

IBM WebSphere Commerce



WebSphere Commerce Accelerator Customization Guide

Version 5.5

IBM WebSphere Commerce



WebSphere Commerce Accelerator Customization Guide

Version 5.5

Note:

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 21.

First Edition, March 2004

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

Order publications through your IBM representative or the IBM branch office serving your locality.

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

www.ibm.com/software/webservers/commerce/rcf.html

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

© Copyright International Business Machines Corporation 2002, 2003. All rights reserved.

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

Contents

About this book	v
Document description	v
Updates to this book	v
Conventions used in this book	v
Path variables	v
Knowledge requirements	vi

Part 1. Customizing the WebSphere Commerce Accelerator **1**

Chapter 1. WebSphere Commerce Accelerator architecture	3
WebSphere Commerce Accelerator run time model	3
Constituent assets	4
Common file assets	5

Data Persistence	6
User interface elements	6
Wizard and Notebook Framing	8
WebSphere Commerce Accelerator assets	9
WebSphere Commerce Accelerator definition files	9
WebSphere Commerce Accelerator home page files	10
WebSphere Commerce Accelerator components	11

Chapter 2. Customization guidelines **15**

Part 2. Additional resources **19**

Notices	21
Trademarks	22

About this book

Document description

This document serves as an overview of the concepts involved with customizing the WebSphere® Commerce Accelerator. It addresses the high level architecture of how the user interface interacts with the business users, and the WebSphere Commerce Server. Supplementary documents, released as they become available, build upon the knowledge developed in this document, and provide the detailed information required to customize the particular components of the WebSphere Commerce Accelerator. They also act as a resource, listing the components and assets upon which the various components depend.

Updates to this book

The latest version of this book, is available as a PDF file from the IBM® WebSphere Commerce technical library Web site:

www.ibm.com/software/commerce/library/

Conventions used in this book

This book uses the following highlighting conventions:

Boldface type	Indicates commands or graphical user interface (GUI) controls such as names of fields, icons, or menu choices.
Monospace type	Indicates examples of text you enter exactly as shown, file names, and directory paths and names.
<i>Italic type</i>	Used to emphasize words. Italics also indicate names for which you must substitute the appropriate values for your system.



This icon marks a Tip - additional information that can help you complete a task.

Important

These sections highlight especially important information.

Attention

These sections highlight information intended to protect your data.

Path variables

This guide uses the following variables to represent directory paths:

WC_installdir

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

- ▶ **AIX** /usr/WebSphere/CommerceServer55
- ▶ **400** /QIBM/ProdData/CommerceServer55
- ▶ **Linux** /opt/WebSphere/CommerceServer55
- ▶ **Solaris** /opt/WebSphere/CommerceServer55
- ▶ **Windows** C:\Program Files\WebSphere\CommerceServer55

WAS_installdir

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

- ▶ **AIX** /usr/WebSphere/AppServer
- ▶ **400** /QIBM/ProdData/WebAS5
- ▶ **Linux** /opt/WebSphere/AppServer
- ▶ **Solaris** /opt/WebSphere/AppServer
- ▶ **Windows** C:\Program Files\WebSphere\AppServer

WCDE_installdir

This is the installation directory for either the WebSphere Commerce Studio, or the WebSphere Commerce - Express development environment.

The default installation directory for WebSphere Commerce Studio is C:\WebSphere\CommerceStudio55.

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

Knowledge requirements

To customize the WebSphere Commerce Accelerator, you require knowledge of the following:

- Blaze Advisor rule technology
- HTML, JavaScript™, and XML
- Structured Query Language (SQL)
- Java™ Programming
- JavaServer Pages technology
- WebSphere Commerce Studio or WebSphere Commerce - Express Developer Edition

Please refer to the *WebSphere Commerce Programming Guide and Tutorials* for more information on customizing WebSphere Commerce. This book is available from the following Web site:

www.ibm.com/software/commerce/library

Part 1. Customizing the WebSphere Commerce Accelerator

This book describes how to customize the WebSphere Commerce Accelerator. By providing a conceptual overview of the design of the WebSphere Commerce Accelerator, this book teaches you how to approach the customization.

This book is intended to help guide you through customizing the WebSphere Commerce Accelerator, and introduce the different generic pieces that make up the WebSphere Commerce Accelerator, as well as explaining how to customize these pieces.

The WebSphere Commerce Accelerator is a collection of tools designed to facilitate the day-to-day business operations of your site. That is, it is intended as an interface for business people to create and modify any number of aspects of the site's operations without having to continually contact the IT staff responsible site. As such, there are numerous components that comprise the WebSphere Commerce Accelerator, each targeting a key aspect of the business. While every effort has been made to make these tools universal, this same goal means that some users may find that certain requirements are not met. If you extended the database during your site's creation, and these extensions relate to any of the tools listed in this document, the data will be invisible to the WebSphere Commerce Accelerator until you customize the relevant tool.

