IBM WebSphere for Digital Media Enabler for Multiplatforms

WebSphere Digital Media Enabler Customization Guide

Version 5.5

IBM WebSphere for Digital Media Enabler for Multiplatforms

WebSphere Digital Media Enabler Customization Guide

Version 5.5

Note

Before using this information and the product it supports, be sure to read the general information under Appendix C, "Notices," on page 65.

First Edition (October 2003)

This edition applies to IBM WebSphere Digital Media Enabler for Multiplatforms Version 5.5 Release 3 (Program 5724–E19) 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 your product.

Contents

Chapter 1. Introduction 1
Target audience 1
Related publications 1
Understanding the WebSphere Digital Media Enabler store 1
Installation Directories 2
Document conventions 2

Chapter 2. Areas of customization . . . 5
Changing the store front 5
Setting up e-mail and store data 6
  User data 6
  Pricing 6
  E-mail and notification 7
  Subscriptions 7
  Media Folder collaboration and Sametime™ . . . 7
Content management 7
Coding apostrophes for translation 8

Chapter 3. Understanding WebSphere Digital Media Enabler directory structures 11
Understanding the JSP directory structure 11
Understanding the web objects directory structure 15
Understanding the WebSphere Digital Media Enabler proxies directory structure 17

Chapter 4. Customizing the store front appearance. 19
Header, sidebar, and footer Web page areas. . . . 19
Main body page area 20
  Affected JSP files 20
  JSP and INC files 20
  Creating the customer-provided static pages . . 21

Chapter 5. Customizing the e-mail and store data setup. 23
Customizing the user registration data 23
Customizing the usage pricing data 26
  Pricing description 26
  Database tables for usage pricing 27
  Contract pricing 30
Delivery and e-mail 31
  Configuring the delivery daemon 31
  Customizing the e-mail daemon and e-mail templates 34
Subscriptions 38
  Setting up the system for subscription models . . 38
  Use cases 38
Setting Media Folder collaboration using a Sametime server 42

Chapter 6. Customizing content management 45
Creating the Content Manager WebSphere Digital Media Enabler item type 45
  Using the content loading utility 45
  Content Manager parameters 45
  Defining your field attributes for loading into Content Manager 46
  Defining your upload user responsibilities . . . 47
  Replicating metadata into the WebSphere Commerce catalog 49
  Defining your fields for loading ACSII delimited input text data 49
  Defining your fields for loading via XML input 50
Integrating with external Content Management systems. 51
  Integrating using WebSphere Digital Media Enabler as the front-end system 51
  Integrating using the external repository as the front-end system 51
  Integrating an external content management system 52
Customizing the ZIP utility for delivery to Macintosh customers 53

Appendix A. Sample subscription product XML 55

Appendix B. The following is a sample Virage HTML template 61

Appendix C. Notices 65
Trademarks 66

Glossary 69

Index 71

Chapter 1. Introduction

Target audience

This document assumes that you are a Java™ web programmer familiar with the functions of WebSphere® Digital Media Enabler, WebSphere Commerce Business Edition, Professional Edition, and IBM® DB2 Content Manager. If you need more information about using these products, refer to the WebSphere Commerce Business Edition, Professional Edition, Content Manager, or WebSphere Digital Media Enabler technical documentation. This document functions as a supplement to those documents and explains the tasks that need to be performed for each WebSphere Digital Media Enabler store.

Related publications

- *WebSphere Commerce Fundamentals, Version 5.5*
- *WebSphere Commerce Quick Beginnings for Windows® 2000, Version 5.5*
- *WebSphere Commerce Quick Beginnings for AIX®, Version 5.5*
- *WebSphere Commerce Quick Beginnings for Solaris, Version 5.5*
- *WebSphere Commerce Programming Guide and Tutorials, Version 5.5*
- *WebSphere Commerce Store Development Guide, Version 5.5*
- *WebSphere Digital Media Enabler Installation and Configuration Guide, Version 5.5*
- *WebSphere Digital Media Enabler Administration Guide, Version 5.5*

Understanding the WebSphere Digital Media Enabler store

WebSphere Digital Media Enabler is a store running on top of the WebSphere Commerce infrastructure and, in essence, is similar in implementation to a standard WebSphere Commerce store.

The WebSphere Digital Media Enabler store differs from a typical WebSphere Commerce store in five areas:

1. Searching the catalog: Because we are dealing with digital assets and unstructured data, a WebSphere servlet application that accesses IBM Content Manager searches and retrieves digital content and its related metadata.
2. Collaboration with media folders: This function is technically similar to a wish list or a gift registry in e-commerce, but follows a point-to-point communication model for collaboration. This function is specific to the media industry.
3. Usage pricing: Fixed catalog pricing is not common when it comes to digital media. With digital media, pricing is not based on pure manufacturing costs but more on the creative value of the asset and licensing/usage conditions imposed by the creator.
4. Additional payment options: These include applying purchases toward a credit card or corporate purchase order, and generating and sending electronic invoices.
5. Uploading and indexing of digital content by content providers

The WebSphere Digital Media Enabler package contains a Digital Media SAR file that will be used to publish a store on WebSphereCommerce using the WebSphere Commerce Administration console.

Installation Directories

WAS_installdir represents the following default installation path for WebSphere Application Server:

▶ 2000

2000 *drive*:\Program Files\WebSphere\AppServer

▶ AIX

AIX/usr/WebSphere/AppServer

▶ Solaris

Solaris/opt/WebSphere/WAS_installdir

WC_installdir represents the following default installation directory for WebSphere Commerce:

▶ 2000

2000 *drive*:\Program Files\WebSphere\CommerceServer55

▶ AIX

AIX/usr/lpp/WebSphere/CommerceServer55/

▶ Solaris

Solaris /opt/lpp/WebSphere/CommerceServe55/

DM_installdir represents the following default installation directory for WebSphere Digital Media Enabler:

▶ 2000

2000 *drive*:\Program Files\WebSphere\CommerceServer55\wdme

▶ AIX

AIX/usr/lpp/WebSphere/CommerceServer55/wdme

▶ Solaris

Solaris/opt/WebSphere/CommerceServer55/wdme

Document conventions

This document uses the following conventions:

Boldface type indicates commands or graphical user interface (GUI) controls such as names of fields, buttons, or menu choices.

`Monospace type` indicates examples of text that you enter exactly as shown as well as file names and directory paths.

Italic type is used for emphasis and for variables for which you substitute your own values.

The word *locale* is used to direct the user to substitute specific indicators for the user's language and region. The following table lists the values that should be used for each *locale*.

Table 1. Locale indicators

Country/Region	**Language code**	**Region code**
Brazil	pt	BR
China	zh	CN
France	fr	FR
Germany	de	DE
Italy	it	IT
Japan	ja	JP
Korea	ko	KR
Spain	es	ES
Taiwan	zh	TW
United States	en	US

Chapter 2. Areas of customization

This document describes four main areas of customization. The areas are:

- Changing and setting up the store front
- E-mail and store data
- Managing content
- Coding apostrophes for translation

An overview of each area follows. Subsequent chapters address each area in detail.

