control policy are defined in each sample store's AccessPolicies.xml file.

Resource Categories

Resource categories define protectable resources.

Resource Groups

Resource groups contain the resources controlled by the access control policy. A resource group can include business objects such as "contract" or "exchange position," or a set of related commands. Each sample store's AccessPolicies.xml file defines the resource groups in the policy.

Action Groups

Action groups define the actions that can be performed on the resource groups in the access control policy. These groups are defined in the AccessPolicies.xml file for each store.

Policy Definitions

Each sample store's policy is defined in the individual store's AccessPolicies.xml file. The ToolTech sample store has two policies.

ToolTech Shipping database assets (Business Edition)

Business

The ToolTech shipping database assets are stored in the following XML files:

- shipping.xml
- store-catalog-shipping.xml
- store-defaults.xml
- shipfulfill.xml

ToolTech shipping database assets can be divided into the following sections:

- Jurisdictions
- Shipping modes
- Calculation codes
- Calculation rules
- Calculation scale
- Calculation range
- Calculation lookup
- Calculation combinations
- Shipping fulfillment

Jurisdictions

The `shipping.xml` file identifies jurisdictions for shipping. Jurisdictions are defined in the `JURST` table, `JURSTGROUP` assigns the jurisdiction to a group and subclass, and `JURSTPREL` assigns the jurisdiction and jurisdiction group to the same subclass.

Shipping modes

A shipping mode is a combination of a shipping carrier and its shipping service. For example, XYZ Carrier, Overnight, is a shipping mode. Information on shipping modes is stored in the `SHIPMODE` table.

Calculation codes

Calculation codes are used to calculate discounts, shipping charges, sales tax, weight ranges, and shipping tax. The `shipping.xml` file contains all the calculation codes for shipping. The `CALCODE` table stores the calculation codes for shipping. The `displaylevel` field indicates the amount calculated, and it's source.

0 = Order Item
1 = Order
2 = Product
3 = Item
4 = Contract

Calculation rules

Each calculation code has a set of calculation rules which define how the calculation will be done. For example, if you are shipping goods to one region you may apply certain rules to the calculation, and if you are shipping goods to another region, you may apply different rules to the calculation. The `CALRULE` table stores the calculation rules for shipping, and shipping by weight. The `flag` field specifies whether the `CalculationCodeQualifyMethod` value of the specific `CalculationCode` should be invoked.

0 = The method will not be invoked.

1 = The method will be invoked.

Calculation scale

A calculation scale is the set of ranges that applies to the calculation. For example, for shipping costs you may have a set of weight ranges that each correspond to a particular cost. That is, a product that weighs between 0 and 5 kg might cost \$10.00 to ship and a product weighing between 5 and 10 kg might cost \$15.00 to ship. The CALSCALE table stores the scale code for shipping and shipping by weight, one for “per order” and one for “per item”.

Note:The CALSCALE table stores the scale code for currencies if they apply to the corresponding range.

Calculation range

The range for the scale codes is stored in the CALRANGE table. calmethod_id_10 is used for “per order” shipping and calmethod_id_11 for per item shipping.

Calculation lookup

Calculation lookup values are the values associated with the calculation scale. There is one lookup value per currency for a given CALRANGE ID. The CARLOOKUP table defines the lookup ID and value.

Calculation combinations

Calculation rules and the scale ranges are combined in the CRULESCALE table. Calculation methods and rules are combined in the STENCALUSG table. The store default for calculations is also stored in this table. The usageflag field controls how the OrderPrepare command uses the calculation.

1 = use - use this CalculationUsage.

2 = check - throw an ECAApplicationException if this calculation does not produce a value for an order item.

Shipping fulfillment

Shipping fulfillment assets associate a shipping jurisdiction group to the calculation rules, and the fulfillment center to the shipmode for the store. The shipping fulfillment information is stored in the SHPJCRULE and SHPARRANGE tables and is displayed in shipfulfill.xml.

ToolTech Catalog database assets (Business Edition)

Business

The catalog.xml file stores the catalog information for the sample stores in WebSphere Commerce. Each sample store has its own catalog.xml file. For more information see the catalog.xml file.

Catalog assets can be divided into the following sections:

- Catalog groups
- Catalog entities

Catalog groups

Catalog groups are groups of categories and products. A category can be a catalog group in itself. For example, Woodworking is a grouping of the categories that make up Woodworking such as Saws and Sanders, while the category Sanders is a grouping of products.

In the ToolTech sample store there are several catalog groups:

- Woodworking
- Power
- Cordless
- Accessories

Catalog groups are created in the CATGROUP table and are associated with a specific catalog in the CATTOGRP table. Catalog groups can have top categories and subcategories. For example, a top category could be Woodworking, and a subcategory could be Saws. If you have a subcategory, you must put it under the top category in the CATGRPREL table.

Catalog entities

The ToolTech sample store catalog is comprised of catalog entities. These entities are comprised of the following:

- products
- items

Information about catalog entities is stored in the CATENTRY table. The CATENTREL table stores relationships between catalog entities such as product-item, bundle, and package relationships. For more information on packages and bundles, see Packages and bundles.

Products

For each product in the ToolTech store there is an entry in the BASEITEM table. In general, an entry in the BASEITEM table can map to multiple products. For the ToolTech store, each entry in the BASEITEM table is mapped to one product.

Each product can only have one version which is determined by an expiration date. Therefore, you can have only one Second Operation Lathe, with 500-3700 rpm, that expires on January 1, 2010. Version information, per product, is stored in the ITEMVERSN table.

In the ToolTech store, inventory is not shared with other sellers. As a result, every product in the catalog has only one entry in the DISTARRING table. Each row of the DISTARRANG table represents a distribution arrangement, enabling a store to sell its own inventory.

Items

An item is a specific instance of a product, defined by attributes. All items are in a category. The information held about a specific item is stored in the ITEMSPC table. Each item's attributes are stored in the ATTRVALUE table. Each item's inventory information is held in the RECEIPT table. The date that a backordered item is expected to be received is found in the RADETAIL table.

Note: In the ToolTech store, the catalog uses pricing for items in the OFFERPRICE table. For more information see the `offering.xml` file.

The time between release-to-fulfillment and shipping for an item is defined by the `shippingoffset` variable. For example, if the `shippingoffset` variable is equal to 86400 seconds and product `baseitem_id_260` is being shipped from fulfillment center 1, `product_baseitem_id_260` will be shipped 86400 seconds (24 hours) after being released to the fulfillment center.

ToolTech contract, business account, and business policy database assets (Business Edition)

Business

In WebSphere Commerce, all stores must have a default contract. The default contract is comprised of terms and conditions (also known as *terms*) which are stored in the `contract.xml` file. The ToolTech sample store has two accounts and four contracts. If the terms and conditions of each contract refer to a policy, the policy information is stored in the `businesspolicy.xml` file. Account information is stored in the `businessaccount.xml` file.

Note: All contracts are stored in the `contract.xml` file. There is a locale specific version of `contract.xml`.

Contract, account, and business policy database asset information can be divided into the following sections:

- Contract
- Account
- Business policy

Contract

The following four contracts can be found in ToolTech:

- Contract number 1234 (belongs to Organization A and Default Organization)
- Contract number 2345 (belongs to Organization A and account1)
- Contract number 3456 (belongs to Organization B and account2)
- Contract number 4567 (belongs to Organization B and account2)

Terms and conditions

All contracts in the sample stores must have terms and conditions. Terms and conditions are the set of rules agreed upon in a contract, and control the purchasing process between a buyer and a seller. For every contract you must have one set of terms for the price. For more information on the specific terms and conditions for each contract and account, see the related links below.

For contract number 2345, all products in the catalog have 20 percent off the standard price. If an order totals more than 1000 dollars , the order must be approved by the Buyer Approver, under the `OrderApprovalTC`. The total sum of all the orders made under this contract should not exceed \$500,000, under the `RightToBuyTC`. Customers cannot spend less than \$100 for the life of the contract under the `ObligationToBuyTC`.

Note: In this contract customers can only spend between \$5,000 to \$500,000 for the life of the contract. Whenever the customer's total purchases reach an amount above \$500,000, a message will be displayed stating that they have exceeded the spending limit for the contract. The `ObligationToBuyTC` is not enforced.

Note: Only customers with Buyer Approver and a Buyer (buy-side) role can approve orders.

Contract number 3456 states that Organization B 's address will be used as the billing address. All products in the Woodworking, Power, and Cordless categories have 25 percent off the standard price. For contract number 4567, products in the Woodworking and Power top categories have a 50 percent discount off the standard price.

You can also have terms and conditions for other types of charges such as shipping charges. Each contract can have only one set of terms and conditions for shipping charges. The buyer and seller must be referred to in the `contract.xml`.

Note: Certain terms and conditions can only be specified at the account level, and are not available in the contract.

Account

Terms and conditions

Each account in the ToolTech store is associated with a contract, and has terms and conditions associated with it. In the ToolTech store, one account can be associated with the default contract, and the other cannot. In the ToolTech store there are two accounts:

- Account1- for Organization A. This account is associated with the default contract, and has purchase order terms and conditions.
- Account2- for Organization B. This account can have payment terms and conditions, and allows a credit line to be used.

The accounts define purchase order terms and invoice terms. The following example defines the formatting and delivery method for invoices for an account, and specifies a blanket purchase order number that can be specified when placing orders for the account. These terms are specified in the `businessaccount.xml` file.

If customers make a purchase using contract 2345, they are required to enter a purchase order (PO) number, outlined in the `PurchaseOrderTC`. A blanket PO number must be entered, unless an individual purchase order number was created. Customers are allowed individual PO numbers, however each number is unique and can only be used for a single order. The buyer contact, seller, and account holder must be referred in the `businessaccount.xml`.

Business policy

Terms and conditions

A business policy outlines the rules followed by a business for a particular business process. If the terms and conditions of your contract refer to a business policy, you must populate `businesspolicy.xml` before importing the `contract.xml` file.

Note: Some policies referenced in `contract.xml` are not contained in `businesspolicy.xml`. These policies are part of the bootstrap data. For more information on the bootstrap data see Bootstrap Files.

ToolTech Organization database assets (Business Edition)

In the ToolTech sample store, there are two organizations that get populated in the `organization.xml` file. These organizations are used by the `businessaccount.xml` file.

The following two organizations can be found in ToolTech:

- Organization A
- Organization B

Organizations are setup in the `organization.xml` file and are defined in the `ORGENTITY` table.

The organization address which is used in contracts for the billing to, and shipping to address, is also populated in the `organization.xml`file.

Note: You cannot publish the ToolTech store more than once using the same organization.

ToolTech sample store organizations (Business Edition)

Business

The ToolTech sample store comes with the ability to shop in different organizations. This allows customers in separate organizations to see different prices for the same products.

When the ToolTech store is published through store publish, the two following Buyer Organizations are created:

- Buyer Organization A
- Buyer Organization B

For each of the Buyer Organizations, an account is also created. Each account has at least one contract published with it. Buyer Organization A allows customers to shop under the default contract. When customers logon to either organization they will see the organizations logo on the right-hand corner of the store header, and an account-based promotion (if it has been set up for that organization) on the Home page. Customers can setup promotions through the WebSphere Commerce Accelerator based on accounts and see a different product on promotion when they login with different organizations. For more information on adding organization's logos, contracts, and accounts, and promotions, see the related links below.

Default organization

When customers register in the ToolTech store and do not provide an organization they are automatically shopping under the default organization. Customers in this case will not be registered as business users. Contract number 1234 belongs to the default organization.

Buyer Organization A

Customers shopping under Buyer Organization A will notice the following conditions when shopping in the ToolTech store:

- Buyer Organization A allows customers to use contracts 1234 and 2345.
- If customers logon as Buyer Organization A and place an order using contract 2345, they have a pre-selected shipping address and shipping method.
- Buyer Organization A has a blanket purchase order number under contract 2345, which is B1234567.

Note: For more information on blanket purchase orders see the related links below.

Buyer OrganizationB

Customers shopping under Buyer Organization B will notice the following conditions when shopping in the ToolTech store:

- Buyer Organization B allows customers to use contracts 3456 and 4567.
- If customers logon as Buyer Organization B and place an order using contract 3456, on checkout they will have a pre-selected billing address.

ToolTech sample store purchase orders (Business Edition)

Business

The ToolTech sample store comes with two types of (PO) purchase orders:

- Blanket purchase orders

- Individual purchase orders

In the ToolTech store, Organization A has a blanket PO number of B1234567. Customers can use this number for as many purchases as they wish. Any other number besides this one is considered to be an individual PO, and can only be used for one shopping flow.

ToolTech sample store contracts and accounts (Business Edition)

Business

The ToolTech sample store has the following four contracts:

- Contract number 1234 (belongs to Default Organization. Account1 allows the default contract.)
- Contract number 2345 (belongs to Organization A and account1)
- Contract number 3456 (belongs to Organization B and account2)
- Contract number 4567 (belongs to Organization B and account2)

Each contract belongs to an organization and account, and has terms and conditions associated with it. For more information on terms and conditions see the related links below.

Note: Buyer Organization A has Account 1 which allows the default contract, even though the contract itself belongs to the Default Organization. This contract can be reused by any organization, if the account allows it.