Chapter 1. WebSphere Commerce Accelerator architecture

The components that make up the WebSphere Commerce Accelerator are built using interface building blocks known as the *Tools Framework*. The widgets defined in this toolset are designed to simplify any customization efforts by providing an extensible and customizable framework allowing you to quickly implement a fully featured user interface while maintaining a common look and feel.

The framework consists of a server runtime component, and a collection of common user interface elements. The server component primarily acts to enforce security policies. The WebSphere Commerce Accelerator displays different menu options and pages depending on the user, and determining which content to display to which user is done by the WebSphere Commerce Server.

WebSphere Commerce Accelerator run time model

This section provides an overview of how the pieces that comprise the WebSphere Commerce Accelerator interact at run time. For the purposes of explanation, assume that a business user has just clicked **Change** on the Ad copy list page, after selecting an existing ad copy. The following figure illustrates the programming model for the WebSphere Commerce Accelerator run time:

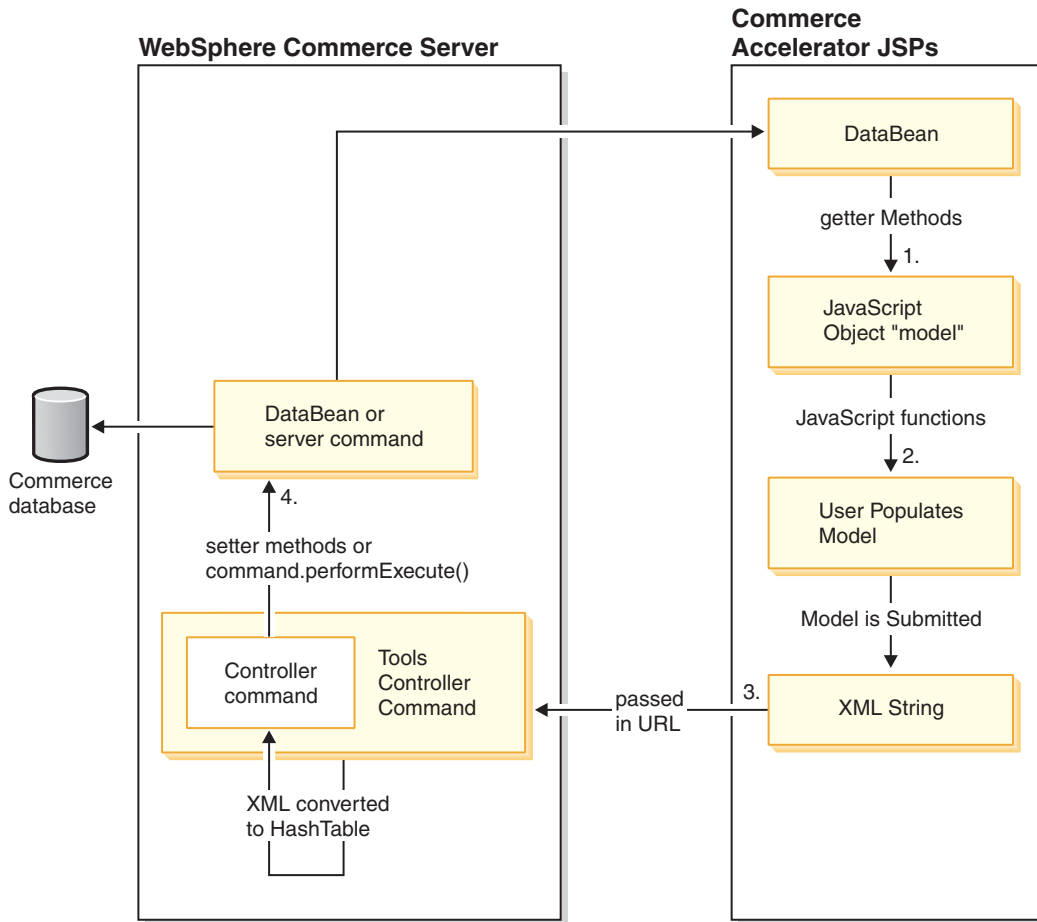


Figure 1. WebSphere Commerce Accelerator run time

1. Beginning at the top with the data bean, the JSP file corresponding to the Ad copy General definition page requests that the CampaignCollateralDataBean populate a JavaScript object in the JSP. This step is optional, since the requested JSP may not require data from the database. For example, if the business user in our example had clicked **Create**, instead of **Change**, the JSP file would not access the data bean since there would be no existing information for the new ad copy. Generically, refer to the component XML file to determine which data bean that the JSP files access.
2. The user further populates the JavaScript object by entering or changing data using HTML forms elements, which is transferred to the JavaScript object using JavaScript functions.
3. When finished, the JavaScript object is converted to an XML string, and passed as part of the URL, to the Tools Controller Command, which converts the XML string into a hashtable.
4. The Controller Command (a subclass of the Tools Controller Command) then sets a data bean or Server Command which updates the database.