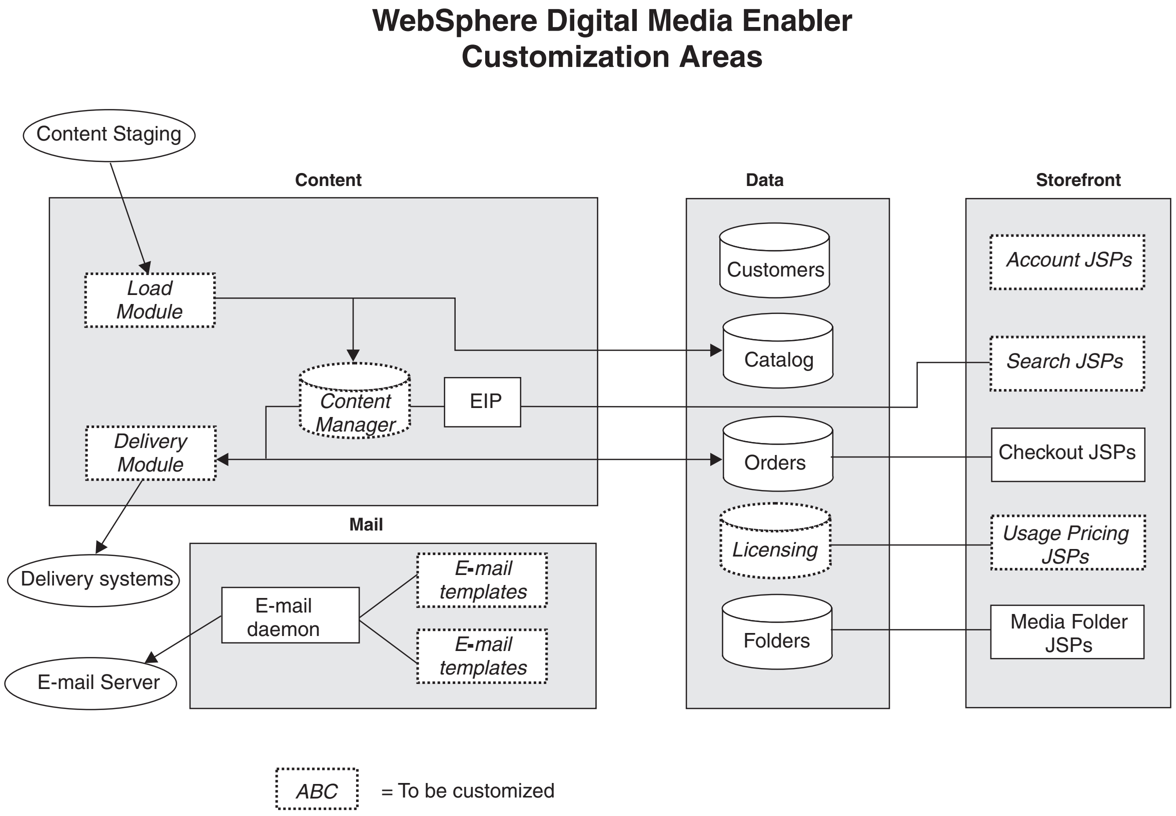

Figure 1. WebSphere Digital Media Enabler customization areas

Changing the store front

The store front assigns a specific look and feel to your WebSphere Digital Media Enabler. It is usually dictated by the specific customer requirements. This customization can be nominal or massive depending on the store requirements.

WebSphere Digital Media Enabler provides header, footer, navigation sidebar, and style sheet designs that enable an out-of-the-box implementation. This level of implementation requires the least effort. You supply only your own image graphics and style sheets to dictate the unique branding or look-and-feel for your store. To change the design of the web pages, such as the layout of search results or the layout of shopping cart page, the customization effort is somewhat larger. The navigation scheme from one screen to another is also an area where you may need additional changes; however, to implement a fully functional Digital Media store, you only need to change the look and feel to make your store unique.

To customize your store, you can:

- Modify the store header, footer, and side bar include files. These provide the store-wide consistency for general look and navigation.
- Modify the style sheets. The style sheets provide a mechanism for setting the colors and fonts for all of the store web pages.
- Customize the search JSP files to fit the digital content data model.
- Customize the registration JSP files to reflect all of the customer profile information that you want to collect during registration.
- Modify any other JSP files that you have elected to change. This step is optional and is required only if you decided to change other pages as a result of unique requirements for your store.
- Create HTML for static informational text pages that are to be included in your store. These typically include Help, About Us, Privacy Policy, and Terms and Conditions pages.

Setting up e-mail and store data

In addition to changes in the store front, you can change the e-mail and store data settings in WebSphere Digital Media Enabler to further customize your store, by changing the following:

- User registration data
- Pricing: Usage and contract
- E-mail and notification
- Subscriptions
- Media Folder collaboration

User data

As in the implementation of a typical WebSphere Commerce store, user data customization allows you to select which type of registration and profile data you would like to capture about your users. WebSphere Digital Media Enabler captures some default user information. You will map these data requirements to existing columns in the WebSphere Commerce user registration tables.

The JSP files related to User Account will be the focus of customization in this case.

Pricing

WebSphere Digital Media Enabler offers both usage and contract pricing. WebSphere Digital Media Enabler performs usage pricing by having a set of usage questions, a set of answers associated with each question, and a price multiplier associated with each answer. This information is stored in the database. Customization is involved to enter the various questions, the answers, and multipliers into the relational database tables. The usage tables are WCDMUSAGEQUESTIONS, WCDMUSAGEANSWERS, WCDMUSAGEDEF, WCDMUSAGE, and WCDMUSAGEPERCENT. You may need to customize the usage pricing JSP files as well.

As a user of WebSphere Commerce , you can take advantage of certain business-to-business (B2B) features such as contract pricing based upon contracts associated between your business and your business customers.

E-mail and notification

WebSphere Digital Media Enabler's e-mail functions are classified in two categories:

- Media Folder e-mail: An e-mail daemon will send Media Folder e-mail messages to the e-mail server. Customize to alter the formatted text of the message body and the subject line.
- Notifications: Leverage the WebSphere Commerce notification subsystem for the following events:
  - Order status change
  - User registration confirmation
  - Order confirmation

To customize, you must change the e-mail body text and subject line in the `mailprocess_`*`locale`*`.properties file`.

Configure the e-mail server name externally via the `mailprocess.properties` file.

Subscriptions

In addition to the pricing methods listed previously, you can customize WebSphere Digital Media Enabler to offer your customers the option of purchasing a subscription. WebSphere Digital Media Enabler supports three types of subscriptions:

- **Duration**: The user can buy the subscription for a particular duration of time.
- **Amount**: The user can buy the subscription for a particular amount.
- **Number of items**: The user can buy the subscription for a particular number of images.

Media Folder collaboration and Sametime™

Users can store images in the Media Folder for purchasing, for viewing later, and for sharing with other users. Users can communicate using Sametime while viewing the contents of a Media Folder.

Content management

Each store has its own specific data model to describe and index the digital assets that they have. You must create an item type in Content Manager that will define all of the metadata for your digital content. At a minimum, this should contain the fields to be displayed on the **Item Details** page as well as all fields that will be used as search filters on the **Advanced Search** page. Do not forget the categorization fields that are included in the **Browse Categories** page. You also need to define the fields that are to be included in a full-text search, generally `Description` or `Caption` and keywords. Once this is determined, you will need to change the JSP files related to the search functions as well as the Content Load utility.

WebSphere Digital Media Enabler has a content loading utility that will load the assets into Content Manager. If watermarking is desired, you will need to add code to invoke the Content Manager watermark feature. The utility also provides the XML file needed to input into the WebSphere Commerce catalog loading utility that needs to be invoked separately.

The content loading utility Java code needs customization to map the fields of the metadata input file prepared for your content to the fields in the Content Manager's item type that you have created. Perform this customization step by updating a table that defines all columns to be loaded. If watermarking is desired, you must provide a watermark bitmap.

If you will not be using Content Manager and will be using the WebSphere Digital Media Enabler session bridge feature to access your own repository, you will still need to load product data into your WebSphere Commerce database. You will most likely extract the required product information (such as content ID, description, etc.) from your existing repository and format XML input to the WebSphere Commerce loader.

The default WebSphere Digital Media Enabler delivery module retrieves the ordered assets from Content Manager, zips them up, and allows for an HTTP download to the user's browser. It then updates the order status to **delivered**.

You will need to change this code to accommodate other delivery methods such as an FTP site, CD-ROM burning, streaming servers, and so on. The destination is always a directory whether it is going to be served by an HTTP server, a Real server, an FTP server or a folder for manual CD burning. There is a configurable parameter for specifying the destination directory for the delivery module.

If you are using the session bridge feature to access an external content repository, you will need to provide the delivery function yourself or modify the WebSphere Digital Media Enabler delivery module to access the external repository.

1. Define metadata attributes for your content that are required to support your store.
2. Identify which attributes are multi-valued and which ones are to be searchable.
3. Define which attributes are to be replicated into the WebSphere Commerce product catalog.
4. Customize the delivery daemon parameters to specify how content will be delivered to customers.
5. Define how content will be priced by setting the values in the usage pricing tables.
6. Set up e-mail properties to specify how users will be notified when content has been ordered.