Contract 1234

This is the store default contract. Contract 1234 has the following terms and conditions:

- ShippingTC - Shipping charges are paid to the seller.
- PriceTC - All products have the standard price.

Contract2345

Contract 2345 has the following terms and conditions:

- PriceTC - All products in the catalog have a 20% discount off the standard price.
- OrderApprovalTC - If an order totals more than \$1000 USD it must be approved by the Buyer Approver.
- RightToBuyTC - The sum of all orders made under the life of this contract cannot exceed \$500,000 USD.
- ObligationToBuyTC - Customers cannot spend less than \$100 for the life of the contract.
- PurchaseOrderTC - Customers must enter a purchase order number. (Belongs to Account1)
- ShippingTC - The shipping address (OrganizationA_NickName1) and shipping method are pre-selected. Shipping charges are paid to the seller.

Note:Only customers with a Buyer Approver and a Buyer (buy-side) role can approve orders. For more information on these roles, see the related links below. For more information on purchase orders, see the related links below.

Note:There are three shipping terms and conditions associated with contract 2345. These are ShippingTCShipToAddress, ShippingTCShippingMode, ShippingTCShippingCharge respectively.

Contract3456

Contract 3456 has the following terms and conditions:

- PriceTC - All products in Woodworking, Power, and Cordless categories have a 25% discount off the standard price.

- PaymentTC - VISA, MasterCard, and American Express are the only allowed payment methods. Organization B's address (OrganizationB_NickName1), is used as the billing address.
- ShippingTC - Shipping charges are paid to the seller.

Contract4567

Contract 4567 has the following terms and conditions:

- PriceTC - Products in the Woodworking and Power categories have a 50% discount off the standard price.
- PaymentTC - The payment methods allowed are VISA, MasterCard, and American Express, and credit line. Credit line belongs to Account2. Organization B's address (OrganizationB_NickName1), is used as the billing address.
- ShippingTC - The shipping address (OrganizationB_NickName1) and shipping method are pre-selected. Shipping charges are paid to the seller.

Notes:

If customer's orders have one item using contract 3456 and another item using 4567, only the credit card method of payment is available.

To buy items using a credit line, contract4567 must be selected.

The contract with the lowest price for each item is automatically selected.

Both account1 and account2 have the DisplayCustomizationTC which displays each organizations logo on the right side of the ToolTech header after logon. For more information see the related links below.

All terms and conditions are set up in the corresponding XML files. For more information see, " ToolTech contract, account, and business policy database assets".

Chapter 5. ToolTech store pages

ToolTech store pages (Business Edition)

► Business

The ToolTech sample store provides all the pages necessary for a functioning store. You can customize these pages for use in your own store, or use them as guides for creating your own store pages.

The ToolTech store includes the following store pages:

- Sample stores
- ToolTech Sample Store
- ToolTech Address Book pages
- ToolTech Search pages
- ToolTech Billing and Shipping Information page
- ToolTech Catalog Item Display page
- ToolTech Main Category page
- ToolTech Product Display page
- ToolTech Subcategory page
- ToolTech Home page
- ToolTech Confirmation page
- ToolTech Product Availability page
- ToolTech Error pages
- ToolTech Header
- ToolTech Logon pages
- ToolTech Left navigation frame
- ToolTech Order Summary page
- ToolTech Order Status page
- ToolTech Order Status Codes
- ToolTech Your Order page
- ToolTech Quick Order page
- ToolTech Registration pages
- ToolTech Add to Requisition List page
- ToolTech New Requisition List page
- ToolTech View Requisition List page
- ToolTech Requisition List page
- ToolTech Edit Requisition List page
- ToolTech Collaborative Workspaces page
- ToolTech collaborative workspaces Members page

The sample store pages are included in the ToolTech store archive file, `tooltech_locale1_locale2.sar`, which is located in the following directory:

► NT

drive:\WebSphere\CommerceServer\samplestores\ToolTech

▶ 2000

drive:\Program Files\WebSphere\CommerceServer\samplestores\ToolTech

▶ AIX

/usr/WebSphere/CommerceServer/samplestores/ToolTech

▶ Solaris

/opt/WebSphere/CommerceServer/samplestores/ToolTech

/opt/WebSphere/CommerceServer/samplestores/ToolTech

▶ 400

/QIBM/ProdData/WebCommerce/samplestores/ToolTech

After you have published a store archive based on the sample store, the store pages are located in the following directory:

▶ NT

drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear\wcstores.war\storedir

▶ 2000

drive:\Program

Files\WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear\wcstores.war\storedir

▶ AIX

/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/storedir

▶ Solaris

/opt/WebSphere/Appserver/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/storedir

/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/storedir

▶ 400

/QIBM/UserData/WebASAdv4/WAS_instance_name/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/storedir

For more information on the technical details of each page, click the corresponding link.

Important: In order to work with WebSphere Commerce Studio, the sample store pages use the following syntax to indicate the URL to access a command:

`action="<%= "AddressAdd"%>"` where AddressAdd is a URL command

instead of the more commonly used syntax:

`action="AddressAdd"`

ToolTech Address book pages (Business Edition)

▶ Business

The ToolTech address book pages allow customers to add addresses, including shipping and billing addresses, to an address book.

When customers are logged on, they can click on the **Account** link on the header, to bring them to the option to work with their address book. Once they click on **Address Book**, they will see a list of addresses in their address book, options to edit and delete each address, and a button to add new addresses.

The sample store address book procedures uses the following JSP files:

- UserAccount.jsp (Account page)
- AddressBook.jsp (Address Book page)
- Address.jsp(Add Address page and Edit Address page)

Note: Address.jsp is used for both the Add Address page and the Edit Address page. If the addressId is provided, the Address.jsp loads as an Edit Address page. Otherwise, it loads as an Add Address page. If the addressId is provided as a parameter for the AddressAdd command, the command updates the address of the specified addressId. Otherwise, a new address is created.

Commands

UserAccount.jsp uses the following commands:

- AddressBookForm
- UserRegistrationForm

Address.jsp uses the following commands:

- AddressUpdate
- AddressAdd

AddressBook.jsp uses the following commands:

- AddressForm

Beans

AddressBook.jsp uses the following beans:

- AddressBookDataBean
- AddressAccessBean
- OrgEntityDataBean

Address.jsp uses the following beans:

- ErrorDataBean
- AddressDataBean

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see Sample store pages: common implementation techniques.

When customers click on **Edit Address Book** from the Account page, the AddressBookForm command is called. AddressBookForm then loads the Address Book page (AddressBook.jsp). When a customer clicks on **Change Personal Information**, the UserRegistrationForm command is called and the Update Registration page is loaded.

After customers complete the fields in the Add Address page (Address.jsp), the system checks to see if the addresses exist. If the address entered exists, the address book is updated. If not, a new address is created, for example:

```
if (addressId != null)
bUpdateAddress = true;
else
bUpdateAddress = false;
```

When customers complete entering the address in the Add Address page, (Address.jsp), and click on **Submit**, the AddressAdd command is called. When customers complete entering the address in the Edit Address page,(Address.jsp), and click on **Submit**, the AddressUpdate command is called.

```
< FORM name="AddressForm" method="GET" action="AddressUpdate" >
}
else
{

< FORM name="AddressForm" method="GET" action="AddressAdd" >
}
```

Error handling

If the customer does not complete the required fields in either the Add Address page (Address.jsp) or the Edit Address page (Address.jsp), the system tells the customer to re-enter the fields. The Address_[locale].properties file determines what fields in the address each locale specifies as mandatory.

The error messages that appear to customers are the following:

```
AddAdd_ERROR1 = Type a nickname in the Nickname field.
AddAdd_ERROR2 = The nickname you entered already exists, please enter a different one.
AddAdd_ERROR3 = Type a name in the Last name field.
AddAdd_ERROR4 = Type an address in the Address field.
AddAdd_ERROR5 = Type a city name in the City field.
AddAdd_ERROR6 = Type a state name in the State/Province field.
AddAdd_ERROR7 = Type a country/region name in the Country/Region field.
AddAdd_ERROR8 = Type a ZIP or postal code in the ZIP/Postal code field.
```

ToolTech search pages (Business Edition)

Business

The search results page, ResultList.jsp is displayed after a customer has entered a search request from the left navigation panel, NavSideBar.jsp, or the advanced search page, AdvancedSearch.jsp.

Beans

ResultList.jspuses the following beans:

- CatalogDataBean
- CategoryDataBean
- CatEntrySearchListDataBean
- CatalogEntryDataBean
- CatalogEntryDescriptionAccessBean
- CatalogEntryAccessBean
- AttributeValueDataBean
- AttributeDataBean
- ItemDataBean

AdvancedSearch.jsp uses the following beans:

- CatalogDataBean
- CategoryDataBean

Commands

ResultList.jsp uses the following commands:

- ProductDisplay
- AdvancedSearchView
- CatalogSearchResultView

AdvancedSearch.jsp uses the following commands:

- CatalogSearchResultView

NavSideBar.jsp uses the following search commands:

- CatalogSearchResultView
- AdvancedSearchView

For other commands used by NavSideBar.jsp, see ToolTech Left navigation frame.

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see Sample store pages: common implementation techniques.

When the customer clicks **GO** from the left navigation panel, NavSideBar.jsp, or clicks **Submit** from the advanced search page, AdvancedSearch.jsp, the ResultList.jsp page displays.

When the CatEntrySearchListDataBean is activated from SearchResults.jsp, it retrieves all of the results for the search from the database:

```
com.ibm.commerce.beans.DataBeanManager.activate(catEntSearchListBean, request);
```

The request parameter is the URL string that is passed to the ResultList.jsp page. In the CatEntrySearchListDataBean, the method getResultList() returns a CatalogEntryDataBean array. Each CatalogEntryDataBean is one result that is listed on the search result page. The getResultList() method returns a specific number of results per page. This number is specified in the URL request address with the *pageSize* variable when the CatEntrySearchListBean is activated. There may be several results matching the user's search criteria. When the results are displayed, customers can navigate through all the results using **Previous** and **Next**. To achieve this functionality, the URL parameter beginIndex is passed to the catEntrySearchListDataBean telling it to retrieve the results starting at the beginIndex number. For example, if the first page starts at zero, beginIndex will be equal to zero. If customers choose 10 results per page, when they click **Next**, beginIndex will be equal to 10.

In the NewFashion sample store only products are displayed in search results. In the ToolTech sample store customers have a choice of what is displayed depending on the search criteria indicated on the Advanced Search page. When the CatalogSearchResultView command is called, the resultType variable is passed to the command as an URL argument. The resultType variable tells the CatEntrySearchListBean to search for and return products, items, or both.

- 1 = List items only.
- 2 = List products only.
- 3 = List both items and products.

The following code lists products only:

```
<input type="hidden" name="resultType" value="2">
```

Limitations

The ToolTech store catalog follows a hierarchical structure with top categories at the top, such as Woodworking , Power, Cordless, and Accessories, and subcategories beneath them, such as Saws, and Lathes.

When customers select **Advanced Search**, if they choose to search in a category, they can only search under the specific top or subcategory chosen. For example, if customers search under the Woodworking category, only products directly under that category will be displayed. Products under the Woodworking, Saws, sub-category will not be displayed. To search the Saws subcategory, customers must select that specific category in order for search results to be displayed.

ToolTech Billing and Shipping Information page (Business Edition)

Business

After customers have clicked **Next** from the Your Order page, the Billing and Shipping Information page, `Shipping.jsp`, displays. For each item in the order, `Shipping.jsp` displays the quantity, contract name, description including attributes, shipping address, and shipping method. From this page customers can do the following tasks, depending on the terms and conditions specified in the order's contracts:

- Select a billing address.
- Select a shipping address for each item in the order.
- Select a shipping method for each item in the order.

Commands

`Shipping.jsp` uses the following commands:

- `OrderCopy`
- `AllocateCheck`
- `AddressBookForm`

Beans

`Shipping.jsp` uses the following beans:

- `OrderDataBean`
- `UseablePaymentTCListDataBean`
- `AddressAccessBean`
- `OrderItemDataBean`
- `ShippingModeAccessBean`
- `AddressDataBean`

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see [Sample store pages: common implementation techniques](#).

When customers click on **Next** from the Your Order page, the OrderItemDisplay command is called with the parameter page=shipmethod. OrderItemDisplay calls the OrderItemDisplayViewShiptoDsp view command, which forwards to OrderItemDisplay.jsp. OrderItemDisplay.jsp includes shipping.jsp because the page parameter is shipmethod.

The UsablePaymentTcListDataBean retrieves the payment information from the payment terms and conditions.

Note: The payment terms and conditions correspond to the payment methods chosen by customers on the Your Order page.

If no billing address is specified in the payment terms and conditions of the contract, customers can select an address from the drop-down list. The drop-down list contains addresses from the customer's address book. The AddressAccessBean retrieves the addresses as follows:

```
// ask buyer to select a billing address if one is not specified in the payment TC
if (!bBillAddressInTC) {
    // Get the list of addresses
    AddressAccessBean abAddressFinder = new AddressAccessBean();
    Enumeration enAddressList = abAddressFinder.findByMemberId(userRef);
```

If customers wish to add a new address, they can click on **Address Book** which launches the AddressBookForm command, and displays the Address Book page, Addressbook.jsp.