Constituent assets

The WebSphere Commerce Accelerator is made up of numerous assets, as illustrated in Figure 1. These assets include the following:

JSP files

JSP files are used to display dynamic information.

JavaScript files

The WebSphere Commerce Accelerator uses JavaScript to perform numerous client-side functions. Also, the data gathered during these client-side interactions is stored as JavaScript, then converted to XML, before being passed back to the server.

XML files

XML files define the various components of the WebSphere Commerce Accelerator. Also, data transmitted between the client and the server is encoded as encrypted XML.

Java classes

Java classes are used to create commands, data beans, and access beans since WebSphere Commerce is a J2EE compliant application.

Properties files

Properties files are used as resource bundles. Separating the strings into properties files facilitates maintainability, and language flexibility. WebSphere Commerce is designed for use by multinational clients whose customer base often requires multiple languages.

Database registries

This is an important piece of the WebSphere Commerce Accelerator, and it is important to understand how it fits in to the tool. The database registries are the by-products of implementing a flexible architecture. Commands, URLs, and Views are all registered according to their storeID, among other data. While this flexibility is less applicable to a customized WebSphere Commerce Accelerator, it is still the framework in which the tools are defined.

Common file assets

The WebSphere Commerce Accelerator depends upon common files to help render the various components. These JSP files are located in a common location, available to all of the WebSphere Commerce Accelerator components. Note, these generic files should not be touched. These files control how the elements function, and changes to these files can easily break the WebSphere Commerce Accelerator.

To understand how the WebSphere Commerce Accelerator uses these files, consider the following example, in which a business user has clicked the menu item to create a new product in the Product Management tool.

When a business user clicks on the menu to initiate the ProductCreate wizard, it calls a URL command, which in this case, would be the ProductCreateWizardView command. The ProductCreateWizardView command opens the wizard.jsp file. This JSP initializes a databean, which accesses and parses the ProductCreateWizard.xml file. Based on the contents of the component's XML file, the wizard's table of contents can be obtained, and displayed using the wizardTOC.jsp. Similarly, the XML file governs the wizard's navigation panel. Once this is obtained, it is displayed using the wizardNavigation.jsp. For more information on how these elements display, refer to "Wizard and Notebook Framing" on page 8. Each of the user interface elements has a corresponding generic file, and in some cases like the wizard, more than one.

Other special files in this category include the ToolsUI.jsp, which acts as the overall, top level container in which the WebSphere Commerce Accelerator

displays. It contains the ToolsUIBanner, the ToolsUIMenu, and ToolsUIMain JSP files, which display the aspect of the WebSphere Commerce Accelerator to which their file name refers.

Once again, the files in this category should not be touched. This information is only shared to provide some context.

Data Persistence

Any work done while using the WebSphere Commerce Accelerator has to be saved to the WebSphere Commerce system. While all data generated using the WebSphere Commerce Accelerator is stored in the database, depending on the nature of the information, it might be stored in one of two formats. The first format in which data is stored is as simple flat data. This storage method is used for data that generally does not require additional processing prior to being used by the Commerce system. For instance, catalog data is stored in the database directly. This is also true of names and descriptions associated with any number of constructs, including for example campaigns, discounts, ad copy, and so on. The idea is that if the data can be obtained from the database, and used directly, then it should be stored in its most simple format.

The second format in which data is stored is as an XML representation of the data in question. This format is used to store the more complex data constructs generated by user interaction with the WebSphere Commerce Accelerator. The data stored in the database as an XML file typically requires some additional processing between when it is obtained from the database, and when it is used by the Commerce system. An example of this kind of data would be the data associated with a rule-based promotion, which must be processed by the rule engine within the Commerce system prior to its use in the external site.

User interface elements

The Tools Framework currently defines ten user interface elements, which are classified as either Container or Base elements. Container elements are those elements that provide a frame, permitting the containment of other elements and widgets. An example of a container element is a Notebook, which contains several panels. Base elements are those elements that can be placed inside a container element, or used elsewhere on an HTML page. An example of a base element is a Dynamic List.