Coding apostrophes for translation

Apostrophes in property or JSP files do not need hard coded character. For example, the escape character in the following message is not necessary:

```
Don/'t do this.
```

It should be written as:

```
Don't do this.
```

Modifying or creating JSP or property files requires that the following steps be implemented to prevent any translation problems with apostrophes for languages other than English:

1. Import DMStringUtil and DMMessageFormat into any JSP, INC, or Java class in which they will be called. For example:

   ```
   <%@page
   import="com.ibm.commerce.digitalmedia.common.util.DMStringUtil"%>
   ```

   ```
   <%@page
   import="com.ibm.commerce.digitalmedia.common.util.DMMessageFormat"%>
   ```

2. If a JavaScript string is from `Resourcebundle.getString()`, use `DMStringUtil` method to wrap the string.

   Example:

   ```
   alert('<%=DMStringUtil.convertStringForjavascript(DMstoretext.getString("MSG"))%>');
   ```

3. Replace instances of `MessageFormat.format()` with `DMMessageFormat`.

   Example:

   ```
   DMMessageFormat.format(DMstoretext.getString("REVIEW_ORDER"),args);
   ```

Chapter 3. Understanding WebSphere Digital Media Enabler directory structures

In order to make customization changes, it is important to understand the structure of the following WebSphere Digital Media Enabler directories:

- JSP directory
- Web objects directory
- Proxies directory

Understanding the JSP directory structure

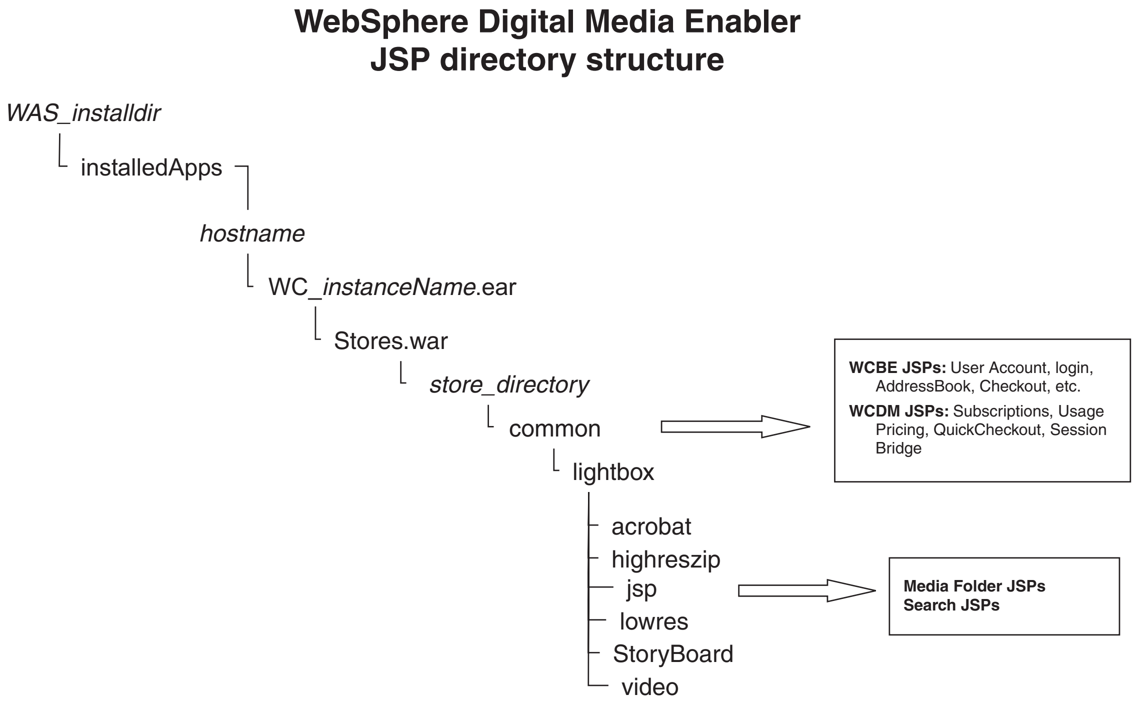

Figure 2. WebSphere Digital Media Enabler JSP directory structure

WebSphere Digital Media Enabler augments the typical WebSphere Commerce store with three additional function sets to accommodate digital media requirements. Those three function sets are:

- Media Folder functionality: The JSP files for this function are located in the following subdirectory:

```
was_installdir/installedApps/hostname/WC_instance_name.ear/
Stores.war/store_directory/common/lightbox/jsp
```

- Digital Media Asset Search functionality: The JSP files for this function are located in the following subdirectory:

```
was_installdir/installedApps/hostname/WC_instance_name.ear/
Stores.war/store_directory/common/lightbox/jsp
```

- Extended functions added to WebSphere Commerce, such as usage pricing and subscriptions, to enable e-commerce for digital media: The JSP files for these functions are in the same subdirectory as the typical store's JSP files:

```
was_installdir/installedApps/hostname/WC_instance_name.ear/
 Stores.war/store_directory/common
```

The following tables list specific JSP file names and corresponding descriptions.

Table 2. Media Folder JSP files

JSP Files	Description
DMFolder2SessionBridge.jsp	Session bridge
DMFolderCopy.jsp	Perform the copying of media folders
DMFolderCreate.jsp	Perform the creation of media folders
DMFolderCreateSuccess.jsp	Create media folder success page
DMFolderEmail.jsp	Perform the e-mailing of media folders
DMFolderEmailSuccess.jsp	E-mail media folder success page
DMFolderEmailView.jsp	Perform the e-mailing of media folders
DMFolderManager.jsp	Display the media folder manager
DMFolderMultiDelete.jsp	Perform the deletion of multiple folders
DMFolderRename.jsp	Perform the renaming of media folders
DMFolderSessionBridge.jsp	Session bridge
DMFolderShare.jsp	Perform the sharing of media folders
DMFolderShareSuccess.jsp	Share media folder success page
DMFolderView.jsp	Display a media folder
DMFolderUnshareSuccess.jsp	Perform the unsharing of media folders

Table 3. Search JSP files

JSP files	Description
DMSearchBrowseCat.jsp	Browse category page
DMMediaDownloadComplete.jsp	Download complete page
DMMediaDownload.jsp	Download specifications page
DMSearchDetail.jsp	Display the Item Detail page
DMSearchResult.jsp	Display the search results
DMSearchpopup.jsp	Popup window containing the preview
DMSearchEntry.jsp	Advanced search page
DMMediaDownloadWait.jsp	Download wait message
DMQuickSearch.jsp	Quick search form
DMSearchSessionBridge.jsp	Session bridge
DMSearchMediaDeleted.jsp	Search deleted media

Table 4. Sametime JSP files

JSP files	Description
DMSametime.jsp	Sametime
DMSametimeFooter.jsp	Sametime footer
DMSametimeHeader.jsp	Sametime header

Table 5. Upload Manager JSP files

JSP files	Description
DMLibrary_Create.jsp	Perform the creation of upload libraries
DMLibrary_Image_details.jsp	Display upload library image details
DMLibraryImageDetailsSuccess.jsp	Display image details
DMLibrary_Image_Info.jsp	Display upload library image information
DMLibrary_Image_Info(old).jsp	Display upload library image information
DMLibrary_Img_Upload.jsp	Upload a digital object to library
DMLibrary_Manager.jsp	Display the upload manager
DMLibrary_Rename.jsp	Rename a library
DMLibraryThumbImageCreate.jsp	Create a thumbnail
DMLibraryThumbImageUpload.jsp	Upload a thumbnail
DMLibrary_Upload_Success.jsp	Upload successful
DMLibrary_View.jsp	View the contents of an upload library
DMLibraryApproval.jsp	Approve the content of an upload library
DMLibraryApprovalDetails.jsp	Approve details of an upload library

Table 6. JSP files for Digital Media e-commerce

JSP files	Description
confirmation.jsp	Order confirmation page
confirmationwcs.jsp	Normal WebSphere Commerce Business or Professional Edition confirm page
subscriptionquickcheckout.jsp	Subscription quick checkout
DMDisplaySubscription.jsp	Display a subscription
DMNewSubscriptionCheckout.jsp	Checkout a new subscription
DMNewSubscriptionConfirmation.jsp	Subscription confirmation
DMSubscription.jsp	Subscription
DMSubscriptionCancel.jsp	Cancel a subscription
DMSubscriptionCancelMain.jsp	Cancel confirmation
DMSubscriptionCheckout.jsp	Subscription checkout
DMSubscriptionConfirmation.jsp	Subscription confirmation
DMSubscriptionMain.jsp	Main subscription
DMExternalRegistration.jsp	Session Bridging of WebSphere Digital Media Enabler to other DAM
DMExternalInterface.jsp	Session Bridging of WebSphere Digital Media Enabler to other DAM
DMOrderDisplayCompleted.jsp	Order details from Order History list

Table 6. JSP files for Digital Media e-commerce (continued)