When customers click on **Next**, the ShipMethodForm is submitted and calls the OrderCopy and AllocationCheck commands. OrderCopy sets the billing address and updates the order items with the selected shipping addresses and shipping methods. AllocationCheck determines if some of the order items are not available, in which case Product Availability page is displayed; otherwise the Order Summary page is displayed.

ToolTech Main Category page (Business Edition)

Business

When customers click **Browse Catalog** from the Home page, CatalogMainDisplay.jsp, the Main Category page, CatalogTopCategoriesDisplay.jsp is displayed.

Commands

CatalogTopCategoriesDisplay.jsp uses the following commands:

- CategoryDisplay
- ProductDisplay

Beans

CatalogTopCategoriesDisplay.jsp uses the following beans:

- CatalogDataBean
- CategoryDataBean
- ProductDataBean
- ItemDataBean

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see Sample store pages: common implementation techniques.

When customers click **Browse Catalog** from the Home page, `CatalogMainDisplay.jsp`, the `CategoryDataBean` retrieves the top categories and the Main Category page, `CategoryTopCategoriesDisplay.jsp`, is displayed showing the titles of categories and subcategories.

If customers are in a pending approval state, they cannot log on to see the catalog. In this state, they can only go to the Account Page and change personal information and edit their address book. If customers are approved or have no pending approval state, the regular catalog is displayed.

ToolTech Product Display page (Business Edition)

Business

When customers select a product, item, package, or bundle from the Subcategory page `CategorySubcategoriesDisplay.jsp`, the Product Display page `CatalogProductDisplay.jsp`, is displayed.

Commands

`CatalogProductDisplay.jsp` uses the following commands:

- `ProductDisplay`

Beans

`CatalogProductDisplay.jsp` uses the following beans:

- `ProductDataBean`
- `AttributeAccessBean`
- `ItemAccessBean`
- `CategoryDataBean`

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see [Sample store pages: common implementation techniques](#).

When customers select a product from the Subcategory page, `CategorySubcategoriesDisplay.jsp`, the `ProductDisplay` command loads the Product Display page, `CatalogProductDisplay.jsp`. The `ItemAccessBean` gets the items from the product and the `AttributeAccessBean` gets the attributes. The items and attributes associated with the product are then displayed on the page.

```
//AttributeAccessBean attributes[] = product.getAttributes(); //AttributeAccessBean attribute;

// get items from product

ItemAccessBean iabs[] = product.getItems(); ItemAccessBean iab;
CatalogEntryDescriptionAccessBean cedab;

CategoryDataBean parentCategory = null;

if (parentCategoryId != null) { parentCategory = new CategoryDataBean ();
parentCategory.setCategoryId(parentCategoryId);
com.ibm.commerce.beans.DataBeanManager.activate(parentCategory, request); } %>
```

When customers click an item, on the Catalog Item Display page, the `CatalogItemDisplay.jsp` page is displayed.

- CatalogSubCategoriesDisplay.jsp(Subcategory Display page)
- CatalogTopCategoriesDisplay.jsp(Main Category page)

Commands

CatalogMainDisplay.jsp uses the following commands:

- LogonForm
- TopCategoriesDisplay
- MerchantCenterView

Beans

CatalogMainDisplay.jsp uses the following beans:

- CatalogDataBean
- CategoryDataBean
- UserRegistrationDataBean
- TopCategoriesDisplay
- RoleDataBean

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see Sample store pages: common implementation techniques.

The CatalogMainDisplay.jsp, checks to see if registered customers have a Buyer Approver role. If so, the Home page loads. If customers are guests and are not registered, they are re-directed to the Logon page.

When approved customers click **Browse Catalog** from the Home page, the TopCategoriesDisplay command is called and the Main Category page CatalogTopCategoriesDisplay.jsp, is displayed where customers can browse the catalog.

The LogonForm.jsp. page is launched by the index.jsp page which provides a URL for invoking the homepage of the sample store. The index.jsp page calls the parameters.jsp file which contains the parameters needed to launch the store.

If customers have the role of Buyer Administrator, they will be able to view the **Go to Approval Tool** link to begin the approval process.

```
function ApprovalToolLink() { //checkBrowser() function is called in NavSideBar.jsp when this
page is loaded. //bRightBrowser variable will be populated. if (bRightBrowser) {
RFQwindow=window.open('<%=ApprovalToolLinkURL%>'); } else {
window.location.href('<%=BrowserVerErrorURL%>'); }
```

The Buyer Administrator approves users and assigns customers the Buyer (buy-side) role to allow them to make purchases in the ToolTech store. This role is also responsible for maintaining the buyer organization information.

Note:The Approval tool will only be displayed if the customer has a Buyer Administrator or Buyer Approver role.

If customers are in a pending approval state, they cannot log on to see the catalog, and are re-directed to the Account page. If customers are approved or have no pending approval state, the regular catalog is displayed.

```
if (userState.equals("0")) { // User is pending approval - cannot logon %> <center> <font
class="text">User State: Pending Approval<br> <a
href="LogonForm?langId=<%=languageId%>&storeId=<%=storeId%>&catalogId=<%=catalogId%>&page=sidebar">Logon</a>
</font> </center> <% } else if ((userState.equals("1") || userState.equals("")) &&
bnRegUser.findUser()) {
```

If customers have logged on using Organization A or B, they will see their organization's logo in the right hand corner of the page header.

ToolTech Confirmation page (Business Edition)

Business

After customers have clicked **Order Now** from the Order Summary page, the OrderOKView command forwards to the Order Confirmation page and the Confirmation.jsp page is displayed. The Order Confirmation page displays the order number, subtotal, total tax, shipping charges, grand total, and estimated shipping date for the whole order. If two orders are processed, the information for both orders will be displayed on the confirmation page.

Beans

confirmation.jsp uses the following beans:

- PayStatusPMDDataBean
- OrderDataBean

Implementation

The PayStatusPMDDataBean retrieves the payment status information for the order. The OrderDataBean retrieves the order information from the system.

ToolTech Product Availability page (Business Edition)

Business

The ToolTech Product Availability page CheckProdAvail.jsp, displays to customers which items in an order are not currently available, and the expected shipping date for those items.

If some of the items are not in stock, customers can select a shipping preference allowing them to split the order and purchase the available items, or wait until all the items are available. Customers also have the option to remove an item from the order.

Note: This page displays when some of the order items are unavailable.

Commands

CheckProdAvail.jsp uses the following commands:

- ProductDisplay
- OrderPrepare
- OrderItemUpdate
- OrderItemMove
- OrderDisplay
- OrderItemDisplay

Beans

CheckProdAvail.jsp uses the following beans:

- OrderDataBean
- OrderItemDataBean
- ItemSpecificationDataBean

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see [Sample store pages: common implementation techniques](#).

The Product Availability page only displays when some of the order items are not available. The AllocationCheck.jsp page finds out if any items are not available.

If all items are unavailable or all items are available, the AllocationCheck.jsp page forwards to the OrderDisplay command which is associated with OrderDisplayPending.jsp; otherwise it includes the CheckProductAvail.jsp page.

The CheckProductAvail.jsp page allows customers to view the availability date for each item in the order. If some items in the order are not in stock, the customer selects a shipping option from the **Choose shipping preference** options for the order.

Customers can select from two shipping options:

1. Wait until the entire order is ready for shipping.
2. Ship the in-stock items now and ship the rest when they are available.

When the customer selects the second option, the OrderItemMove command splits the order into two separate orders.

ToolTech error pages (Business Edition)

Business

The ToolTech sample store has the following error pages:

- BrowserVerError.jsp (Browser Version Error page)
- GenericError.jsp (Generic Error page)
- InvalidQuantityError.jsp (Invalid Quantity Error Page)
- MergeError.jsp (Merge Error Page)

Implementation

The Browser Version Error page BrowserVerError.jsp, displays when customers use the incorrect version of Microsoft Internet Explorer after clicking the **Buyer Approver** link or the **RFQ Requests** link. The following error message is displayed:

You have attempted to access a page using a Web browser that is not supported. To access this page, you must use Microsoft Internet Explorer 5.5 or higher. You can download this version of the Web browser by going to www.microsoft.com/downloads.

The Generic Error page GenericError.jsp, displays when there are system problems with the ToolTech store. The following error message is displayed:

The store is currently experiencing problems. Try again later.

The Invalid Quantity Error page `InvalidQuantityError.jsp`, displays when customers attempt to add a product to the shop cart from the Product Display page, and enter an invalid quantity, such as a letter, in the **Quantity** field.

The Merge Error page `MergeError.jsp`, displays when the system attempts to merge more than one order together with different payment methods. Merging usually occurs when customers try to access the Your Order page.

When customers attempt to access a page that is protected by access control, the access control pages are caught by the error message key `_ERR_USER_AUTHORITY`. If registered customers attempt to access a page protected by access control, the following error message displays:

You are not authorized to access the page.

If guest customers attempt to access a page protected by access control, the following error message is displayed:

You must login before you can access the store.

ToolTech header (Business Edition)

Business

The ToolTech sample store has a header at the top of each page (`NavHeader.jsp`) that includes links to the following:

- Home
- Catalog
- Account
- Current Order
- Order Status
- Requisition List
- Logoff
- Collaborative Workspaces

When customers are logged in, registered and approved, the above links are displayed otherwise, the **Account** and **Logoff** links are the only links displayed. If customers have added their organization's logo to the `businessaccount.xml` file, they will see it on the right-hand side of the header. For more information on adding your organization's logo to ToolTech see the related links below.

Note: The **Collaborative Workspaces** link is only displayed to registered customers if the collaborative workspaces feature is enabled in Store Services. For more information on collaborative workspaces, see the Implementation section below.

Commands

`NavHeader.jsp` uses the following commands:

- `StoreCatalogDisplay`
- `TopCategoriesDisplay`
- `LogonForm`
- `RequisitionListDisplay`
- `OrderItemDisplay`
- `TrackOrderStatus`

- Logoff
- StoreCollabListDisplay

Beans

NavHeader.jsp uses the following beans:

- StoreDataBean
- CatalogDataBean
- UserRegistrationDataBean

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see Sample store pages: common implementation techniques.

When NavHeader.jsp is loaded, the UserRegistrationBean filters out the links that customers do not have access to. Only the specific features that customers have access to will be able to see them on the header. When each link is clicked, it takes customers to the appropriate page.

When customers click **Home**, the StoreCatalogDisplay command loads the CatalogMainDisplay.jsp page.

When customers click **Catalog**, the TopCategoriesDisplay command loads the Main Category Display Page CatalogTopCategories.jsp.

When customers click **Account**, the LogonForm command loads the LogonForm.jsp, which then redirects to LogonDisplay.jsp.

When customers click **Current Order**, the OrderItemDisplay command loads the Your Order OrderItemDisplay.jsp page.

When customers click **Order Status**, the TrackOrderStatus command loads the TrackOrderStatus.jsp page.

When customers click **Requisition List**, the RequisitionListDisplay command loads the Requisition List page.

When customers click **Collaborative Workspaces**, the StoreCollabListDisplay command loads the Collaborative Workspaces, StoreCollabListDisplay.jsp page. The **Collaborative Workspaces** link is contained within the body of a custom tag (the *ifEnabled* tag), and can be enabled or disabled based on what option is selected in Store Services. As long as the custom tag remains in place, Store Services can be used to automatically switch between a site that has collaboration support and one that does not, without having to modify the JavaServer page. To permanently enable or disable collaboration support in the page, the custom tags and the **Collaborative Workspaces** link can be removed from the JavaServer page by clicking **Apply Permanently**, in the Store Services GUI. It is not recommended to manually remove or alter the custom tag or link it surrounds. Instead, use the **Apply Permanently** button in Store Services.

Note: The custom tag that surrounds the **Collaborative Workspaces** link cannot be copied into the JavaServer pages of other stores. These tags are only intended to work in the store that originally contained them. For more information on collaboration, see the related links below.

When customers click **Logoff**, they are logged off. Once customers logoff, their registered information is not available and they are not able to browse the catalog or order items. They are treated as guest shoppers. To be able to shop once again, customers have to logon.

Note: If the store has enabled the customer care feature, it will check for customers names if they are registered. Customers names are then sent to the customer care applet to be shown on both the Customer Service Representative side and on the customer's side. Since the NavHeader.jsp page is included on every page in ToolTech, a part of the code also determines whether the page is personalized or not. If a JavaServer page, such as UserRegistrationUpdate.jsp, is a personal page, code is added on the UserRegistrationUpdate.jsp page before it includes the NavHeader.jsp.

```
<flow:ifEnabled feature="customerCare">

<%

// Set header type needed for this JSP for LiveHelp. This must
// be set before NavHeader.jsp

request.setAttribute("liveHelpPageType", "personal");

%>

</flow:ifEnabled>

<%

String incfile;

incfile = includeDir + "NavHeader.jsp";

%>

<jsp:include page="<%=incfile%>" flush="true"/>
```

When this code sees that the request is for a personal page, the Customer Service Representative will not be allowed to view the pages customers are on. For more information on the Customer Care feature, see the “Related Concepts” below.

ToolTech logon pages (Business Edition)

Business

The sample store logon pages allow registered customers to log in. The logon page is the first point of entry into the ToolTech store. Registered customers can enter their user ID and password to gain entry to the store, or select a link if they have forgotten their passwords. New customers can select a link to register for the first time. All customers can select the language to be displayed in the store.