The tools framework defines the following user interface elements:

Tools user interface center

The tools user interface center is the top level container element. The tools user interface center provides the menu system and user history navigation tools to the WebSphere Commerce Accelerator, as well as role-based access control. It also provides other visual cues, such as a progress indicator for page loading, a menu frame, in addition to the main content frame.

Wizard

The wizard is a container element. Wizards organize multiple pages of content. The wizard features a table of contents with a fixed navigation, intended to help the user keep their bearings. Wizards present a series of pages which must be completed in sequence, using **Previous** and **Next** buttons to navigate among the pages. Users cannot click on the table of contents to move to a specific panel, which is useful when you want the user to view and enter data into every panel. You have the flexibility to

define default values for fields, and if necessary, place a **Finish** button on a panel before the last panel in the series.

Wizards can also provide decision making mechanisms to end users, and can be linked to other wizards to accomplish larger tasks. This concept is called wizard chaining.

Notebook

The notebook is a container element. The notebook organizes multiple pages of content. Notebooks feature a table of contents from which end users can select the page that they want to view. Notebooks are typically used when the end user is expected to update large sets of information which do not require a specific sequence. Notebooks are also useful when users are updating information previously defined using a Wizard since users may want to skip directly to the panel containing the information they wish to change.

Because of the parallel, nonsequential nature of this element, it is not appropriate for use in situations that require logical branching in the task flow. For example, changing subsequent forms based on earlier choices.

Dynamic list

The dynamic list is a base element. It displays a list with navigation controls (**Previous** and **Next** buttons) to browse through multiple screens of data. You can configure the list so that end users can sort the displayed data using any of the displayed columns. The dynamic list also includes action buttons which perform defined actions on one or more items selected from the list. These buttons display on the right side of the list.

Use dynamic lists to display all of the defined assets of a given type. For example, when a user selects the **Auctions** menu item, the Auctions list page displays, listing all of the defined auctions, and the actions a user can perform.

Dialog

The dialog element is a container element. Dialogs consist of a single content page, most commonly used to display summary information, or request simple input data, such as search criteria.

Slosh bucket

The slosh bucket is a base element that allows users to select multiple objects from a list of available objects, which meet the selection criteria. The slosh bucket displays two lists side by side; one that contains the selected items, and another which displays the items from which the user is permitted to select. Four control buttons are displayed between the lists: **Add**, **Add All**, **Remove**, **Remove All**.

This element is generally used to pick a subset of unique entries from a larger set of data. Good examples of this are:

- Choosing which customer profiles are targeted by a suggestive selling initiative.
- Choosing the customer groups to which a discount applies.

Calendar

The calendar element is a base element. It is a tool that enables the user to select a date comprised of a year, month, and day. Provide a calendar element everywhere a user is expected to select a date. It is typically accessible through a calendar icon. For an example, refer to the Campaign initiative scheduling dialog.

Dynamic tree

The dynamic tree is a base element. It organizes and displays content in a manner similar to the tree view of the Windows® Explorer. The dynamic tree is capable of loading data from the database on demand, loading a specified number of levels and dynamically retrieving node leaves as their parents are expanded. This allows the user to browse data without having to wait for the whole tree to load before presentation. Each leaf node of the tree can have a context-sensitive menu which enables an end user to perform several actions.

Search dialog

The search dialog is a container element. It is used to collect data with which to search the database. This element contains one or more base elements in which users define the criteria upon which they want to base their search. The results are returned to the component from which the search dialog was launched.

Wizard and Notebook Framing

Wizards and notebooks use the following frameset definition:

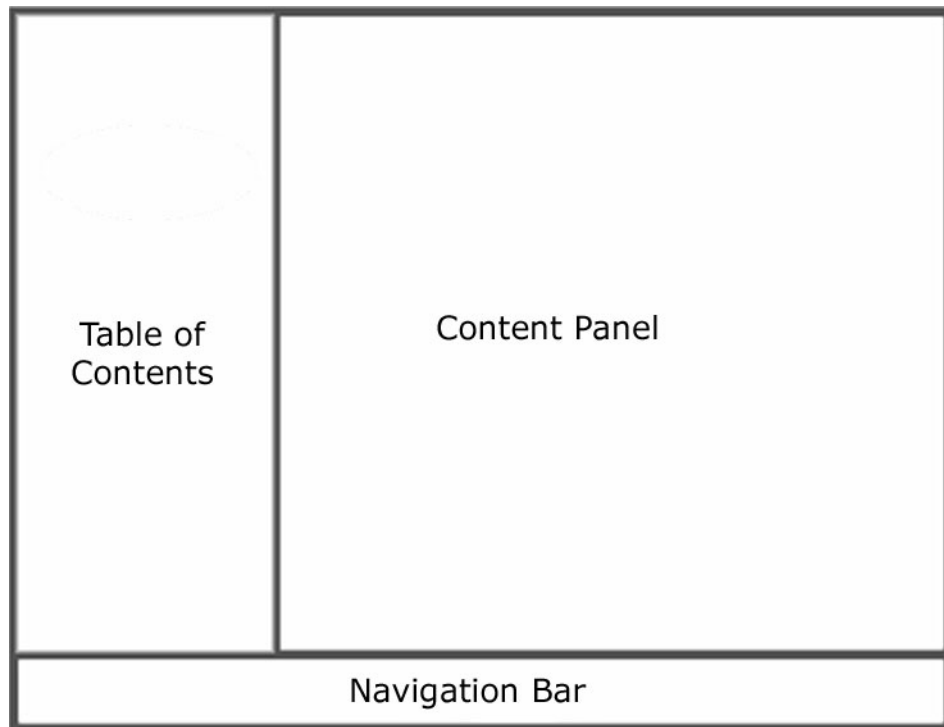


Table Of Contents Frame (Frame name: TABS)

This frame displays a list of the panel names. In a notebook, the user clicks on a panel name, changing the location of the content panel to that of the panel clicked.

Navigation Bar (Frame name: NAVIGATION)

This frame displays action buttons. When pressed, these buttons perform some action which applies to the container element as a whole. Examples include data submission, or cancelation of changes to a notebook. This frame also handles errors from the server if the submission fails, and can also perform some actions if the submission succeeds. An HTML form is used to submit data to the server. This form contains one element to hold

the submission data, however, additional elements can be added. This form is then submitted by the `finish()` function described by the outer frame.

Content Panel (Frame name: CONTENTS)

This frame displays panel pages.

This frameset is stored in a JavaScript "parent" object. All of the data entered by users on the various content panels is also stored in the outer frame, within a JavaScript object model, which is essentially a hashtable. Each time the content panel changes, all of the information from that frame is put into the object model.

WebSphere Commerce Accelerator assets












While the tools framework and the constituent elements define the WebSphere Commerce Accelerator, there are two additional asset types of which you must be aware. These include those assets responsible for rendering the WebSphere Commerce Accelerator, and those assets that the WebSphere Commerce Accelerator creates and modifies.

WebSphere Commerce Accelerator definition files

The WebSphere Commerce Accelerator is defined by numerous XML files, JSP files, and properties files. The XML files, which are further governed by DTD files, define the framework. The JSP files define the display content. All of the text strings are stored in properties files to facilitate national language translations.

The XML files that define the WebSphere Commerce Accelerator can be found in the `/WC_installdir/xml/tools/common` directory. While all of the XML files defining the WebSphere Commerce Accelerator framework are located in this directory, it would be uncommon that you would need to perform a customization on most of them, as the resulting change would affect every application of the WebSphere Commerce Accelerator. Still, by modifying the following files, you can modify the menu options that are available within the WebSphere Commerce Accelerator for any given user role.

Table 1. Primary WebSphere Commerce Accelerator definition XML files

File name	Version
CommerceAcceleratorB2C.xml	Consumer Direct Store
 CommerceAcceleratorB2B.xml	Business Direct Site
 CommerceAcceleratorCHS.xml	Channel Hub
 CommerceAcceleratorCPS.xml	Catalog Asset Store
 CommerceAcceleratorHCP.xml	Hosting Hub
 CommerceAcceleratorMHS.xml	Hosted Store
 CommerceAcceleratorMPS.xml	Hosted Storefront Asset Store
 CommerceAcceleratorRHS.xml	Reseller Store
 CommerceAcceleratorRPS.xml	Reseller Storefront Asset Store
 CommerceAcceleratorSCP.xml	Supplier Hub
 CommerceAcceleratorSHS.xml	Supplier Store
 CommerceAcceleratorSPS.xml	Supplier Asset Store

Attention

Applying fix packs, or performing migration may overwrite any changes made to these files.

Other files in this directory that you might want to customize include:

Table 2. Secondary WebSphere Commerce Accelerator definition XML files




File name	Function
AcceleratorHelpMap.xml	Contains links from pages within the tools to online help.
mclogon.xml	Contains the default logon flow configuration for Accelerator.
resources.xml	Defines all of the XML and resource bundle files used by the "common" namespace.
Roles.xml	Defines the user roles for access control.

Attention

Applying fix packs, or performing migration may overwrite any changes made to these files.