JSP files	Description
DMOrderList.jsp	Order History List
edit_registration.jsp	Registration/subscription page
imagecart.jsp	Shopping cart page
OrderDisplayPending.jsp	Check out page (main JSP)
OrderDisplayPendingWCS.jsp	Normal WebSphere Commerce Business or Professional Edition checkout page
OrderDisplayQuestions.jsp	Usage questions page
OrderItemDisplay.jsp	Shopping cart (main JSP)
OrderOK.jsp	Confirmation page (main JSP)
OrderPopUp.jsp	Usage Questions processing (hidden page)
ProductDisplay.jsp	Display merchandise related to a digital asset
usagePopup.jsp	Usage popup
usagequestions.jsp	Usage questions for a single item
usagequestionsall.jsp	Usage questions for entire order
userRegistrationForm.jsp	Registration page (main JSP)

Table 7. Typical WebSphere Commerce store JSP files used in WebSphere Digital Media Enabler

JSP files	Description
account.jsp	User registration page
AddBillAddress.jsp	Add a new address
address.jsp	Address display page
addressbook.jsp	Address book display page
AddressBookForm.jsp	Address book (main JSP)
AddressForm.jsp	Address item (main JSP)
billingaddress.jsp	Billing address display
emptyshopcart.jsp	Empty shopping cart page
footer.jsp	Web page footer
forgetpassword.jsp	Forget password page
forgetpassword_err.jsp	Error page for forget password page
GenericError.jsp	WebSphere Commerce Business or Professional Edition error page
getResource.jsp	WebSphere Commerce Business or Professional Edition internal page
header.jsp	Web page header
home.jsp	Main Web home page
LoginForm.jsp	Process login
Logoff.jsp	Process logoff
logonFail.jsp	Process failed login
password.jsp	Change password page
PasswordNotify.jsp	Notify password change
register.jsp	Registration page

Table 7. Typical WebSphere Commerce store JSP files used in WebSphere Digital Media Enabler (continued)

JSP files	Description
RegisterForm.jsp	Registration page (main JSP)
ResetPasswordError.jsp	Error for reset password
ResetPasswordForm.jsp	Reset password (main JSP)
shipaddress.jsp	Shipping address
shipping.jsp	Shipping methods
sidebar.jsp	Web page side navigation bar

Understanding the web objects directory structure

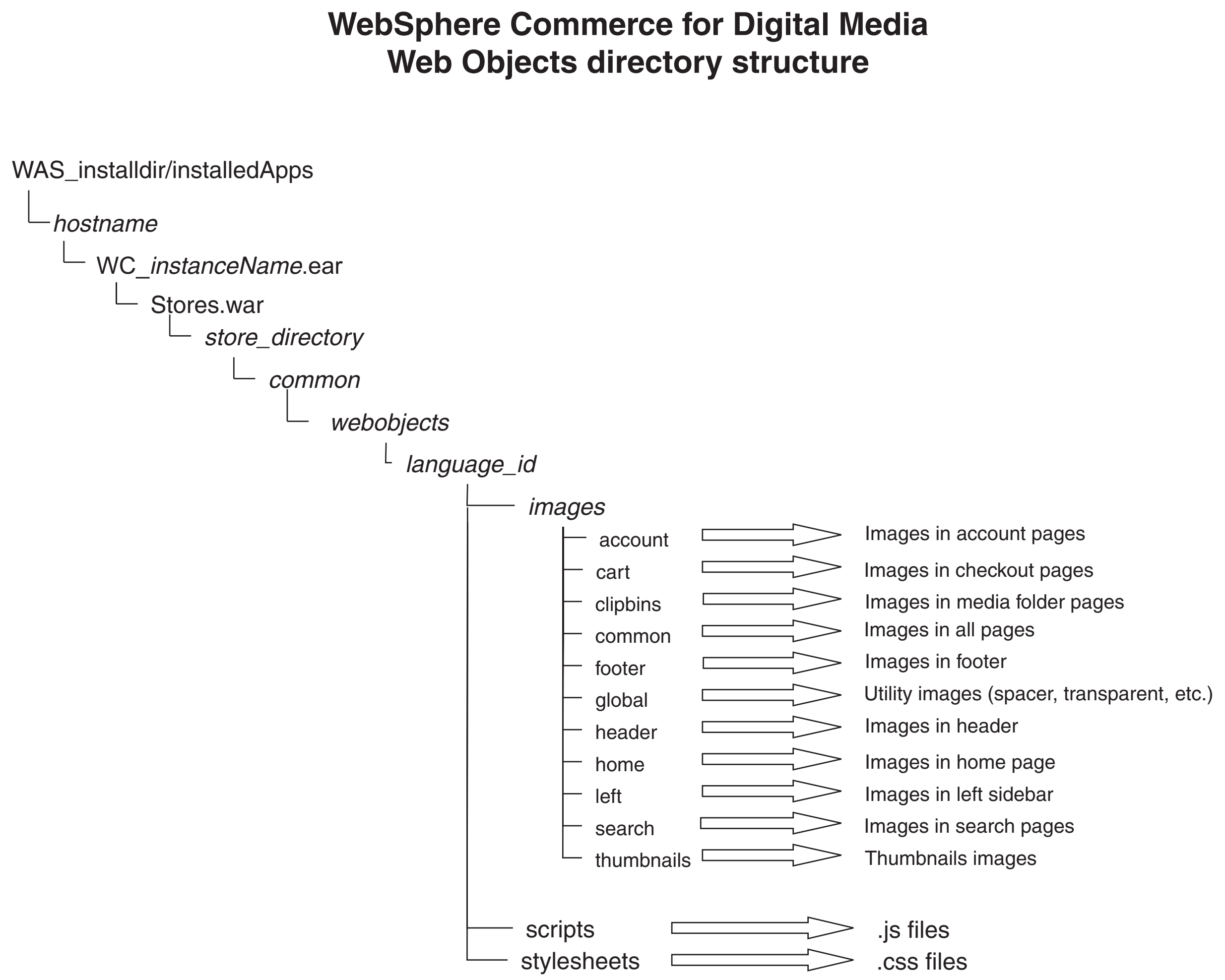

Figure 3. WebSphere Digital Media Enabler web objects directory

The graphics and design elements for the store are located under the `webobjects` directory:

WAS_installdir/installedApps/*hostname*/*WC_instance_name*.ear/
Stores.war/*store_directory*/common/webobjects/*language_id*

The directory structure shown below illustrates how the current WebSphere Digital Media Enabler JSP files are organized. If you change the structure, make sure you change the various WebSphere Digital Media Enabler properties files to reflect your modifications. Aside from the JSP files themselves, you need to check the following files where most path values and other variables are externalized:

- *WAS_installdir*/installedApps/*hostname*/WC_*instance_name*.ear/ Stores.war/*store_directory*/common/lightbox/jsp/DMConfig.inc
- *WAS_installdir*/installedApps/*hostname*/WC_*instance_name*.ear/ properties/DMproperties.properties
- *WAS_installdir*/installedApps/*hostname*/WC_*instance_name*.ear/ Stores.war/*store_directory*/delivery/delivery.properties
- *WAS_installdir*/installedApps/*hostname*/WC_*instance_name*.ear/ Stores.war/WEB-INF/classes/*store_directory*/dmstoretext_*locale*.properties
- *WAS_installdir*/installedApps/*hostname*/WC_*instance_name*.ear/ Stores.war/WEB-INF/classes/*store_directory*/dmstoreconfig_*locale*.properties
- *WAS_installdir*/installedApps/*hostname*/WC_*instance_name*.ear/ Stores.war/*store_directory*/delivery/delivery_*locale*.properties
- *WAS_installdir*/installedApps/*hostname*/WC_*instance_name*.ear/ Stores.war/*store_directory*/email/mailprocess.properties
- *WAS_installdir*/installedApps/*hostname*/WC_*instance_name*.ear/ Stores.war/*store_directory*/email/mailprocess_*locale*.properties
- *WAS_installdir*/installedApps/*hostname*/WC_*instance_name*.ear/ Stores.war/WEB-INF/classes/*store_directory*/dmstoreload.properties