The sample store logon procedure uses the following files:

- Logoff.jsp (Includes the parameters for the Logoff command. but is not displayed to the customer)
- LogonDisplay.jsp(Logon Page)
- LogonForm.jsp (Includes the parameters for the Logon command, but is not displayed to the customer)
- LogonResetPassword.jsp(Forget Your Password page)
- LogonResetPasswordNotify.jsp (A panel that formats the e-mail notification that gets sent to customers)
- LogonResetPasswordChange.jsp (A panel to change the customer's password. If a customer's password has expired, this page appears after the customer has logged in.)

Commands

LogonDisplay.jsp uses the following commands:

- Logon
- LogonForm
- ResetPasswordForm
- UserRegistrationForm

LogonResetPassword.jsp uses the following commands:

- ResetPassword
- LogonForm

LogonResetPasswordChange.jsp uses the following commands:

- ResetPassword

Beans

LogonDisplay.jsp uses the following beans:

- SupportedLanguageAccessBean
- LanguageDescriptionAccessBean
- ErrorDataBean
- CatalogDataBean

LogonResetPassword.jsp uses the following beans:

- ErrorDataBean

Implementation

Note:For more information on implementation techniques common to all sample store pages, including multicultural information, see [Sample store pages: common implementation techniques](#).

When customers enter a logon ID and password and click **Submit**, the Logon command is called. If the logon is successful, the StoreCatalogDisplay command is executed and the CatalogMainDisplay.jsp page is displayed. If the logon is not successful, the LogonForm command is called and the Logon Page is reloaded and displays an error message. If customers logon as Buyer Organization A or B, they will see their organization's logo on the right hand corner of the header. For more information on adding an organization's logo to the store, see the related references below.

If customers forget their password and click **Forget Your Password?**, the ResetPasswordForm loads the Reset Password page, LogonResetPassword.jsp. The ResetPassword command resets the customers passwords and calls the ResetPasswordForm command. After passwords have been reset, an e-mail notification is sent to customers informing them of the new password.

If customers passwords have expired, after they complete their login on the LogonDisplay.jsp page, they will be forced to change their password and be transferred to the LogonResetPasswordChange.jsp page.

When customers are notified to change a reset password, the ResetPasswordForm command loads the LogonResetPasswordChange.jsp page. If the process is successful, the LogonForm command is called and the LogonForm.jsp displays the Account page, otherwise the ChangePassword command is called and the LogonResetPasswordChange.jsp page is re-displayed.

When customers click **Register**, the UserRegistrationForm command gets called and the Registration page is displayed.

When customers select a language, the LogonForm command is called to switch the language and reload the Logon page. The LanguageDescriptionAccessBean gets the display name of the language, in the language that was selected by the customer. The SupportedLanguageAccessBean and the LanguageDescriptionAccessBean retrieve all of the supported languages within the store from the database, and display them to customers on the Logon page.

Error handling

When there is a problem with the Login, the LogonForm command will be called. LogonForm reloads LogonDisplay.jsp (via LogonForm.jsp) and displays an error message describing the problem.

The type of login problems that can occur are the following :

- Missing user IDs or passwords.
- Invalid user IDs or passwords.
- The customer associated with the user ID has been denied access, or does not have authorization.

If there is a problem with resetting a password or changing a password, the LogonResetPasswordChange.jsp page and the LogonResetPassword.jsp page check for the following errors:

- Missing parameters.
- Invalid passwords (Depends on the store's password policy. This includes minimums and maximums for length, consecutive letters, and digits).

ToolTech left navigation frame (Business Edition)

Business

The left navigation frame (NavSideBar.jsp) in the ToolTech store allows registered customers to do the following:

- Search the catalog for items.
- Link to the Advanced Search page.
- Enter SKUs and items for a quick order.
- Launch the RFQ Tool.
- Link to the Interest List (for RFQ) page.
- Enable live chat with customer assistance

Note: The **Search**, **Advanced Search**, and **Quick Order** links can be seen by all registered customers. The Request for Quote section, including **Interest List (for RFQ)** and **Launch RFQ Tool** can only be seen by customers with the Buyer (buy-side) role. The **Live Chat with Customer Assistance** link, can only be seen if enabled in the store through Store Services. Guest shoppers cannot see any of the links on the left navigation frame.

Commands

NavSideBar.jsp uses the following commands:

- CatalogSearchResultView
- AdvancedSearchView
- OrderItemAdd
- QuickOrderView
- InterestItemDisplay

Beans

NavSideBar.jsp uses the following beans:

- InterestItemAccessBean
- RoleDataBean
- UserRegistrationDataBean

Implementation details

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see Sample store pages: common implementation techniques.

When customers enter a keyword in the Search field and click **GO**, the CatalogSearchResultView command submits the search criteria and the ResultList.jsp page is displayed with the results of the search.

If customers click on the **Advanced Search** link, the AdvancedSearchView command loads the Advanced Search page.

When customers enter a SKU in the Quick Order field and click **GO**, the QuickOrderForm command calls the OrderItemAdd command to add the SKUs to the order. The QuickOrderView command loads the Quick Order, QuickOrder.jsp page.

If customers click on **Enter More Items** from the navigation sidebar, the QuickOrderView command loads the Quick Order page. Customers can then enter a list of SKUs and add them to the order.

The Request for Quote section will only be displayed to customers with a Buyer (buy-side) role. The RFQ Tool requires that an interest list be filled with items. If customers click on **Interest List (for RFQ)**, the Interest List (for RFQ) page InterestItemDisplay.jsp is displayed. If customers click on **Launch RFQ Tool**, and have the Buyer (buy-side) role, the RFQ Tool is loaded and displayed to the customers.

If customers do not have any items in their interest list, a message alert pops up informing customers that the interest list is empty and the Interest List (for RFQ) page displays providing instructions on creating an RFQ request.

Note: Customers must have Internet Explorer 5.5 in order to access the **Launch RFQ Tool** link. In order to submit a RFQ request, customers must be a business user and have the Buyer (buy-side) role assigned to them. If customers do not supply their parent organization at the time of Registration, they will be logged in as a non-business user. They will then need to register again and provide a buyer organization other than the default organization or root organization.

If customers click **Live Chat with Customer Care**, a pop-up window is displayed on the screen, and customers can chat online in real-time with a customer service representative. This link will only be displayed if this feature is enabled in Store Services. The **Live Chat with Customer Assistance** link is contained within the body of a pair of custom tags (the <flow:ifEnabled feature="customerCare"> tag and </flow:ifEnabled> tag), and can be enabled or disabled based on what option is selected in Store Services. As long as the custom tag remains in place, Store Services can be used to automatically switch between a site that has collaboration support and one that does not, without having to modify the JavaServer page. To permanently enable or disable collaboration support in the page, the custom tags and the **Live Chat with Customer Care** link can be removed from the JavaServer page by clicking **Apply Permanently**, in the Store Services GUI. It is not recommended to manually remove or alter the custom tags or the code it surrounds. Instead, use the **Apply Permanently** button in Store Services.

Note:The custom tag that surrounds the **Live Chat with Customer Care** link cannot be copied into the JavaServer pages of other stores. These tags are only intended to work in the store that originally contained them. For more information on collaboration, see the related links below.

ToolTech Order Summary page (Business Edition)

Business

In the Order Summary page (`OrderDisplayPending.jsp`), customers can review their detailed order information, including a description of the items bought, as well as SKU number, quantity, manufacturer, total price, shipping address, shipping method, contract, discount amounts, and shipping costs.

If customer's contracts specify a purchase order (PO) number, they must enter the number in the **Purchase order** field. The input field for the PO number is only displayed if the number is required. If there is more than one order, there will be a Purchase Order input box for each order that requires a PO. The number entered in each box must be unique or must be the blanket PO. Customers must then complete the order by providing payment information and clicking **Order Now**. Before clicking **Order Now**, customers can choose to make the order a scheduled order.

Note: If credit line was chosen as a payment method in the Your Order, `ShoppingCart.jsp`, page, the credit line account number is retrieved from database. Customers cannot change the payment method from credit line to another option on this page.

The sample store Order Summary page uses the following JSP files:

- `OrderDisplayPending.jsp` (Order Summary page)
- `StandardCreditCard.jsp` (If customers choose credit card as a payment method in the Your Order, `ShoppingCart.jsp`, page.)
- `SimpleCreditLine.jsp` (If customers choose credit line as a payment method in the Your Order, `ShoppingCart.jsp`, page.)
- `MultiOrderProcess.jsp` (Used to process two orders at once if customers create two orders by splitting the order into two.)
- `AllocationCheck.jsp` (Used to check if there is sufficient inventory available for the order items.)

Commands

`OrderDisplayPending.jsp` uses the following commands:

- `OrderProcess`
- `OrderSchedule`
- `MultiOrderProcess`

Beans

`OrderDisplayPending.jsp` uses the following beans:

- `OrderDataBean`
- `OrderItemDataBean`
- `AddressDataBean`
- `ErrorDataBean`
- `ShippingModeDescriptionDataBean`
- `UsablePaymentTCListDataBean`
- `ContractDataBean`
- `TradingAgreementAccessBean`

Implementation details

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see [Sample store pages: common implementation techniques](#).

When customers initiate the order process by clicking **Next** in the Your Order, ShoppingCart.jsp, page, they move through a series of checkout pages, one of which is the Order Summary page (OrderDisplayPending.jsp).

```
OrderAccessBean orderABFinder = new OrderAccessBean(); //Retrieve all
Orders for this User Enumeration ordersABList = orderABFinder.findByStatusMemberAndStore("P",
userId, new Integer(storeId)); OrderAccessBean
next_order; ControllerCmdExecUnit
execUnit; TypedProperty iProperties =
commandContext.getRequestProperties(); String ToOrderId =
""; int counter = 0;
```

Payment information is dynamically included based on the payment method selected by customers on the Your Order page. This is shown in the following code:

```
// use the paymentMethod parameter to find the selected PaymentTCInfo
<jsp:useBean id="usablePaymentTCs"
class="com.ibm.commerce.payment.beans.UsablePaymentTCListDataBean" scope="page" >
</jsp:useBean>
<%
usablePaymentTCs.setOrderId(new Long(orderRn));
com.ibm.commerce.beans.DataBeanManager.activate(usablePaymentTCs, request);
PaymentTCInfo[] paymentTCInfo = usablePaymentTCs.getPaymentTCInfo();
PaymentTCInfo selectedPaymentTCInfo = null;
for (int i = 0; i < paymentTCInfo.length; i++) {
if (paymentTCInfo[i].getLongDescription().equals(paymentMethod)) {
selectedPaymentTCInfo = paymentTCInfo[i];
}
}
...<%
incfile = storeDir + "" + selectedPaymentTCInfo.getAttrPageName()+".jsp";
%>
<jsp:include page="<%=incfile%>" flush="true"/>
```

This page includes an option that allows customers to submit their credit card information. The UsablePaymentTCListDataBean is used to get the available credit card names from the Payment Manager, and the action for the form is set to OrderProcess.

If customers have chosen to split their order, two orders will be shown on the page. When they click **Order Now**, the MultiOrderProcess command is called. MultiOrderProcess is a view command registered in the VIEWREG table, and is associated with MultiOrderProcess.jsp. MultiOrderProcess.jsp executes OrderProcess multiple times depending on the numbers of orders in the order summary page. In ToolTech, the checkout flow only allows up to two orders in the order summary page.

Note: If there are two orders, the option to schedule the orders is not displayed.

If customers choose to make the order a scheduled order, and select the **Scheduled Order** checkbox, the OrderSchedule command is called to schedule the order based on the parameters specified by the customers. Scheduling the order creates a new scheduled order based on the current order and leaves the current order intact. In this case, MultiOrderProcess is not called.

```
function OrderSchedule()
{
var form = document.CardInfo;
var payInfo="";
```

```
var StartDate = date();
payInfo="pay_cardBrand="+form.cardBrand.value+"&pay_cardNumber="+form.cardNumber.value+"&pay_cardExpiryMonth="
return payInfo;
}
```

After the form is submitted, and if the order process is successful, the OrderOKView is called. The OrderOKView command is registered in the VIEWREG table in the database, and is associated with the Confirmation.jsp that displays information to confirm the order.

When customers click on **Previous**, they are directed back to the Shipping.jsp page.

Note: If customers belong to Organization A, when the total order amount is greater than 1000 dollars, the order must be approved before it can be submitted. Only the contract belonging to Organization A, has the approval condition.

Error Handling

If there is an error, the DoPaymentErrorView is called. DoPaymentErrorView is associated in the database with OrderDisplayPending.jsp. As a result, when there is an error, the Order Summary page (OrderDisplayPending.jsp) redisplay with an error message.

If there is an error with the PO number, the ValidatePONumberErrorView or DoPaymentErrorView will be called depending on the exception caught.

ToolTech Order Status page (Business Edition)

Business

Customers can click on the **Order Status** link in the header menu to access the Order Status page, TrackOrderStatus.jsp. The Order Status page displays the following:

- Orders awaiting approval
- Processed orders
- Scheduled orders

From this page customers can view the status and details of their orders, and choose to re-order processed orders and cancel scheduled orders.