WebSphere Commerce Accelerator home page files

You might want to customize the home page for the WebSphere Commerce Accelerator that belongs to a particular instance. To locate the home pages, do the following:

1. Open the WebSphere Commerce development environment (**Start > Programs > IBM WebSphere Commerce development environment**) and switch to the  Business  Professional J2EE Navigator view  Studio Express Project Navigator view.
2. Navigate to the CommerceAccelerator project.
3. Navigate to the Web Content/tools/common directory.

The home pages have the following file names:

Table 3. WebSphere Commerce Accelerator home page files












File name	Version
AcceleratorHome.jsp	Default
AcceleratorHomeB2C.jsp	Consumer Direct
 Business AcceleratorHomeB2B.jsp	Business Direct
 Business AcceleratorHomeCHS.jsp	Channel Hub
 Business AcceleratorHomeCPS.jsp	Catalog Asset Store
 Business AcceleratorHomeHCP.jsp	Hosting Hub
 Business AcceleratorHomeMHS.jsp	Hosted Store
 Business AcceleratorHomeMPS.jsp	Hosted Storefront Asset Store

Table 3. WebSphere Commerce Accelerator home page files (continued)




File name	Version
 AcceleratorHomeRHS.jsp	Reseller Store
 AcceleratorHomeRPS.jsp	Reseller Storefront Asset Store
 AcceleratorHomeSCP.jsp	Supplier Hub
 AcceleratorHomeSHS.jsp	Supplier Store
 AcceleratorHomeSPS.jsp	Supplier Asset Store

These home page files contain the menu definitions for the WebSphere Commerce Accelerator. If the menu text in the WebSphere Commerce Accelerator does not reflect the terminology that you use in your company, these are the files that you need to update. Furthermore, these files refer subsequently to their respective resource properties files, which are located in the `/WC_installdir/instance.ear/properties/com/ibm/commerce/tools/properties` directory.

WebSphere Commerce Accelerator look and feel

The WebSphere Commerce Accelerator look and feel is governed by a cascading style sheet. The simplest customization of the WebSphere Commerce Accelerator is modifying this file to make changes to the format, font, or behavior of the text and other elements within the interface. In this file, you will also find URLs that point at the various background images.

To update the style sheet, do the following:

1. Open the WebSphere Commerce development environment (**Start > Programs > IBM WebSphere Commerce development environment**) and switch to the   J2EE Navigator view
 Express Project Navigator view.
2. Navigate to the CommerceAccelerator project.
3. Navigate to the Web Content/tools/common directory.
4. Locate the file named `centre_locale.css`, where *locale* represents the locale for which you want to make changes. For example, `centre_en_US.css` is the cascading style sheet for U.S. English.
5. Update the file as appropriate to reflect your changes.
6. Save your changes.
7. Redeploy the file.

Attention

Applying fix packs, or performing migration may overwrite any changes made to these files.

WebSphere Commerce Accelerator components

Typically, each component is made up of one or more JSP files, which display in the content panel. The navigation and table of contents frames rely on XML files for their content. These XML files also define the panels that make up the tools by referencing the component JSP files, and their respective properties files. The

sample code below details the Campaign wizard, and can be found in the `/WCDE_installdir/xml/tools/campaigns/` directory:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wizard SYSTEM "../common/WizardPanels.dtd" > 1
<!--
=====
* Licensed Materials - Property of IBM
*
* 5724-A18
*
* (c) Copyright International Business Machines Corporation. 2003
* All rights reserved.
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
=====
-->

<wizard resourceBundle="campaigns.campaignsRB" 2
        windowTitle="campaignWizardTitle"
        finishConfirmation=""
        cancelConfirmation="campaignWizardCancelConfirmation"
        tocBackgroundImage="/wcs/images/tools/toc/W_market.jpg"
        finishURL="CampaignSave" >

    <panel name="campaignGeneralPanel"
          url="CampaignGeneralPanelView"
          helpKey="MC.campaigns.CampaignGeneral.Help" /> 3



    <panel name="campaignBusinessPanel"
          url="CampaignBusinessPanelView"
          hasFinish="YES" 4
          helpKey="MC.campaigns.CampaignBusiness.Help" />

    <databean name="campaign"
              class="com.ibm.commerce.tools.campaigns.CampaignDataBean" /> 5

    <jsFile src="/wcs/javascript/tools/common/DateUtil.js" />
    <jsFile src="/wcs/javascript/tools/campaigns/Campaign.js" /> 6