Understanding the WebSphere Digital Media Enabler proxies directory structure

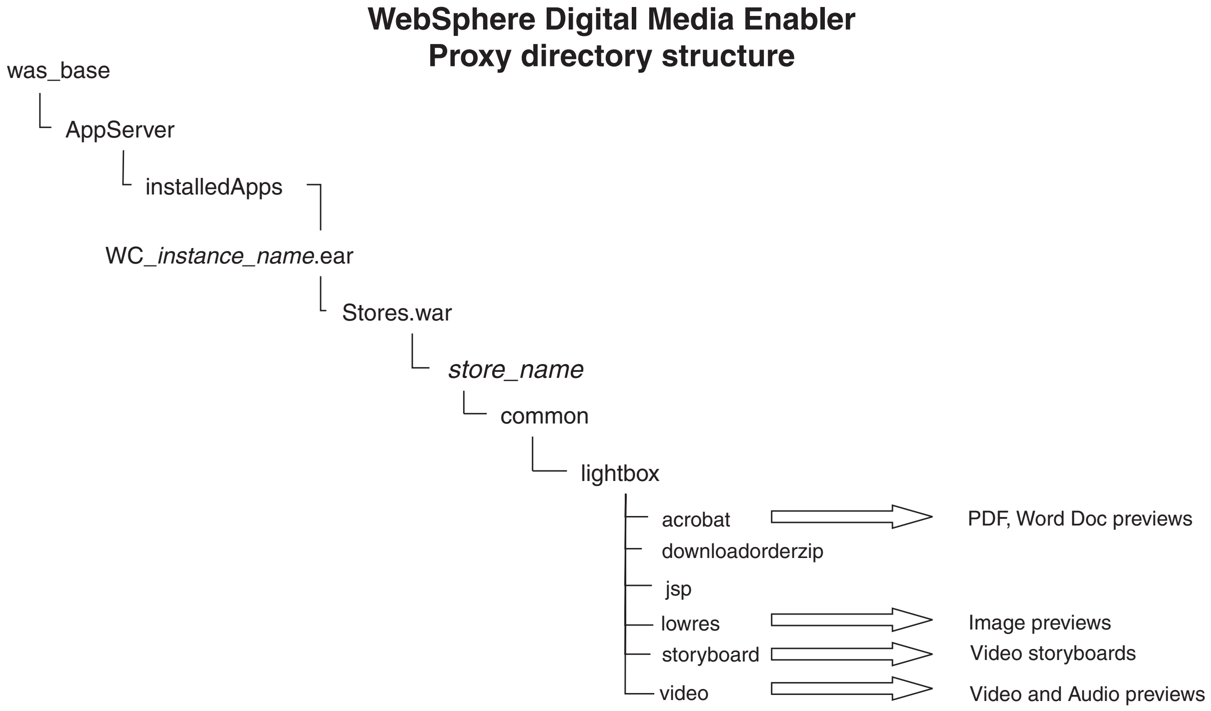

Figure 4. WebSphere Digital Media Enabler proxy directory structure

When loading content, the high-resolution and all other proxies are stored in Content Manager. In addition, thumbnails are stored on the file system for direct access by the HTTP server. The current WebSphere Digital Media Enabler code expects the proxies to be organized as shown in the previous directory structure illustration. If you change the structure, make sure you change the various WebSphere Digital Media Enabler properties files to reflect your modifications. Aside from the JSP files themselves, check the following files where most path values and other variables are externalized:

- `*WAS_installdir*/installedApps/*hostname*/*WC_instance_name*.ear /Stores.war/*store_directory*/common/lightbox/jsp/DMConfig.inc`
- `*WAS_installdir*/installedApps/*hostname*/WC_*instance_name*.ear/ properties/digitalmedia.properties`

Note that thumbnails are in the `webobjects` directory. Video proxies are streamed directly from Content Manager, so some may not be in the file system.

Chapter 4. Customizing the store front appearance

This chapter details how to change the appearance of the WebSphere Digital Media Enabler default store to customize your own store's look and feel. You can customize your store front appearance by changing the WebSphere Digital Media Enabler JSP files.

Header, sidebar, and footer Web page areas

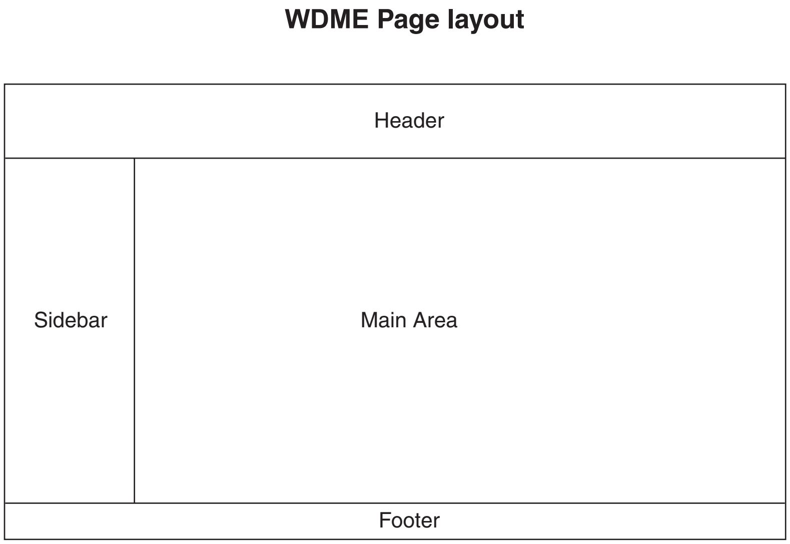

Figure 5. WebSphere Digital Media Enabler page layout

The diagram illustrates the general design framework for web pages throughout WebSphere Digital Media Enabler. Typically, the store administrator or designee provides the header, sidebar, and footer areas in the form of three separate HTML files as well as style sheets and all image graphics. The main area is where the WebSphere Digital Media Enabler functions manifest themselves.

Within an HTML table, incorporate your HTML code for your header, footer and sidebar and save them into the following files:

- *WAS_installdir*/installedApps/*hostname*/*WC_instance_name*.ear/ Stores.war/*store_directory*/common/lightbox/jsp/DMHeader.inc
- *WAS_installdir*/installedApps/*hostname*/*WC_instance_name*.ear/ Stores.war/*store_directory*/common/lightbox/jsp/DMSidebar.inc
- *WAS_installdir*/installedApps/*hostname*/*WC_instance_name*.ear/ Stores.war/*store_directory*/common/lightbox/jsp/DMFooter.inc

WebSphere Digital Media Enabler uses this overall layout by default as illustrated in the code snippet below:

```
<TABLE cellSpacing=0 cellPadding=0 width=600 border=0>
<TBODY>
<TR>
  <TD vAlign=top colSpan=3 height=42>
  <%
  //-----------------
  // HEADER STARTS HERE
  //-----------------
  %>
  <%@ include file="header.inc" %>
  </TD>
</TR>
<TR>
  <TD bgcolor=white align="left" valign="top">
    <%
    //--------------------------------
    // SIDEBAR STARTS HERE
    //--------------------------------
    %>
    <%@ include file="sidebar.inc" %>
  </TD>
<TD bgcolor="#FFFFFF" width="600" rowspan="6" valign="top">
    <%
    //---------------
    // MAIN CONTENTS HERE
    //---------------
    %>
    <%@ include file="????_main.inc" %>
  </TD>
</TR>
</TBODY>
</TABLE>
<%
//-----------------
// FOOTER STARTS HERE
//-----------------
%>
<%@ include file="footer.inc" %>
```

Main body page area

Following are the JSP files in the main area that you may need to modify.

Affected JSP files

As mentioned previously, one of the customization areas is in the data model that customers use to describe the digital content; therefore, the following search JSP files will need to change, at least to reflect the data field labels and possible values:

- Advanced Search: DMSearchEntry.jsp
- Search Results (display of metadata): DMSearchResult.jsp
- Item Details (display of metadata): DMSearchDetail.jsp
- Browse Categories: DMSearchBrowseCat.jsp

The next area of customization is in the user profile information to be captured on the Web. User registration/account updates are located in: register.jsp, edit_registration.jsp

The last area is in displaying the questions regarding usage for the purpose of pricing. Usage requests are located in: usagequestions.jsp, usagequestionsall.jsp

JSP and INC files

Although not recommended, you may change the pages in the main area. Most of the JSP files in the main area have visual components broken out into includable files (*.inc), so that you can insert them wherever your GUI design dictates.