Commands

TrackOrderStatus.jsp uses the following commands:

- OrderDetail
- OrderCopy
- ScheduledOrderCancel (Cancels scheduled orders)

Beans

TrackOrderStatus.jsp uses the following beans:

- OrderAccessBean
- SchedulerOrderDataBean
- OrderDataBean
- SchedulerConfigAccessBean (Gets the job reference number for scheduled orders)
- AttributeValueDataBean
- AttributeDataBean

- CatalogEntryAccessBean
- ItemDataBean

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see [Sample store pages: common implementation techniques](#).

When customers click on the **Order Status** link on the header menu, the OrderAccessBean retrieves all the orders and properties of each order including status, total price, order date, and purchase order number.

```
while (ordersList.hasMoreElements()) { next_order = (OrderAccessBean) ordersList.nextElement();
String orderStatusCode = next_order.getStatus();
```

```
//get purchase order number
```

```
String poNumber = orderDB.getPurchaseOrderNumber();
if (poNumber.equals("")) {
```

```
poNumber = tooltechttext.getString("Status_none");
}
```

Orders Awaiting Approval

All orders in the approval state have a status of *W*, shown below:

```
if (orderStatusCode.equals("W"))
```

Processed Orders

If customers decide to re-order a processed order, by clicking on the **Re-order** link, the OrderCopy command copies the order and calls the OrderItemDisplay command which launches the Your Order, shoppingcart.jsp page.

```
OrderCopy?fromOrderId_1=<%=nextOrderId%>&toOrderId=.**.&OrderItemId_1=*&URL=OrderItemDisplay
```

Processed orders can be filtered out to not be displayed depending on their status.

```
if (orderStatusCode.equals("N") || orderStatusCode.equals("M") ||
orderStatusCode.equals("B") || orderStatusCode.equals("C") || orderStatusCode.equals("R")
|| orderStatusCode.equals("S") ){
```

```
//get the date ordered
String orderDateString;
SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
Date orderDate; orderDate = next_order.getPlaceOrderTimeInEJBType(); orderDateString
```

Note:For more information on order status codes, see the [Related References](#) link at the bottom of this page.

Scheduled orders

When customers create scheduled orders, the scheduled orders are retrieved using the code below:

```

CommandContext commandContext = (CommandContext)
request.getAttribute(ECConstants.EC_COMMANDCONTEXT);
Long userId = commandContext.getUserId();
SchedulerOrderByMemberIdListDataBean orderABFinder = new
SchedulerOrderByMemberIdListDataBean();
orderABFinder.setDataBeanKeyMemberId(userId.toString());
com.ibm.commerce.beans.DataBeanManager.activate(orderABFinder, request);
com.ibm.commerce.order.beans.SchedulerOrderDataBean bnSchOrd[] =
orderABFinder.getSchedulerOrderByMemberIdList

```

The interval for each scheduled order is displayed using the following code:

```

SchedulerConfigAccessBean SchConBean = new SchedulerConfigAccessBean();
SchConBean.setInitKey_jobReferenceNumber(bnSchOrd[i].getJobReferenceNumber());
String Interval = ""; String frequency = SchConBean.getInterval()+"";
if(frequency.equals("0")) Interval = tooltechtext.getString("OrderSum_Interval_6"); else
if(frequency.equals("86400")) Interval =
tooltechtext.getString("OrderSum_Interval_1");
if(frequency.equals("604800")) Interval =
tooltechtext.getString("OrderSum_Interval_2");
else if(frequency.equals("1209600")) Interval =
tooltechtext.getString("OrderSum_Interval_3");
else if(frequency.equals("1814400")) Interval =
tooltechtext.getString("OrderSum_Interval_4");
else if(frequency.equals("2592000")) Interval =
tooltechtext.getString("OrderSum_Interval_5");

```

All scheduled orders have an order status of *I*. If customers select **Cancel**, a java script message will be displayed asking customers if they really want to cancel the order. If customers select **Yes**, the order is cancelled and the customer is brought back to the `TrackOrderStatus.jsp` page.

When customers click on the order number link for any order, the resulting Order Details page, `OrderDetails.jsp`, displays.

Note: The Order Details page is similar to the Order Summary page, without the payment information.

ToolTech order status codes (Business Edition)

Business

When customers view the Order Status page, `TrackOrderStatus.jsp`, they can view the details and status of their orders. Each status is displayed in a table on the Order Status page, and is represented by a letter.

The following table defines the order status' that are displayed to the customer:

Order Status	Description
N	The order has been denied approval.
M	The order is approved, pending payment authorization
B	The order is back-ordered.
C	The order is approved, and payment is authorized.
R	The order is in fulfillment.
S	The order has been shipped.

W	The order is pending approval.
---	--------------------------------

ToolTech Your Order page (Business Edition)

Business

The Your Order page, `shoppingcart.jsp`, displays the contents of customers orders and is the first step in the checkout process. The Your Order page displays the SKU, quantity, description, manufacturer, part number, contract, and pricing for each item in the order. Customers can also create a new requisition list from the order, update the quantity, change the contract for each order item, and remove an item from the order. Customers must select a payment method before proceeding with the checkout.

The following pages are included in the ToolTech order process:

- `ShoppingCart.jsp` (Your Order page)
- `StandardCreditcard.jsp` (Includes the parameters for the payment information but is not displayed to customers)
- `SimpleCreditLine.jsp` (Includes the parameters for the payment information but is not displayed to customers)
- `AllocationCheck.jsp`
- `CheckProdAvail.jsp`
- `Shipping.jsp`
- `OrderDisplayPending.jsp`
- `Confirmation.jsp`
- `MultiOrderProcess.jsp`

Commands

`ShoppingCart.jsp` uses the following commands:

- `OrderItemDisplay`
- `OrderItemUpdate`
- `ProductDisplay`
- `OrderItemDelete`
- `RequisitionListCopy`

Beans

`ShoppingCart.jsp` uses the following beans:

- `OrderItemDataBean`
- `OrderListDataBean`
- `UsablePaymentTCListDataBean`
- `FormattedMonetaryAmountDataBean`
- `OrderAccessBean`
- `OrderDataBean`
- `ContractDataBean`
- `AttributeValueDataBean`
- `AttributeDataBean`
- `ItemDataBean`

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see Sample store pages: common implementation techniques.

The Your Order page, `ShoppingCart.jsp`, is displayed to customers when they do the following:

- Click on **Current Order** on the menu bar.
- Add to an order from the Item Display page.
- Add a requisition list to an order from the Requisition List page.
- Re-order from the Order Status page.
- Order from the Quick Order page.

The Your Order Page, `ShoppingCart.jsp` is forwarded from the main checkout page controller, `OrderItemDisplay.jsp`. Before reaching `ShoppingCart.jsp`, `OrderItemDisplay.jsp` checks to see if customers have more than one order, due to customers splitting an order and not completing an order. If there is more than one order, `OrderItemDisplay.jsp` calls `OrderMerge` to make sure that there is always one order in the Your Order page at all times.

The `OrderDataBean` in the `OrderItemDisplay.jsp` page is used to retrieve the pending order information for customers. If there are more than one pending order, it merges these into one order.

`OrderItemDisplay.jsp` includes the empty shopping cart page if there are no pending orders for the customers. The `EmptyShopCart.jsp` page is loaded and a message is displayed to the customer.

```
if (!bHasShopCart) { incfile = storeDir + "Emptyshopcart.jsp"; }
```

If customers have not selected a shipping address the `ShipAddress.jsp` is displayed.

```
if (state.equals("shipaddress") || state.equals("newshipaddress") ) { incfile = storeDir + "shipaddress.jsp"; }
```

If customers have not selected a shipping method, and the `page` parameter equals `shipmethod` it includes `Shipping.jsp`.

```
if (state.equals("shipmethod")) { incfile = storeDir + "Shipping.jsp"; }
```

If customers click on **New Requisition List**, `RequisitionListCopy` calls `RequisitionListUpdateView` which forwards to `RequisitionListUpdate.jsp`. After the new list is created, the Requisition List, `RequisitionListDisplay.jsp`, page is displayed.

If customers choose to update their order by changing the quantity, or selecting a contract, after clicking on **Update**, the `OrderItemUpdate` command updates the order items, and reloads the Your Order page. `OrderItemUpdate` is also called when customers click **Next**.

```
<form name="ShopCartForm" action="OrderItemUpdate" method="GET"> <input type="hidden" name="orderId" value="*"> <input type="hidden" name="URL" value="OrderItemDisplay">
```

```
<input type="hidden" name="orderItemId_<%=counter%>" value="<%=orderItem.getOrderItemId()%>" > <input type="text" name="quantity_<%=counter%>" size="2" value="<%=orderItem.getQuantityInEJBType().intValue()%>">
```

After customers have selected a payment method and updated their orders, they can click on **Next** to continue processing the order.

Error Handling

If the payment drop down is empty, and customers click on **Next**, an error message will be displayed. If the requisition list name field is blank, and customers click on **New Requisition List**, an error message will also be displayed.

ToolTech Quick Order page (Business Edition)

Business

When customers click on **Enter More Items**, on the navigation sidebar, the Quick Order page, `QuickOrder.jsp` is displayed. The Quick Order page allows customers to enter the SKUs and quantities for items to add to their order. If no quantity is entered, the quantity defaults to one.

Commands

`QuickOrder.jsp` uses the following commands:

- `OrderItemAdd`
- `QuickOrderView`

Beans

`QuickOrder.jsp` uses the following beans:

- `ErrorDataBean`
- `OrderDataBean`

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see [Sample store pages: common implementation techniques](#).

When customers enter a SKU in the navigation sidebar and click on **GO**, the `OrderItemAdd` command adds the SKU to the order and redirects to the `OrderItemDisplay` URL. The `OrderItemDisplay` command then loads the Your Order, `ShoppingCart.jsp`, page.

If customers click on **Enter More Items** from the navigation sidebar, the `QuickOrderView` command loads the Quick Order page.

When customers click **Submit**, if the SKUs entered are valid, they will be added to the shopping cart and the Your Order, `shoppingcart.jsp`, page will be displayed. If the SKUs entered are not valid, `QuickOrder.jsp` will be displayed with a list of bad SKUs at the top of the Quick Order page.

ToolTech registration pages (Business Edition)

Business

When new customers click on **Register** on the Logon page, the `UserRegistrationNew.jsp` page is displayed. This page allows customers to fill in fields of personal information including selecting a password, language, and currency. After customers have filled in the information they are re-directed to the User Account, `UserAccount.jsp`, page to enter the store.

The sample store registration procedure uses the following files:

- `UserRegistrationNew.jsp` (Registration page)
- `UserRegistrationForm.jsp` (Includes the parameters for the `UserRegistrationAdd` command, but is not displayed to the customer.)
- `UserRegistrationUpdate.jsp` (Update Registration page)

- UserRegistrationAddPendingApproval.jsp
- UserAccount.jsp (User Account page)

Commands

UserRegistrationNew.jsp uses the following commands:

- UserRegistrationAdd

UserRegistrationUpdate.jsp uses the following commands:

- UserRegistrationUpdate

UserRegistrationAddPendingApproval.jsp uses the following commands:

- LogonForm

UserAccount.jsp uses the following commands:

- UserRegistrationForm
- AddressBookForm

Beans

UserRegistrationNew.jsp uses the following beans:

- ErrorDataBean
- CurrencyDescriptionAccessBean
- SupportedLanguageAccessBean
- LanguageDescriptionAccessBean

UserRegistrationUpdate.jsp uses the following beans:

- UserRegistrationDataBean
- OrgEntityDataBean
- ErrorDataBean
- CurrencyDescriptionAccessBean
- SupportedLanguageAccessBean
- ResourceBundleDataBean

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see *Sample store pages: common implementation techniques*.

After customers click **Register** on the Logon page, the UserRegistrationForm command is called. UserRegistrationForm is registered with UserRegistrationForm.jsp in the database. If UserRegistrationForm is called with a parameter called *new*, then UserRegistrationForm.jsp will include the UserRegistrationNew.jsp. UserRegistrationNew.jsp is used for registering a new user. If the *new* parameter is missing, then UserRegistrationForm includes the UserRegistrationUpdate.jsp, which updates customer's existing information. UserRegistrationUpdate.jsp is called by UserRegistrationForm from the Account page.

When customers select a currency, the CurrencyDescriptionAccessBean pre-selects the appropriate value in the drop-down list.

When customers select a language, the `SupportedLanguageAccessBean` pre-selects the appropriate value in the drop-down list and the `LanguageDescriptionAccessBean` gets the display name of the language in the language currently selected by the shopper.

When customers enter an organization, they must enter the name of the organization if the organization is created directly under Root Organization, such as *Buyer Organization A*. If the organization to be entered is an organization unit and does not exist directly under Root Organization, customers must enter the organization unit names followed by the organization name, and separated by a slash (/).

For example, if the distinguish name is

`ou=RD, ou=Sales, o=Sears, o=Root Org`

customers will have to enter the following:

`RD/Sales/Sears`

ToolTech converts the above entry to the distinguish name format by adding `o=`, or `ou=` before calling the `UserRegistrationAdd` command. Only the entity directly under Root Organization is `O`. Other entities are `OU`.

If an invalid organization is entered, an error message is displayed on the screen with instructions on how customers can enter their organization.