</wizard>
```

Notes:

1. This defines the Document Type Definition (DTD). This file is located in `/WCDE_installdir/xml/tools/common/` directory.
2. This defines the properties file in which all of the user interface labels are defined.
3. This is the help key, which defines the key to refer to if the user clicks **Help**. These keys are defined in the `AcceleratorHelpMap.xml` file in the `/WCDE_installdir/xml/tools/common/` directory.
4. This specifies that the `CampaignBusinessPanel` has a **Finish** button.
5. This defines the data bean the wizard interacts with to get the campaign data from the database. You can specify multiple data beans.
6. These lines refer to JavaScript include files. To locate these files, do the following:
 - a. Open the WebSphere Commerce development environment (**Start > Programs > IBM WebSphere Commerce development environment**) and switch to the
 J2EE Navigator view
 Express Project Navigator view.
 - b. Navigate to the `CommerceAccelerator` project.
 - c. Navigate to the `Web Content/javascript/tools/component` directory, where `component` refers to the WebSphere Commerce Accelerator component with which you are working.
 - d. Update the appropriate JavaScript file as appropriate to reflect your changes.
 - e. Save your changes.

f. Redeploy the file.




Attention




Applying fix packs, or performing migration may overwrite any changes made to these files.




Chapter 2. Customization guidelines




The WebSphere Commerce Accelerator is a complex tool made up of different kinds of files and technologies. As a result, many customization scenarios for the tool are not trivial tasks. You must ensure that you have fully considered the future implications of a given customization. While IBM strives to ensure compatibility between assets created using previous and current versions of the tools, IBM cannot anticipate the breadth or depth of customer developed customization scenarios. Previous customization may cease to function when an updated version of the tool is released.

Consideration, therefore, should be given to the following strategy recommendations for customization. The information contained in this table represents the best practices with respect to WebSphere Commerce Accelerator customization, based on the current design.

File type	Recommended Customization Approach
JSP files	<ol style="list-style-type: none">1. Open the WebSphere Commerce development environment (Start > Programs > IBM WebSphere Commerce development environment) and switch to the  Business  Professional J2EE Perspective Project Navigator view  Studio Express Project Navigator view.2. Navigate to the CommerceAccelerator project.3. Navigate to the Web Content/tools/ directory.4. Create a custom folder.5. Copy the target file from the appropriate component folder to your custom folder.6. Double-click the new file in the custom folder to open it in the editor window.7. Update the file as necessary.8. Update the VIEWREG database table to point to the customized version of the file.

File type	Recommended Customization Approach
Properties files	<ol style="list-style-type: none"> Open the WebSphere Commerce development environment (Start > Programs > IBM WebSphere Commerce development environment) and switch to the <ul style="list-style-type: none">   J2EE Perspective Project Navigator view  Express Project Navigator view. Navigate to the WebSphereCommerceServerExtensionsLogic project. Navigate to the src folder. Right-click on the src folder, and select New, and then Package. In the Name Field on the New Java Package dialog, enter a unique name. For the purposes of this scenario, enter <code>com.myCompany.component.properties</code>, where <i>myCompany</i> represents some custom directory name, and <i>component</i> represents the tool component to which the properties file corresponds. Packages created by IBM, and included in WebSphere Commerce follow a naming convention which begins, "com.ibm..." Click Finish to create the package. Right-click on the <code>com.myCompany.component.properties</code> folder, and select New, and then Other. In the New dialog, click Simple, File, and then Next. In the File name field, enter a unique name. For the purposes of this scenario, enter <code>componentRB_locale.properties</code>, where <i>component</i> represents the tool component to which the file corresponds, and <i>locale</i> represents the locale from which your business users will access the WebSphere Commerce Accelerator. Click Finish to create the file and open it in an editor. Update the file as necessary. Update the <code>resources.xml</code> file, as documented below, to include the following update: <pre data-bbox="678 1150 1247 1220" style="font-family: monospace;"> <resourceBundle name="campaignsRB"> <bundle>com.ibm.commerce.tools.campaigns.properties.CampaignsRB</bundle> <bundle>com.myCompany.component.properties.componentRB</bundle> </resourceBundle> </pre>
XML files	<p>If you customize XML files that were originally shipped with the product, you should make the updates directly to the file in question. However, you must be aware that migration will overwrite your changes. If you wish to keep you changes, you must make backup copies of any changed files, and reapply the changes to your new files.</p> <p>If you need to create new XML files, do the following:</p> <ol style="list-style-type: none"> Create a new directory in which you will store your custom XML files. Copy the xml file from your <code>/WCDE_installdir/xml/tools/component</code> directory, into your custom directory. You must copy the file into a subdirectory with a name that matches the original component name. Open the custom <code>resources.xml</code> file in an editor. Update the file as necessary. Update the XMLPath value in the <code>instance_name.xml</code> file such that you prepend the custom directory path to the existing list of paths.

File type	Recommended Customization Approach