You must design your JSP files and include the INC file wherever you want it. For example:

```
<table>
<tr><td> Pick from the following display options</td></tr>
<tr><td><jsp:include page="DMSearchDisplayOptions.inc"/></td></tr>
</table>
```

Following is a list of INC files that are included in many search JSP files and Media Folder JSP files within WebSphere Digital Media Enabler:

Table 8. Include files

INC files	Description
DMFolderViewPass.inc	Contains media folder parameters
DMConfig.inc	WebSphere Digital Media Enabler configuration file
DMFolderForms.inc	Common media folder forms
DMJSPFunc.inc	Common media folder Java functions
DMSearchAllFolderDropDown.inc	Search all folder drop downs
DMJSFunc.inc	Common media folder JavaScript functions
DMKeywordsSearch.inc	Common includes for Search (used in home.jsp)
DMHeader.inc	Web page header
DMFooter.inc	Web page footer
DMSearchForms.inc	Common forms for search function
DMSidebar.inc	Web page side navigation bar
DMSearchDisplayOptions.inc	Display options for image results
DMSearchFolderDropDown.inc	List of media folders for current user
DMSearchKeywords.inc	List of related keywords for current asset
DMSearchMetaData.inc	Display metadata for current asset
DMSearchPagesBar.inc	Lists the number of pages and links to next-previous
DMSearchPlayIt.inc	Process the preview display for current asset
DMSearchStoryBoard.inc	Shows the storyboard for a video asset
DMSearchMediaTypes.inc	Types of media supported and links to them
DMQuickTime.inc	Shows quicktime download
DMLibraryJSPFunc.inc	Common library JSP methods
DMLibraryForms.inc	Upload Library HTML forms
DMLibConfig.inc	JSP configuration variables
DMLibrary_View_Pass.inc	Passes form variables
DMSubscription.inc	Subscription variables
DMLibraryViewImageTable.inc	Display library view table
DMSameTime.inc	Sametime variables

Creating the customer-provided static pages

Each WebSphere Digital Media Enabler customer must provide some additional static HTML pages to include into the WebSphere Digital Media Enabler store. These pages have to be converted to JSP files and linked to a URL that allows for the e-commerce session to be kept alive. Examples of such static pages are Help, About Us, Privacy Policy, and Terms and Conditions.

Here are the steps to include these pages into the WebSphere Digital Media Enabler store:

- Open the HTML page provided by the customer and change the URL path to all images and graphics to contain the JSP variable:

  ```
  <%=WEBOBJECTS_RELATIVE_URL%>
  ```

  For example, the logo will be referred to in the HTML as:

  ```
  <img src="<%=WEBOBJECTS_RELATIVE_URL%>/images/common/ibmlogo2.jpg"
   border="0">
  ```
- Save the HTML as a JSP file in the directory:

  ```
  WAS_installdir/installedApps/hostname/WC_instance_name.ear
  /Stores.war/store_directory/common
  ```

  For example, `PrivacyPolicy.jsp`.
- Locate the `DMCommon.jsp` in the

  ```
  WAS_installdir/installedApps/hostname/WC_instance_name.ear
  /Stores.war/store_directory/common
  ```

  directory and modify it with the following logic inserted in the `IF` block:

  ```
  else if (includefile.equals("privacy"))         {
  %>
  <jsp:include page="PrivacyPolicy.jsp"/>
  <%
  }
  ```

  where `privacy` is a unique keyword you created to identify this link.
- Go to the JSP pages where the link to this static page is to be created. For the HREF of the link, be sure to insert this code:

  ```
  <a href="javascript:commonlinks('privacy')">Privacy Policy</a>
  ```

  Also, make sure you include the following JS file in your header: `links.js`. This JavaScript file contains the function commonlinks. This function will automatically call the proper JSP page with all the necessary WebSphere Commerce session parameters.

Chapter 5. Customizing the e-mail and store data setup

To review, areas of customization for e-mail and store data setup are:

- User registration data
- Pricing
- Delivery and e-mail
- Subscriptions
- Media folder collaboration

Customizing the user registration data

This customization is typical of a standard WebSphere Commerce store deployment. First, you will need to define the user profile information to be captured and map them to existing fields in the WebSphere Commerce user subsystem tables. Here is an example:

Field number on registration screen mockup	Screen name	Table	Column name	Column type
		ADDRESS	ADDRBOOK_ID	BIGINT NOT NULL
		ADDRESS	ADDRESS_ID	BIGINT NOT NULL
1	Address Line 1	ADDRESS	ADDRESS1	VARCHAR (50) NULL
2	Address Line 2	ADDRESS	ADDRESS2	VARCHAR (50) NULL
		ADDRESS	ADDRESS3	VARCHAR (50) NULL
		ADDRESS	ADDRESSTYPE	CHARACTER (5) NULL
		ADDRESS	BESTCALLINGTIME	CHARACTER (1) NULL
		ADDRESS	BILLINGCODE	VARCHAR (17) NULL
		ADDRESS	BILLINGCODETYPE	CHARACTER (2) NULL
3	Title	ADDRESS	BUSINESSTITLE	VARCHAR (128) NULL
4	City	ADDRESS	CITY	VARCHAR (128) NULL
5	Country	ADDRESS	COUNTRY	VARCHAR (128) NULL
6	E-mail	ADDRESS	EMAIL1	VARCHAR (256) NULL
		ADDRESS	EMAIL2	VARCHAR (256) NULL
7	Fax	ADDRESS	FAX1	VARCHAR (32) NULL
		ADDRESS	FAX2	VARCHAR (32) NULL
		ADDRESS	FIELD1	VARCHAR (64) NULL
		ADDRESS	FIELD2	VARCHAR (64) NULL
		ADDRESS	FIELD3	VARCHAR (64) NULL
8	First Name	ADDRESS	FIRSTNAME	VARCHAR (128) NULL
		ADDRESS	ISPRIMARY	INTEGER NULL
		ADDRESS	LASTCREATE	TIMESTAMP NULL
9	Last Name	ADDRESS	LASTNAME	VARCHAR (128) NULL

Field number on registration screen mockup	**Screen name**	**Table**	**Column name**	**Column type**
		ADDRESS	MEMBER_ID	BIGINT NOT NULL
10	Middle Name	ADDRESS	MIDDLENAME	VARCHAR (128) NULL
		ADDRESS	NICKNAME	VARCHAR (254) NOT NULL
		ADDRESS	OFFICEADDRESS	VARCHAR (128) NULL
		ADDRESS	ORGNAME	VARCHAR (128) NULL
		ADDRESS	ORGUNITNAME	VARCHAR (128) NULL
		ADDRESS	PACKAGESUPPRESSION	INTEGER NULL
11	Prefix	ADDRESS	PERSONTITLE	VARCHAR (50) NULL
12	Phone	ADDRESS	PHONE1	VARCHAR (32) NULL
		ADDRESS	PHONE1TYPE	CHARACTER (3) NULL
		ADDRESS	PHONE2	VARCHAR (32) NULL
		ADDRESS	PHONE2TYPE	CHARACTER (3) NULL
		ADDRESS	PUBLISHPHONE1	INTEGER NULL
		ADDRESS	PUBLISHPHONE2	INTEGER NULL
		ADDRESS	SELFADDRESS	INTEGER NOT NULL, DEFAULT 0
		ADDRESS	SHIPPINGGEOCODE	VARCHAR (254) NULL
13	State/Province	ADDRESS	STATE	VARCHAR (128) NULL
		ADDRESS	STATUS	CHARACTER (1) NULL
		ADDRESS	TAXGEOCODE	VARCHAR (254) NULL
14	Postal Code	ADDRESS	ZIPCODE	VARCHAR (40) NULL
15	Hint for Password	USERREG	CHALLENGEANSWER	VARCHAR (254) NULL
		USERREG	CHALLENGEQUESTION	VARCHAR (254) NULL
16	User ID	USERREG	LOGONID	VARCHAR (254) NOT NULL
17	Password	USERREG	LOGONPASSWORD	CHARACTER (128) NULL
		USERREG	PASSWORDEXPIRED	INTEGER NULL
		USERREG	STATUS	INTEGER NULL
		USERREG	USERS_ID	BIGINT NOT NULL
18	Company	USERDEMO	COMPANY_NAME	VARCHAR (128) NULL
19	Company Type	USERDEMO	FIELD7	VARCHAR (64) (List in IBM Standard field OK for start)
20	Company Type Other	USERDEMO	FIELD7	VARCHAR (64) NULL