When customers have finished registration, and user registration approval has been set up for the organization, they are put into a pending approval state and can only access the Account section of the store, where they can change personal information or edit the address book. Customers with the role Buyer Administrator must approve the user registration request through the organization Administration Console. A link to the organization's Administration Console is displayed on the Home page of the store, but is only visible if customers have the role of Buyer Approver or Buyer Administrator. Refer to the related links below for further information.

If customers click **Live Chat with Customer Assistance**, a pop-up window is displayed on the screen, and customers can chat online in real-time with a customer service representative. This link will only be displayed if this feature is enabled in Store Services. The **Live Chat with Customer Assistance** link is contained within the body of a custom tag (the *ifEnabled* tag), and can be enabled or disabled based on what option is selected in Store Services. As long as the custom tag remains in place, Store Services can be used to automatically switch between a site that has collaboration support and one that does not, without having to modify the `JavaServer` page. To permanently enable or disable collaboration support in the page, the custom tags and the **Live Chat with Customer Assistance** link can be removed from the `JavaServer` page by clicking **Apply Permanently**, in the Store Services GUI. It is not recommended to manually remove or alter the custom tag or link it surrounds. Instead, use the **Apply Permanently** button in Store Services.

Note:The custom tag that surrounds the **Live Chat with Customer Assistance** link cannot be copied into the `JavaServer` pages of other stores. These tags are only intended to work in the store that originally contained them. For more information on collaboration, see the related links below.

Error handling

If customers do not complete the fields during registration, an error message is displayed and what was entered is re-displayed to the customer. If there is no error, all fields are initialized to an empty state.

The `UserRegistration_[locale].properties` file and `UserRegistrationB2B_[locale].properties` file determine what fields in the address each locale specifies as mandatory.

UserRegistration_[locale].properties is used for customers in organization C, and UserRegistrationB2B_[locale].properties is used for customers in organization B.

ToolTech Add to Existing Requisition List page (Business Edition)

Business

The Add to Existing Requisition List page AddToExistReqList.jsp, is displayed when customers click **Add to Requisition List**, on the Item Display page CatalogItemDisplay.jsp, and choose the **Existing** radio button.

Commands

AddToExistReqList.jsp uses the following commands:

- RequisitionListItemUpdate

Beans

AddToExistReqList.jsp uses the following beans:

- OrderAccessBean
- RequisitionDataBean
- PRequisitionByMemberIdAndStoreListDataBean
- SRequisitionByMemberIdAndStoreIdListDataBean

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see Sample store pages: common implementation techniques.

When customers select the **Existing** radio button, select an existing requisition list, and click on **Add to Requisition List** on the Item Display page, the Add to Existing Requisition List page is displayed. If customers have not created requisition lists before, a message is displayed asking the customers to go back to the Item Display page CatalogItemDisplay.jsp, and add the item to a new requisition list.

Each requisition list owned by logged on customers is displayed with the requisition list name, the date it was last updated, and whether it is a shared or private list. The RequisitionDataBean and PRequisitionByMemberIdAndStoreListDataBean, retrieve the private lists from the database.

```
//My private lists
```

```
RequisitionDataBean[] requisitionListsP;
```

```
PRequisitionByMemberIdAndStoreIdListDataBean pReqListDB=new  
PRequisitionByMemberIdAndStoreIdListDataBean();  
pReqListDB.setDataBeanKeyMemberId(userRefNum.toString());  
pReqListDB.setDataBeanKeyStoreId(storeId);  
com.ibm.commerce.beans.DataBeanManager.activate(pReqListDB, request); requisitionListsP =  
pReqListDB.getPRequisitionByMemberIdAndStoreIdList();
```

```
for (int i=0; i<requisitionListsP.length; i++) {  
com.ibm.commerce.beans.DataBeanManager.activate(requisitionListsP[i], request); }
```

The RequisitionDataBean and SRequisitionByMemberIdAndStoreListDataBean, retrieve the shared lists in the customer's organization from the database. Only the lists owned by the logged on customers will be displayed.

```

//All shared lists in my organization RequisitionDataBean[] requisitionListsSAll;

SRequisitionByMemberIdAndStoreIdListDataBean sReqListDB=new
SRequisitionByMemberIdAndStoreIdListDataBean();
sReqListDB.setDataBeanKeyMemberId(userRefNum.toString());
sReqListDB.setDataBeanKeyStoreId(storeId);
com.ibm.commerce.beans.DataBeanManager.activate(sReqListDB,
request);
= sReqListDB.getSRequisitionByMemberIdAndStoreIdList();
requisitionListsSAll

for (int i=0; i<requisitionListsSAll.length; i++)

{com.ibm.commerce.beans.DataBeanManager.activate(requisitionListsSAll[i], request); }

if (form.reqListId.length==null) {
form.requisitionListId.value=form.reqListId.value;
} else {
for (var i=0;i<form.reqListId.length;i++) {
if (form.reqListId[i].checked) {
form.requisitionListId.value=form.reqListId[i].value;
break;
}
}
}

```

After customers have chosen a requisition list, the `RequisitionListItemUpdate` command updates the requisition list with the new order item. The `RequisitionListUpdateView` displays the Edit Requisition List page with the new order items. After the `RequisitionListItemUpdate` command is completed, the view is redirected to `RequisitionListUpdateView`. `RequisitionListUpdateView` is associated with `RequisitionListUpdate.jsp` in the database.

ToolTech New Requisition List page (Business Edition)

Business

When customers click on **Create New Requisition List**, on the Requisition List page, the New Requisition List page, `RequisitionListCreate.jsp`, is displayed. Customers must choose a name for the new list, whether it is to be private or shared, and can enter in the SKU's and quantities of the items to be in the new list.

Note: Customers can only enter a maximum of 30 items per requisition list on this page. If customers want to add more items, they must use the Edit Requisition List page. If customers enter the valid SKU and leave the quantity blank, the quantity will be defaulted to one. If the SKU field is left empty, the quantity field will be ignored.

Commands

`RequisitionListCreate.jsp` uses the following commands:

- `RequisitionListItemUpdate`
- `RequisitionListUpdateView`
- `RequisitionListCreateView`

Beans

`RequisitionListCreate.jsp` uses the following beans:

- `ErrorDataBean`
- `ProductDataBean`

Implementation

When customers click on **Create New Requisition List**, on the Requisition List page, the `RequisitionListCreateView` command is called and the New Requisition List page, `RequisitionListCreate.jsp`, is displayed. `RequisitionListCreateView` is registered with `RequisitionListCreate.jsp` in the database. All fields are initialized to empty as seen in the code below:

```
//Form is loading under regular condition. Initialize all fields to empty. strName = "";
strStatus = "Y";

for (int i=0; i<31; i++) { strPartNumber[i] = ""; strQuantity[i] = ""; } strPartNumber[1] =
partNumber; strQuantity[1] = quantity;
```

After customers have entered in the new information and clicked on **Save**, the `RequisitionListUpdate` command creates a new requisition list and populates it with the new items. `RequisitionListUpdateView` then loads the Edit Requisition List page. Customers can then edit the newly created requisition list by changing the type, adding a new item, or changing the quantity. `RequisitionListUpdate.jsp` is used to edit the requisition list.

Note: The default requisition list is private.

Error Handling

If customers enter inaccurate information such as an incorrect SKU or quantity, the `ErrorDataBean` displays an error message on the page as seen below:

```
if (strMessageKey != null && strMessageKey.length() > 0) {
// We have an error

    strMessageParams = bnError.getMessageParam();
    if (strMessageKey.equals(ECMessageKey._ERR_BAD_MISSING_CMD_PARAMETER)) {
strErrorMessage = tooltechttext.getString("Newreq_Error_Missing_Parameter");
    } else if (strMessageKey.equals(ECMessageKey._ERR_GETTING_SKU)) {
TypedProperty nvps = bnError.getExceptionData();
    if (nvps != null) {
String catEntryId = nvps.getString("catEntryId");
ProductDataBean prodDB=new ProductDataBean();
prodDB.setProductID(catEntryId);
com.ibm.commerce.beans.DataBeanManager.activate(prodDB, request);
strErrorMessage = tooltechttext.getString("Newreq_Error_ProdSKU")+prodDB.getPartNumber();
    }

    } else if (strMessageKey.equals(ECMessageKey._ERR_PROD_NOT_EXISTING)) {
strFieldName = (String)strMessageParams[0];
strErrorMessage = tooltechttext.getString("Newreq_Error_SKU") + strFieldName;
    } else if (strMessageKey.equals(ECMessageKey._ERR_INVALID_INPUT)) {
strErrorMessage = tooltechttext.getString("Newreq_Error_Quantity")+jhelper.getParameter("quantity");
    }
//Retrieve form data entered before
strName = jhelper.getParameter("orderDesc");
strStatus = jhelper.getParameter("status");
for (int i=1; i<31; i++) {
strPartNumber[i] = jhelper.getParameter("partNumber_"+i);
strQuantity[i] = jhelper.getParameter("quantity_"+i);
if (strPartNumber[i]==null) strPartNumber[i] = "";
if (strQuantity[i]==null) strQuantity[i] = "";
    }
}
```

ToolTech View Requisition List page (Business Edition)

Business

Customers can select a requisition list on the Requisition List page, `RequisitionListDisplay.jsp`, and see the details of the selected list on the `RequisitionListDetailDisplay.jsp` page. Details include the SKU, quantity, description, manufacturer, and part number of the order items. From this page, customers can choose to add the requisition list to their order, or to display all of the requisition lists that they own plus the shared requisition lists owned by other people in the same organization.

Note: If customers choose a shared list owned by someone else from the Requisition List page, they can only view the details of that selected requisition list. Customers cannot edit other customer's requisition lists.

Commands

`RequisitionListDetailDisplay.jsp` uses the following commands:

- `ProductDisplay`
- `OrderItemAdd`
- `OrderItemDisplay`
- `RequisitionListView`

Beans

`RequisitionListDetailDisplay.jsp` uses the following beans:

- `ItemDataBean`

Implementation

After buyers select a shared requisition list owned by other buyers on the Requisition List page, `RequisitionListDisplay.jsp`, the `RequisitionListDetailView` command calls the `RequisitionListDetailDisplay.jsp` page.

If customers decide to add the requisition list to their order, and click on **Add List to Order**, the `OrderItemAdd` command adds the order to the list and calls the `Your Order, OrderItemDisplay.jsp`, page. The `OrderItemAdd` command is associated with `OrderItemForm` in the Requisition List page.

```
<FORM NAME="OrderItemAddForm" ACTION="OrderItemAdd" Method="get"> <input type="hidden" name="storeId" value="<%=storeId%>"> <input type="hidden" name="orderId" value="."> <input type="hidden" name="langId" value="<%=languageId%>"> <input type="hidden" name="catalogId" value="<%=catalogId%>"> <input type="hidden" name="URL" value="OrderItemDisplay?orderId=">
```

If customers select to display all requisition lists by clicking on, **View all Requisition Lists**, the `RequisitionListView` command loads the Requisition List page, `RequisitionListDisplay.jsp`.

ToolTech Requisition List page (Business Edition)

Business

When customers click on the **Requisition List** link on the header, the Requisition List page, `RequisitionListDisplay.jsp`, is displayed. Customers can view the list of requisition lists, including all requisition lists owned by the them, and all shared lists owned by someone who is in the same organization. Customers have the option to remove or edit a list owned by them, create a new one, or view another customer's shared list.

Commands

RequisitionListDisplay.jsp uses the following commands:

- RequisitionListView
- RequisitionListUpdateView
- RequisitionListDelete
- RequisitionListDetailView
- RequisitionListCreateView

Beans

RequisitionListDisplay.jsp uses the following beans:

- RequisitionDataBean
- PRequisitionByMemberIdAndStoreListDataBean
- UserInfoDataBean
- SRequisitionByMemberIdAndStoreListDataBean

Implementation

After customers click on the **Requisition List** link on the header, the Requisition List page, RequisitionListDisplay.jsp is displayed. The RequisitionDataBean and PRequisitionByMemberIdAndStoreListDataBean, retrieve the private lists owned by the logon user from the database.

```
//My private lists RequisitionDataBean[] requisitionListsP;  
PRequisitionByMemberIdAndStoreIdListDataBean pReqListDB=new  
PRequisitionByMemberIdAndStoreIdListDataBean();  
pReqListDB.setDataBeanKeyMemberId(userRefNum.toString());  
pReqListDB.setDataBeanKeyStoreId(storeId);  
com.ibm.commerce.beans.DataBeanManager.activate(pReqListDB, request); requisitionListsP =  
pReqListDB.getPRequisitionByMemberIdAndStoreIdList(); for (int i=0; i<requisitionListsP.length;  
i++) { com.ibm.commerce.beans.DataBeanManager.activate(requisitionListsP[i], request); }
```

The RequisitionDataBean and SRequisitionByMemberIdAndStoreListDataBean, retrieve the shared lists in the organization from the database.

```
//All shared lists in my organization RequisitionDataBean[] requisitionListsSAll;  
SRequisitionByMemberIdAndStoreIdListDataBean sReqListDB=new  
SRequisitionByMemberIdAndStoreIdListDataBean();  
sReqListDB.setDataBeanKeyMemberId(userRefNum.toString());  
sReqListDB.setDataBeanKeyStoreId(storeId);  
com.ibm.commerce.beans.DataBeanManager.activate(sReqListDB, request); requisitionListsSAll =  
sReqListDB.getSRequisitionByMemberIdAndStoreIdList(); for (int i=0;  
i<requisitionListsSAll.length; i++) {  
com.ibm.commerce.beans.DataBeanManager.activate(requisitionListsSAll[i], request); } %>
```

```
//shared lists of others in my organization Vector requisitionListsSOther=new Vector(); %>
```

The RequisitionListView command loads the Requisition List page. RequisitionListView is associated with RequisitionListDisplay.jsp in the database. In the JSP, RequisitionListsTypeForm includes the *requisitionListsType* parameter for RequisitionListView. The parameter can be *All* or *Mine*.