JavaScript files	<p>If you customize JavaScript files that were originally shipped with the product, you should make the updates directly to the file in question. However, you must be aware that migration will overwrite your changes. If you wish to keep you changes, you must make backup copies of any changed files, and reapply the changes to your new files.</p> <p>If you need to create new JavaScript files, do the following:</p> <ol style="list-style-type: none"> 1. Open the WebSphere Commerce development environment (Start > Programs > IBM WebSphere Commerce development environment) and switch to the <ul style="list-style-type: none">  Business  Professional J2EE Perspective Project Navigator view  Studio Express Project Navigator view. 2. Navigate to the CommerceAccelerator project. 3. Navigate to the Web Content/javascript folder. 4. Right-click on the javascript folder, and select New, and then Folder. 5. In the Folder name field, enter custom. 6. Repeat this process until you have created the following path: Web Content/javascript/custom/<i>component</i> where <i>component</i> represents the tool component to which the JavaScript file corresponds. 7. Right-click on the custom/<i>component</i> folder, and select New, and then JavaScript file. 8. In the File name field, enter a unique name. For the purposes of this scenario, enter <i>component.js</i>, where <i>component</i> represents the tool component to which the file corresponds. 9. Click Finish to create the file and open it in an editor. 10. Update the file as necessary. 11. Update any JSP files as appropriate such that they reference the new JavaScript file in the new custom folder.

File type	Recommended Customization Approach
Image files	<p>If you customize image files that were originally shipped with the product, you should replace the files directly with your updates. However, you must be aware that migration will overwrite your changes. If you wish to keep you changes, you must make backup copies of any changed files, and reapply the changes to your new files.</p> <p>If you need to create new JavaScript files, do the following:</p> <ol style="list-style-type: none"> 1. Open the WebSphere Commerce development environment (Start > Programs > IBM WebSphere Commerce development environment) and switch to the <ul style="list-style-type: none">  Business  Professional J2EE Perspective Project Navigator view  Studio Express Project Navigator view. 2. Navigate to the CommerceAccelerator project. 3. Navigate to the Web Content/images folder. 4. Right-click on the images folder, and select New, and then Folder. 5. In the Folder name field, enter custom. 6. Repeat this process until you have created the following path: Web Content/images/custom/<i>component</i> where <i>component</i> represents the tool component to which the image file corresponds. 7. Right-click on the custom/<i>component</i> folder, and select New, and then Image file. 8. In the File name field, enter a unique name. For the purposes of this scenario, enter <i>imageName</i>, where <i>imageName</i> represents some custom image name. 9. Using the File type field, select the image type for the file that you want to create. 10. Click Finish to create the file and open it in an editor. 11. Update the file as necessary. 12. Update any JSP files as appropriate such that they reference the new image file in the new custom folder.
Commands	<ol style="list-style-type: none"> 1. In the WebSphere Commerce development environment, navigate to the WebSphereCommerceServerExtensionsLogic project. 2. Right-click on the project, and select New, and then Source Folder. 3. In the Folder Name field on the New Source Folder dialog, enter myCustomSource. 4. Click Finish to create the new folder. 5. Right-click on the new folder, and select New, and then Package. 6. In the Name Field on the New Java Package dialog, enter <i>com.myCompany.logic.component</i>, where <i>myCompany</i> represents some custom directory name. 7. Click Finish to create the new package. 8. Right-click on the new package, and select New, and then Class. 9. In the Name field, enter <i>MyCommandNameControllerCmdImpl</i>. 10. In the Superclass field, enter the following: <i>com.ibm.commerce.tools.component.CommandNameControllerCmdImpl</i>. 11. Click Finish. The file opens in the editor. 12. Update the CMDREG database table to reference the new controller command class.

Part 2. Additional resources

This document serves as an introduction to customizing the WebSphere Commerce Accelerator. While the information provided here should give you some understanding of how the tools were designed, the following documents are intended to provide the lower level details required to customize the various components that comprise the WebSphere Commerce Accelerator. The following supplemental documents are available:

Tools User Interface Customization

Details general WebSphere Commerce User Interface widget customization.

Campaigns

Details the customization of campaigns, campaign initiatives, e-Marketing Spots, ad copy, and campaign scheduling.

Catalog management

Details the customization of catalog management tools.

Reporting

Details the customization of business intelligence and the reporting framework.

Furthermore, you should refer to the *WebSphere Commerce Store Development Guide*, and the *WebSphere Commerce Programming Guide and Tutorials* as necessary.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:

*IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.*

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Canada Ltd.
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
Canada*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Trademarks

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

400	AIX	IBM
iSeries	OS/400	WebSphere

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

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

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



Printed in USA