Field number on registration screen mockup	Screen name	Table	Column name	Column type
21	Industry Sector	USERS	FIELD1	Corporate Education Media/ Entertainment Publishing Other
22	Industry Sector Other	USERS	FIELD1	VARCHAR (64) NULL
23	What delivery format do you prefer?	USERDEMO	MARITALSTATUS	*CHAR*(1) Download digital file CD Film
24	What web connection speed do you have?	USERDEMO	INCOME	INTEGER 56k modem DSL/Cable Modem T1 or greater
25	What type of computer do you use?	USERDEMO	FIELD1	CHAR(1) Apple Macintosh Windows PC
26	How did you hear about us?	USERDEMO	AGE	INTEGER Advertisement Colleague Direct Mail Trade Show
27	How often do you work on projects requiring stock footage?	USERDEMO	GENDER	CHAR(1) Daily Weekly Monthly A few times a year
28	What categories of stock images are you most interested in?	USERDEMO	FIELD5	VARCHAR(254) (will match top level categories once defined)
29	Are there other features you would like to see on our u?	USERDEMO	HOBBIES	VARCHAR(254) NULL

Field number on registration screen mockup	Screen name	Table	Column name	Column type

30	Please send me information regarding new collections.	USERDEMO	FIELD2	CHAR(1)
				Y=Checked
				Check / Uncheck

31	Please do not send me HTML version of the IBM company newsletter.	USERDEMO	FIELD3	CHAR(1)
				Y=Checked
				Check / Uncheck

32	I agree to the Terms and Conditions of Use.	USERDEMO	FIELD4	CHAR(1)
				Y=Checked
				Check / Uncheck

33	XYZ Corporate Account	USERDEMO	FIELD6	Integer 1=Corporate Account
34	Subscription Type	USERPROF	Field1	P/D/A
35	Subscription Value	USERPROF	Field2	
36	Subscription	ORDER	Field3	NEWSUBSCRIPTION |(P/D/A) RENEWSUBSCRIPTION |(P/D/A) SUBSCRIPTION |(P/D/A)
37	Autoconnect to Sametime at logon	BUSPROF	MANAGER	VARCHAR (256) 1=Checked Check/Uncheck

Once you have mapped the fields and defined the field labels to be displayed on the user registration form, you will need to change the following JSP files to capture the information:

- `WAS_installdir/installedApps/hostname/WC_instance_name.ear/Stores.war/store_directory/common/edit_registration.jsp`
- `WAS_installdir/installedApps/hostname/WC_instance_name.ear/Stores.war/store_directory/register.jsp`

Customizing the usage pricing data

This section describes how to customize a store's usage pricing. It describes the methodology and design considerations as well as detailing the steps to implement the desired algorithm.

Pricing description

Prices are calculated based upon a point system where the answer to each question is associated with a weighted percentage value.

The percentage associated for each answer will vary based upon the selection; for example, the percentage assigned to the market question may be 100% for local, 150% for regional, or 200% for national. A percentage value of 100% means that the answer applies, but does not change the price.

After all the questions are answered, follow these steps to assign a price to each order item.

1. Take the base price for the product (for example, 1.00 or 1.05).
2. Multiply the product base price by the base price for the usage category (from the USAGEDEF table based on the answer to question 0), giving the base price for this usage.
3. For each answer in the usage table, multiply the current usage price by the percentage in the USAGEPERCENT table, as determined by looking up the value of the answer for the specific usage and question number.
4. Multiply the result based upon the percentage associated with the number of answers for the pick list.

Example calculation:

Assume:

- Base price for the image (product) = 1.00
- Base price for the usage = 1000.00
- Percentage for answer to question1 = 150% (multiplier f 1.50)1.50
- Percentage for answer to question1= 150% (multiplier f 1.50)1.50
- Percentage for the number of pick list items = 200% (multiplier of 2.00)

Resulting price is:

1.00 x 1000 x 1.50 x 1.50 x 2.00 = $4500

Database tables for usage pricing

Five tables are used in the pricing mechanism.

1. USAGEDEF table

   Each row contains the usage number, usage text, and usage base price. The status 'A' indicates active and 'I' indicates inactive. Examples are:

Table 9. WCDMUSAGEDEF

Usage ID	Language_id	UsageText	Status	Base Price
01	–1	Use 001	A	1000
02	–1	Use 002	A	1500
03	–1	Use 003	A	1000
43	–1	Use 043	A	500

   Reference the LANGUAGE_ID column of the LANGUAGE table from WebSphere Commerce for other Language_ids.

2. USAGEQUESTIONS table

   Each row contains the question id and question text. The question is the same as a "factor". The status 'A' indicates active and 'I' indicates inactive. The

QuestionType field is used to indicate the type of question display that should be used such as **S** for Select Box and **C** for Checkbox fields (used for pick list) Examples are:

Table 10. WCDMUSAGEQUESTIONS

Question ID	Language_id	Question Text	Status	QuestionType
Factor 1	–1		A	S
Factor 2	–1		A	S
Factor 3	–1		A	S
Factor 4	–1		A	S
Factor 5	–1		A	S
Factor 6	–1		A	S
10 Pick List 1	–1		A	C

Questions 1-n are factors. Questions for picklists can have multiple answers.

3. USAGEANSWERS table

   The usage answers table contains each answer for each question and the number of points assigned to each answer. Since the point value also depends upon the usage category, the answer values are not taken from this table, but instead from the USAGEPERCENT table.

   Since the value assigned to the answers to the pick list questions is based upon the number of answers and not the answers themselves, the answers are defined in a special manner. In this case, for ease of calculation, answer IDs are assigned with values equal to the question ID x 10 added to the number of items. Thus, if the question ID is 8, the answer IDs for question 8 are 81, 82, 83, 84, 85, and 86. If three answers are chosen, the multiplier value is based upon answer 83.

   Examples are:

Table 11. WCDMUSAGEANSWERS

AnswerID	Language_id	Question ID	AnswerText	Points
1	–1	1	Multiplier1	n/a
2	–1	1	Multiplier2	n/a
3	–1	1	Multiplier3	n/a
4	–1	2	Multiplier4	n/a
5	–1	2	Multiplier5	n/a
8	–1	8	Multiplier6	n/a
82	–1	8	Multiplier7	n/a
83	–1	8	Multiplier8	n/a

4. USAGE table

   The usage table information is created when the order is placed. It contains all of the usage information for the specific customer request. Examples are:

Table 12. WCDMUSAGE

Order ID	ORDITEM ID	Usage ID	Question id	Answer id	AnswerText
000001	00001	1	1	1	n/a
000001	00001	1	2	1	n/a

Table 12. WCDMUSAGE (continued)

Order ID	ORDITEM ID	Usage ID	Question id	Answer id	AnswerText
000001	000001	1	3	3	n/a

5. USAGEPERCENT table

   The USAGEANSWERS table contains an answer for each question/usage combination and the number of points assigned to each answer. This table contains rows of questions/answers combinations that apply to the individual usage. The point values represent the percentage multiplier for each answer. Examples are:

Table 13. WCDMUSAGEPERCENT

Usage ID	**AnswerId**	**Question id**	**Points**
01	1	1	1.00
01	2	1	1.50
01	3	1	2.00
01	4	2	1.00
01	5	2	1.25

With the database client GUI such as DB2 Command Center, you can create rows into the USAGEDEF, USAGEQUESTIONS, and USAGEANSWERS tables.

Contract pricing

To set up contract pricing, you must create a contract and set approval for user registration. The following steps illustrate how to add contract pricing to your WebSphere Digital Media Enabler customization.

WebSphere Commerce setup for contract pricing

Follow the steps below to create a contract.

1. Create an Organization (for example, Buyer Organization A): **WebSphere Commerce Organization Administration Console: Access Management > Organizations > New**
2. Create an organization owner by creating a new User: **WebSphere Commerce Organization Administration Console: Access Management > Users > New**
3. Create an Account and associate it with the organization and owner created in the previous steps: **WebSphere Commerce Accelerator: Sales > Accounts > New**
4. Create a Contract for the account just created: **WebSphere Commerce Accelerator: Sales > Accounts > Contracts > New**. Fill in the information for the **General**, **Customers**, **Products and Prices**, and **Shipping Payment** tabs.
5. Submit the Contract for deployment: **WebSphere Commerce Accelerator: Sales > Accounts > Contracts > Submit**

Follow the steps below to set approval for user registration with an organization:

1. Add an Approval (for example, User Registration) to an organization: **WebSphere Commerce Administration Console: Access Management > Organizations**
2. Add an approval Member Group to user: **WebSphere Commerce Organization Administration Console: Access Management > Users > Member Groups**

Process for contract pricing

Assuming that a contract has been created and it has been set up to require approval for users registering with an organization, the following takes place:

1. A user registers with an organization.

2. The registration approver is notified by e-mail for approval of the new user registration.
3. When the user attempts to log in, a message is displayed informing the user that the registration is pending approval and that access is limited.
4. The approver approves or rejects the user registrations using WebSphere Commerce Accelerator or the WebSphere Commerce Administration Console.
5. The approver must manually add the new user to a role in the root organization.
6. The user can log in if the registration is approved.
7. After the user logs in and attempts to buy a media object, the price is recalculated according to the contract.
8. The user's organization is displayed on the account (edit registration) page.