When customers click on a requisition list owned by someone else, the RequisitionDetailListView command is called and the View Requisition List page, RequisitionListDetailDisplay.jsp, is displayed to

the customer. `RequisitionListDetailDisplay.jsp` shows the items in a shared requisition list that belongs to someone else, who is in the same organization as the customer.

When customers click on their own requisition list, the `RequisitionListUpdateView` command is called and the Edit Requisition List page, `RequisitionListUpdate.jsp`, is displayed to the customer.

If customers click on **Create New Requisition List**, the `RequisitionListCreateView` command is called and the New Requisition List, `RequisitionListCreate.jsp`, is displayed.

ToolTech Edit Requisition List page (Business Edition)

Business

From the Edit Requisition List page `RequisitionListUpdate.jsp`, customers can edit their requisition list. Customers can do the following:

- Change the requisition list type
- Add new items
- Change the quantities of items
- View item details
- Remove items
- Add the list to their order

Commands

`RequisitionListUpdate.jsp` uses the following commands:

- `RequisitionListUpdate`
- `RequisitionListItemUpdate`
- `OrderItemAdd`
- `OrderItemDisplay`
- `ProductDisplay`
- `RequisitionListUpdateView`

Beans

`RequisitionListUpdate.jsp` uses the following beans:

- `OrderDataBean`
- `ItemDataBean`
- `ErrorDataBean`
- `CatalogDataBean`
- `ProductDataBean`

Implementation

When customers change the type of requisition list and click **Change Type**, the `RequisitionListUpdate` command is called and adds the new item to the list. The `RequisitionListUpdate` command is associated with `RequisitionListTypeForm`.

When customers add new items to the requisition list and click **Add**, the `RequisitionListUpdate` command is called and adds the new items to the list. The `RequisitionListUpdate` command is associated with `AddRequisitionListItemForm`.

When customers decide to change the quantities of the items in the requisition list page, and click **Update Quantity**, the RequisitionListItemUpdate command is called and adds the new item to the list. The RequisitionListItemUpdate command is associated with UpdateRequisitionListItemForm.

When customers click **Add List to Order**, the OrderItemAdd command retrieves the new information and calls the OrderItemDisplay command which displays the Your Order page. The OrderItemAdd command is associated with OrderItemAddForm.

```
<FORM NAME="OrderItemAddForm" ACTION="CatalogItemAdd" method="post">
<input type="hidden" name="storeId" value="<%=storeId%>">
<input type="hidden" name="orderId" value=".">
<input type="hidden" name="langId" value="<%=languageId%>">
<input type="hidden" name="catalogId" value="<%=catalogId%>">
<input type="hidden" name="URL" value="OrderItemDisplay?orderId=*&orderId_*=&quantity_*=">
  <input type="hidden" name="allocate" value="*n">
  <input type="hidden" name="reverse" value="*n">
  <input type="hidden" name="backorder" value="*n">
<%
for (int k=0; k<orderItemAB.length; k++) {
Double quantity = orderItemAB[k].getQuantityInEJBType();
%>
<input type=hidden name="catEntryId_<%= k %>" value="<%= orderItemAB[k].getCatalogEntryId() %>" >
<input type=hidden name="quantity_<%= k %>" value="<%= quantity.intValue() == quantity.doubleValue() ? Integer.toString(
<%
}
%>
</FORM>
```

Error Handling

The Edit Requisition List page handles three types of errors. If customers add a new item with an incorrect SKU number such as 1245, the following error message is displayed:

You have entered an invalid SKU number:1245

If customers add a new item with a valid SKU but invalid quantity such as -1, the following error message is displayed:

You have entered an invalid quantity:-1

If customers update the quantity of an item in the list, such as changing the quantity from 2 to -1, the following message is displayed:

You have entered an invalid quantity:-1

The quantity for that item will remain unchanged.

When customers enter a product SKU number such as 2345, in the **SKU** field, and the item SKU number cannot be resolved from the product SKU, for example, there is no product/ item relationship, the following message is displayed:

Cannot resolve the item SKU number from the product SKU number:2345

Sample store pages: common implementation techniques

Most sample store pages use the following implementation techniques. This page uses the InFashion store as an example but can be applied to all sample stores. For more information on techniques specific to an individual page, see the reference file for that page.

Multicultural content

The sample store pages are also used to display multicultural content. That is, the same set of pages can be used for several different locales. Most of the code to enable multicultural display is in `getResource.jsp`. `getResource.jsp` does the following things:

- Retrieves the current locale from the command context and stores it as the `locale` variable.
- Retrieves the store directory and stores it as the `storeDir` variable.
- Retrieves the store's name and stores it as the `storeName` variable.
- Loads the language-specific properties file using the `ResourceBundle` API and stores it as the `infashiontext` variable.

In order for a JavaServer Page file to access the above variables, `getResource.jsp` must be included in the JavaServer Page file, using the compile-time include action:

```
<%@ include file="include/getResource.jsp"%>
```

Since `getResource.jsp` is included in almost all of the samples store JavaServer Page files, it may be executed several times in a single request. To avoid duplication of effort, much of the information retrieved in this page is stored in the request context. For example:

```
String storeDir = (String) request.getAttribute("storeDir"); String includeDir = (String)
request.getAttribute("includeDir"); String fileDir = (String) request.getAttribute("fileDir");
String bundleDir = (String) request.getAttribute("bundleDir");
```

```
String storeName = "";
```

```
if (storeDir == null) {
```

```
storeDir = sdb.getJspPath(); fileDir = sdb.getFilePath(); includeDir = storeDir + "include" +
"/"; bundleDir = sdb.getDirectory(); storeName =
sdb.getDescription(cmdcontext.getLanguageId()).getDisplayName();
request.setAttribute("storeName", storeName); request.setAttribute("storeDir", storeDir);
request.setAttribute("fileDir", fileDir); request.setAttribute("includeDir", includeDir);
request.setAttribute("bundleDir", bundleDir); }
```

Language-specific messages

Language-specific messages like "Thank you for ordering!" are stored in resource bundle properties files. These files are located in the following directory:

▶ NT

```
drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear\wcstores.war\WEB-INF\classes\storeDir
```

▶ 2000

```
drive:\Program
```

```
Files\WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear\wcstores.war\WEB-
```

INF/classes/storedir

▶ AIX

/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/WEB-INF/classes/storedir

▶ Solaris

/opt/WebSphere/Appserver/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/WEB-INF/classes/storedir
/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/WEB-INF/classes/storedir

▶ 400

/QIBM/UserData/WebASAdv4/WAS_instance_name/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war

For example, if your store directory is “storedir”, then the English property file will be:

▶ NT

drive:\drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear\wcstores.war\WEB-INF\classes\infashiontext_en_US.properties.

▶ 2000

drive:\Program
Files\WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear\wcstores.war\WEB-INF\classes\storedir

▶ AIX

/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/WEB-INF/classes/storedir/infashiontext_en_US.properties.

▶ Solaris

/QIBM/UserData/WebASAdv4/WAS_instance_name/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war

Contents of this file are loaded using the Java `java.util.ResourceBundle` API from the JSP file `getResource.jsp`. The bundle is stored as the variable `infashiontext`. A language-specific message is displayed as follows:

```
<title><%=infashiontext.getString("REGISTER_TITLE")%></title>
```

Setting content encoding

Most modern browsers understand HTML data encoded in UTF-8 encoding. However, some older browsers may only understand native encoding. For example, an older Japanese browser may only understand HTML data encoded in “Shift_JIS” encoding. To solve this problem, the sample store specifies language-specific encoding in the resource bundle properties files as the property name `ENCODESTATEMENT`. For example, the English property file `infashiontext_en_US.properties` contains the following entry:

```
ENCODESTATEMENT = text/html; charset=ISO_8859-1
```

For individual JSP files, the encoding is set using the JSP request object as follows:

```
<% response.setContentType(infashiontext.getString("ENCODESTATEMENT")); %>
```

Setting of encoding type needs to be done as early in the JSP file as possible, as the HTTP header is sent out before any HTML content. As a result, if you set the content type (which is sent as HTTP header) after any HTML content, it may not have the desired effect. The browser may not be able to display the data properly.

Including the header, footer, and left navigation frame

Almost all sample store pages display the header (`header.jsp`), footer (`footer.jsp`), and left navigation frame (`sidebar.jsp`) pages. They are included in the other JSP files using the following execution-time include directive:

```
<% String incfile;  
  
incfile = includeDir + "header.jsp"; %> <jsp:include page="<%=incfile%>" flush="true"/>
```

If you know the exact location of the JSP files, you can simplify the include process by using the following: `<jsp:include page="/storedir/include/header.jsp"/>`

where `header.jsp` is located in the `storedir` directory under the Web application document root.

ToolTech Collaborative Workspaces page (Business Edition)

Business

After registered customers click **Collaborative Workspaces** on the header, the Collaborative Workspaces page `StoreCollabListDisplay.jsp`, is displayed.

Note: In order to view this page, the collaborative workspaces feature must be enabled, and customers must be registered.

Commands

`StoreCollabListDisplay.jsp` uses the following commands:

- `StoreCollabMembersDisplay`

Beans

`StoreCollabListDisplay.jsp` uses the following beans:

- `UserRegistrationDataBean`
- `CollabManagerAccessBean`
- `UserRegistryAccessBean`
- `CollabSpaceBean`

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see [Sample store pages: common implementation techniques](#).

When customers click **Collaborative Workspaces** on the ToolTech sample store header, the `UserRegistrationBean` checks to see if customers are registered. If customers are registered, the Collaborative Workspaces page is displayed. Customers can then view the list of collaborative workspaces that they have been invited to. The name of each workspace, a description, and a link to the members of each workspace is displayed. If customers are not registered they are redirected to the Logon page.

When customers click on the name of a workspace they are redirected to the collaborative workspace on the QuickPlace server. When customers click on the **View Members** link, they are redirected to the Members page, `StoreCollabMembersDisplay.jsp`, for that workspace.

Note: If customers have not been invited to any workspaces, the Collaborative Workspaces page will display a message informing customers that they have not been invited to any workspaces, and must contact their Account Representative for further details.

ToolTech collaborative workspaces Members page (Business Edition)

Business

After customers click **View Members** on the Collaborative Workspaces page, they are redirected to the Members page, `StoreCollabMembersDisplay.jsp`.

Beans

`StoreCollabMembersDisplay.jsp` uses the following beans:

- `CollabSpaceBean`
- `CollabManagerAccessBean`
- `UserAccessBean`
- `AddressAccessBean`

Implementation

Note: For information on implementation techniques common to all sample store pages, including multicultural information, see *Sample store pages: common implementation techniques*.

When customers click **View Members** on the Collaborative Workspaces page, the `StoreCollabMembersDisplay` command loads the Members page, `StoreCollabMembersDisplay.jsp`. On this page customers can view the name, phone number, e-mail address, and role of each member in the collaborative workspace.

Chapter 6. ToolTech use cases

ToolTech sample store use cases (Business Edition)

Business

The ToolTech sample store comes with use cases that follow customers through registration, logon, shopping, and order completion. These use cases can help you to understand the flow of the ToolTech sample store, and can be used as a guide for creating use cases for your own store.

The following use cases are provided:

- Logon by a registered customer use case
- Register a new customer use case
- Approve a buyer use case
- Update personal information use case
- Add a new address use case
- Edit address book use case
- Create new requisition list use case
- Add an item to a requisition list use case
- Remove items from a requisition list use case
- Add items to an order use case
- Create an order using a SKU use case
- Create an order from a requisition list use case
-

Logon by a registered customer use case (Business Edition)

Business

This use case will describe what the functional flow of events are when a registered customer logs onto a seller's web site.

Preconditions

The customer has previously registered with the store, and accesses the store site using a supported browser.

Basic flow

1. The system displays the Logon page.
2. The customer selects a shopping language and enters their user ID and password in the **User ID** and **Password** fields.
3. The customer clicks **Submit** and the information is submitted.
4. The customer is transferred to the Home page of the store after logging on is finished.

Alternate flow

The customer forgets their password and would like to reset their password.

1. The customer selects the **Forgot Your Password?** link.
2. The system displays the Forgot Password page.
3. The customer enters 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 customer's e-mail address.
6. A confirmation message is displayed.
7. The customer can transfer to the Logon page after they have received their password.

Exception flow

The customer's 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.

Register a new customer use case (Business Edition)

Business

This use case will describe the functional flow of events when a customer registers in the ToolTech store.

Precondition

The customer accesses the store site using a supported browser.

Basic flow

1. The system displays the Logon page.
2. The customer clicks **Register** and is transferred to the Registration page.
3. The customer completes the User ID, Password, Verify password, Employee ID, Last Name, Buyer Organization, Address, City, State/Province, Country/Region, ZIP/Postal code, E-mail address, Preferred currency, and Preferred language fields, as well as any other optional fields desired.
4. The customer clicks **Submit** and submits the information.

Exception flow

1. The customer leaves the **Buyer Organization** field blank
2. The system registers the customer as a business user.
3. The system displays a message indicating that the customer will be treated as a default user and not a business user.

Postcondition

The customer is waiting for approval to shop in the store from the Buyer Approver.

Approve a buyer use case (Business Edition)

Business

This use case will describe what the functional flow of events are when a Buyer Approver gives approval to a customer to shop in the store.

Pre-conditions

The approval feature has been enabled, and the customer has registered in the store and is waiting to begin shopping. The Buyer Approver has logged into the store.

Basic flow

1. The system displays the Home page.

2. The Buyer Approver clicks **Go to Approval Tool** and is transferred to the Organization Administration Console.
3. The Buyer Approver clicks **Approval->Approval Requests**, and a list of buyers which are pending approval is displayed.
4. The Buyer Approver verifies the correctness of the customer's information and approves or rejects the request for approval.

Post-condition

An e-mail notification is sent to the customer.

Update personal information use case (Business Edition)

Business

This use case will describe what the functional flow of events are when a customer updates their personal information.

Pre-condition

Customer is a registered buyer and is logged on in the store.

Basic Flow

1. The system displays the Account page.
2. The customer clicks **Change Personal Information** and is transferred to the Update Registration page.
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.

Post-condition

The system updates the edited fields.

Add new address use case (Business Edition)

Business

Customers can add a new address to their address book.

Actor

Customer

Main flow

The customer clicks **Add a new address**. The system displays a page with the following fields:

- Nickname (NewFashion only)
- First name
- Last name
- Street address (composed of two text boxes)
- City

- State or Province
- ZIP code or Postal Code
- Country/Region
- Phone number

The customer types the information in the fields and clicks **Submit**. The system adds a new address to the address book (E1).

Alternative flow

None

Exception flows

E1: Missing Mandatory Field

If any of the following fields are missing, system issues an error message.

- First name
- Last name
- Street address
- City
- State or Province
- ZIP code or Postal Code
- Country/Region

If the nickname entered already exists in the customer's address book, the system will report an error message.

The use case resumes from the beginning.

Edit an address use case (Business Edition)

Business

Customers can edit addresses in their address book.

Actor

Customer

Main flow

The customer selects an address from address book to edit. The system retrieves and displays the selected address details:

- Nickname (NewFashion only)
- First name
- Last name
- Street address
- City
- State or Province
- ZIP code or Postal Code
- Country or Region
- Phone number

In NewFashion, the nickname entered by the customer is displayed at the top of the page.

The customer makes the desired changes and clicks **Submit**. The system updates the address (E1).

Alternate flow

None

Exception flows

E1: The system notifies the customer of missing mandatory information, and requests the missing information which can include the following:

- First name
- Last name
- Street address
- City
- State or Province
- ZIP code or Postal Code
- Country or Region

If the nickname already exists in the address book, an error message will be displayed.

Customer enters missing mandatory information.

Create a new requisition list use case (Business Edition)

Business

This use case describes the functional flow of events when a customer creates a new requisition list.

Pre-condition

The customer is a registered buyer and is logged into the store.

Basic flow

1. The customer selects an item to purchase from the catalog and is transferred to the Item Display page.
2. Under the Requisition list heading, the customer selects the **new** radio button and clicks **Add to Requisition List**.
3. The customer is transferred to the New Requisition List page.
4. The system displays a page with an entry box to fill in a name for the requisition list and SKU# and Quantity fields.
5. The customer adds the required information to the fields, clicks **Save**, and the information is submitted.

Post-condition

A new requisition list is created.

Alternate flow

1. Customer selects item to purchase from catalog and is transferred to the Item Display page.
2. The customer clicks **Add to Order** and is transferred to the Your Order page.
3. The customer enters a name for the requisition list in the name field then clicks on **New Requisition List** and is transferred to the Edit Requisition List page.

Post-condition

The system creates a new requisition list and adds all the items with current quantities from the current pending order to this requisition list.

Add an item to a requisition list use case (Business Edition)

Business

This use case describes the functional flow of events when a customer adds items to an existing requisition list.

Pre-condition

The customer has logged into the store and has created at least one requisition list.

Basic flow

1. From any page, the customer clicks **Requisition List** on the header and is transferred to the Requisition List page.
2. The system displays a list of the customer's requisition lists.
3. The customer selects a requisition list and is transferred to the Edit Requisition List page.
4. The system displays the items in the requisition list and fields to enter the SKU numbers and quantities of new items.
5. The customer fills in the fields with the appropriate information.
6. The customer clicks **Add** and the information is submitted.

Post-condition

The item is added to the requisition list and is displayed with the other items in the requisition list.

Alternate flow

1. The customer selects an item to purchase from the catalog and is transferred to the Item Display page.
2. Under the Requisition list heading, the customer selects the **existing** radio button and clicks **Add to Requisition List**.
3. The customer is transferred to the Add to Requisition List page.
4. The system displays a page with the customer's requisition lists.
5. The customer selects a requisition list, clicks **Add to Requisition List**, and is transferred to the Edit Requisition List page.
6. The system displays the current items in the list as well as the new one selected.
7. The customer clicks **Place Order** and the requisition list is saved.

Post-condition

The item is added to the requisition list and the customer completes the order or continues shopping.

Remove items from a requisition list use case (Business Edition)

Business

This use case describes the functional flow of events when a customer updates items in a requisition list that they own.

Pre-condition

The customer has created at least one requisition list.

Basic flow

1. From any page, the customer clicks **Requisition List** on the header and is transferred to the Requisition List page.
2. The system displays a list of requisition lists owned by the customer.
3. The customer selects a requisition list and is transferred to the Edit Requisition List page.
4. The system displays all of the items in the selected requisition list.
5. The customer clicks **Remove** beside the item they wish to have removed.

Post-condition

The requisition list is updated.

Add items to an order use case (Business Edition)

Business

This use case describes the functional flow of events when a customer adds items to an order.

Pre-condition

The customer is logged on to the store.

Basic flow

1. The customer clicks **Browse Catalog** from the Home page and is transferred to the Main Category page.
2. The customer selects a category and is transferred to the category display page.
3. The system displays a page with the top category chosen and the main categories beneath it.
4. The customer selects a product and is transferred to the Product Display page.
5. The system displays a page with items within the product line chosen
6. The customer selects an item and is transferred to the Item Display page.
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 customer is transferred to the Your Order page.

Post-condition

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.

Create an order using a SKU use case (Business Edition)

Business

This use case describes the functional flow of events when a customer creates an order using a SKU.

Pre-condition

The customer is logged into the store.

Basic flow

1. The customer enters a number in the **Enter SKU** field under the Quick Order heading in the left navigation frame and clicks **GO**.
2. The customer is transferred to the Quick Order page.
3. The system displays a page with items entered for the current order.

Post-condition

The order is updated with the items for which a valid SKU was entered. If there is no current order a new order is created.

Exception

If the customer enters an invalid SKU, the system displays a message on the Quick Order page indicating that the SKU number entered was not found in the catalog.

Chapter 7. ToolTech troubleshooting

ToolTech sample store troubleshooting (Business Edition)

Business

This file contains common issues that may occur when setting up the ToolTech sample store. For further information, refer to the related topics in the online help or contact your help desk.

ToolTech publish fails when published the second time.

When the ToolTech store is published twice using the same organization, the second publish will fail. In order to avoid this you should do the following:

1. Create a new Organization using the Administration Console.
2. In Store Services, use this new organization as a seller organization when creating the SAR file.

Publishing should now work.

Note: Buyer organizations should not be used to publish the store. It is recommended that you create a new organization to be used as the store owner before you publish the ToolTech store. It is not recommended to use the default organization for publishing.

Registration Fails when LDAP is configured.

When you register a new customer in the ToolTech store, registration will fail if WebSphere Commerce is configured to run with LDAP. This is because the Buyer Organization A and Buyer Organization B have not been migrated to LDAP as yet. In order to avoid this, you should do the following:

1. Migrate Organization A and Organization B to LDAP before registering a new customer in the store.
2. Manually synchronize the LDAP registry with WebSphere Commerce.

The link to the Buyer Approval GUI for the Buyer Administrator role is not visible.

Buyer approval is enabled in the store by setting a Buyer Approval group and a user is assigned the role of Buyer (buy-side) and Buyer Administrator, however when logging into the store, the link to the Buyer Approval GUI is not visible. In order to avoid this, do the following:

1. Turn off WebSphere Commerce caching for the store. If you were running with a cache, you may have to clean up the WebSphere Commerce and WebSphere Application Server (WAS) cache directories.
2. Assign the role of Buyer Approver and Buyer (buy-side) to the customer.

You should now be able to see the link to the Buyer Approval GUI.

Note: For Order Approvals, customers must have Buyer (buy-side) and Buyer Approver roles.

Buyers pending approval are not visible in the Buyer Approval GUI.

When the Buyer Approver GUI is invoked and the **Approval** link is clicked, there are no buyers pending approval, even though a customer is registered under Buyer Organization A or Buyer Organization B. In order to avoid this, do the following:

1. Check the roles assigned to the user that you are using to approver other users.
2. Make sure approvers have the Buyer (buy-side) and Buyer Administrator roles assigned to them.

Note: For Order Approvals, customers must have Buyer (buy-side) and Buyer Approver roles.

When splitting an order the following message appears: “Purchase order .. is not a predefined purchase order...”

When an individual purchase order (PO) number is used for a split order, an error occurs and the order cannot be completed. To avoid this do the following:

1. Make sure that you use an individual PO number once only. Using this number more than once results in an error.
2. Use a blanket PO such as B1234567, instead of an individual PO.

Note: The blanket PO only applies to order items purchased under 2345.

The shipping mode created from the editor cannot be included in the contract.

In order for a newly created shipping mode to show in the Business Relationship Management (BRM) tool, you must create the shipping mode, and a policy for the shipping mode in the Shipping Notebook. You can manually create a policy for the shipping mode to ensure that the contract will accept it.

Note: The Shipping Notebook only allows for the creation of new shipping modes.

An excluded sub-category of products is still displayed in the store.

When you exclude a sub-category from a contract in the BRM, the excluded sub-category is still displayed in the store. If you click on the sub-category, no products will be displayed.

All products can be ordered even though a contract is set up to allow only a small subset of products.

Only one category is chosen to be included for sale for a given contract, but when the store pages are displayed, all categories and products are still available and orderable. To avoid this do the following:

1. Avoid using Buyer Organization A, since it supports the default contract which allows all products to be visible in the store. More restrictive contracts will not take affect in Organization A.
2. Repeat the process using Buyer Organization B.

Buyer Organization A has one contract in the Business Relationship Management (BRM) GUI, but the store displays two contracts.

When contracts are accessed through the account for Buyer Organization A, the account for Organization A shows one contract in the store, but customers registered under Buyer Organization A can see two contracts, 1234 (the default contract), and 2345, on the Item Display page.

Buyer Organization A itself only has one contract, but is enabled to support the default contract. In the GUI, only contracts belonging to this organization are displayed.

A Buyer Approver is logged off from ToolTech after logging out from the Buyer Approver GUI.

After a Buyer Approver approves the customers who are pending approval, logs out of the GUI, the store stops working. To work around this, do the following:

1. Log in as Buyer Approver again.
2. Do not log out when the Buyer Approval GUI window is closed.

Note: The reason the store stops working is because a cookie is shared between the two browser windows. If you log out from a launched GUI, you are also automatically logged out from the original window.

Clicking on the Request for Quotation (RFQ) GUI, causes a generic application error.

The RFQ link is not accessible by all customers. You need be assigned certain roles before this link is accessible. In order to prevent this error, do the following:

1. Check the roles assigned to the user. In order to access the RFQ GUI, you need to have a Buyer (buy-side) role assigned, and Internet Explorer 5.5.
2. Make sure each user only belongs to one organization.
3. Change the access control policies so that the user groups are not scoped by organization. This means, anyone who plays the Buyer role can create RFQs.

For this work around:

1. Go to the Administration Console.
2. Click on **Access Management**.
3. Click on **Policies**.
4. Select the policy:
RFQBuyersForOrgExecuteRFQCreateCommandsOnStoreEntityDataResourceGroup.
5. Click on **Change**.
6. Change the policy user group to RFQ Buyers.
7. Refresh the registry or re-start WebSphere Commerce.

Note: For another work around for RFQs, see the related link below.

Notices

Any reference to an IBM licensed program in this document is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Director of Licensing
Intellectual Property & Licensing
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
U.S.A.

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.

This document 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.

This product is based on the SET protocol.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Trademarks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both:

AIX	CICS	DB2
DB2 Extenders	Encina	HotMedia
IBM	iSeries	MQSeries

SecureWay
400

VisualAge

WebSphere

Blaze Advisor is a trademark of HNC Software, Inc. in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Lotus and Domino are trademarks of Lotus Development Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Oracle is a registered trademark of Oracle Corporation.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC. For further information see <http://www.setco.org/aboutmark.html>.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.