Delivery and e-mail

Configuring the delivery daemon

The current delivery method supported by WebSphere Digital Media Enabler is an HTTP download of the ordered assets. You will need to customize the parameters needed by that delivery module based on your customer environment.

There are two properties files that need to be changed:

- `WAS_installdir/installedApps/`*`hostname`*`/`*`WC_instance_name`*`.ear/Stores.war/`*`store_directory`*`/delivery/delivery.properties`
- `WAS_installdir/installedApps/`*`hostname`*`/`*`WC_instance_name`*`.ear/Stores.war/`*`store_directory`*`/delivery/delivery_`*`locale`*`.properties`

Most of the pertinent delivery parameters are in `delivery.properties`. Here is an example:

```
################## Delivery Parameters ####################

#-------------------------------------------------------------------------------------------#
#
# Change the below for Database properties      #

#-------------------------------------------------------------------------------------------#
#

# Database URL. Same as WebSphere Commerce Database URL
# DB2 example = jdbc:oracle:thin:@wcora:1521:orcl
# ORACLE example = jdbc:db2:MALL
WCDM.DATABASEURL=jdbc:db2:MALL
# Database username and password must be same as WebSphere Commerce
# Database Schema username and password.
WCDM.DATABASEUSER=vsadmin
WCDM.DATABASEPASSWORD=vsadmin
# Database Driver
# DB2 example = COM.ibm.db2.jdbc.app.DB2Driver
# ORACLE example = oracle.jdbc.driver.OracleDriver

WCDM.DATABASEDRIVER=COM.ibm.db2.jdbc.app.DB2Driver

# Look in the member/orgentity table and find out the member id for the store.
WCDM.STOREMEMBERID=-2000r
#----------------------------------------------------------------------------
#
# Change the below for Locale properties file
#
# The locale specific parameters are read from locale specific properties file#
#------------------------------------------------------------------------------
#

WCDM.LOCALE=en_US
WCDM.LANGUAGE=-1

#--- Store ID
DM.STORE_ID=10051
#------------------------------------------------------------------------------#
# Delivery System Parameters
#
# Make sure they point to the right directory
```

```
#
# Change the below for Action File and daemon sleep time Parameters
#
#-----------------------------------------------------------------#

# Action File. Change this to No if you want to stop the Delivery Daemon
WCDM.ACTIONFILE=D:\\WEBSPH~1\\APPSER~1\\installedApps\\WC_Enterprise_App_demo.ear\\wcstores.war\\
DigitalMedia\\delivery\\DeliveryAction.txt

# Daemon Sleep Time in minutes
WCDM.DAEMONSLEEPTIME=1

#-------------------------------------------------------------------------#

# Email Attachment Parameters
#
# Either HTML or PDF can be created as an attachment
#
# The Attachment is created in the delivery daemon and send with the order
# email in email daemon
#
#----------------------------------------------------------------------------#

# Create the html Attachment File
WCDM.CREATEHTML=YES

# Create the PDF File from the HTML File
WCDM.CREATEPDF=NO

# Command to convert html to pdf. Needed if PDF is attached.
WCDM.CONVERTHTMLTOPDF=htmldoc --webpage -f

# Path to store the html and pdf.
WCDM.ATTACHMENTFILEPATH=D:\\WEBSPH~1\\APPSER~1\\installedApps\\WC_Enterprise_App_demo.ear\\wcstores.war
\\DigitalMedia\\delivery\\orders

#-------------------------------------------------------------------------------------#
# Download order Parameters
#
#-------------------------------------------------------------------------------------#

# Zip dir is the directory where one wants to store the delivery zip
# A directory is created inside the zipdir. i.e. <storeid><userid><orderno>
WCDM.ZIPDIR=D:\\WEBSPH~1\\APPSER~1\\installedApps\\WC_Enterprise_App_demo.ear\\wcstores.war\\
DigitalMedia\\delivery\\orders/

# The delivery daemon will zip the media files using the JAVA API
# If one wants to use the WINZIP Command then change this to NO and provide the wzzip path in WCDM.ZIPCMD
WCDM.MEDIAZIPJAVAAPI=NO

# Command to zip the files. The Licensed WinZip Command Line Utility must be installed.
WCDM.ZIPCMD=C:/progra~1/winzip/wzzip -rp -u -x*.zip

#-----------------------------------------------------------------------------------------#
# Default Delivery Parameters
#
# Should be changed only for customization
#
#-----------------------------------------------------------------------------------------#

# Number of days the download order should be kept.
# If the number is -1 then the files are never deleted
WCDM.DOWNLOADEDORDERDELETEDAYS=-1

# After creating zip files delete the images.
WCDM.DELETEIMAGES=YES

#Zip Extension
WCDM.ZIPEXT=zip

# There can be 2 files generated, one is orderno.zip and the second is the zip for each orderitem.
# If one needs only the complete order zip and not individual item zip say YES else NO.
WCDM.ONLYORDERZIP=YES

# Which Orders Not to be Downloaded.
# If the Delivery Method is download then don't zip the order
# If one wants to zip all orders irrespective of delivery method then change this to XXXXXXXX
WCDM.REJECTORDERS=DOWNLOAD

# Want to download only High Res
WCDM.ONLYHIGHRES=YES

# File Extension for the thumbnails
WCDM.THUMBEXT=jpg

### CD's which should not be downloaded using the Delivery System
### The Catentry MFPartNumber starts with the  SKU-

# CD MFPartNumber
# The Item MF PartNumber staring with SKU- will not be considered for download as it
 indicates that it is a CD.
# If no Such check required change it to XXXX
```

```
WCDM.CDMFPARTNUMBER=SKU-

# Image Search Prefix. To Search Content Manager on the image number or to append the prefix
to image number.
# If one wants to search just on the imageNumber then make the value as blank
# If one wants to search on imageNumber-254363 where 254363 is the actual image no
# then add a Prefix For ex. WCDM.SEARCHPREFIX=imageNumber-
WCDM.SEARCHPREFIX=

#-------------------------------------------------------------------------------------#
# Do not change the below Parameters              #

# Order Attachment Create Field2 Value
WCDM.ORDERATTACHMENTCREATED=0.00000

# Order Approved Field2 Value
WCDM.ORDERAPPROVED=1.00000


################# Table Names ###################

WCDM.USAGETABLE=WCDMUSAGE
WCDM.USAGEDEFTABLE=WCDMUSAGEDEF
WCDM.USAGEANSTABLE=WCDMUSAGEANSWERS
WCDM.USAGEQUETABLE=WCDMUSAGEQUESTIONS

# WCS Users table
WCDM.USERSTABLE=USERS

# WCS Users Demographic table
WCDM.USERSDEMOTABLE=USERDEMO

# WCS Address table
WCDM.ADDRESSTABLE=ADDRESS

# WCS product table
WCDM.CATENTRYTABLE=CATENTRY

# WCS product description table
WCDM.CATENTDESCTABLE=CATENTDESC

# WCS Orders table
WCDM.ORDERSTABLE=ORDERS

# WCS Orders table
WCDM.ORDERITEMSTABLE=ORDERITEMS

# WCS Shipping Mode table
WCDM.SHIPMODETABLE=SHIPMODE

# WCS Shipping Mode Description table
WCDM.SHIPMODEDSCTABLE=SHPMODEDSC

# WCS Order Comment table
WCDM.ORDERCOMMENTTABLE=ORCOMMENT

# WCDM Email table
WCDM.EMAILTABLE=WCDMEMAIL

# WCDM User Catalog View
WCDM.USERCATALOGVIEW=WCDMUSERCATALOGVIEW

################# End Delivery Parameters ###################
```

In the `delivery_locale.properties` file, you change the locale-specific properties.

```
################# Delivery Locale Specific Parameters ###################

# Terms and Conditions File
WCDM.TERMSFILE=D:/WebSphere/WAS_installdir/installedApps/hostname/WC_Enterprise_App_demo.ear/
Stores.war/DigitalMedia/delivery/terms_en_US.htm

# Path where the thumbnails are located.
# Attaching the thumbnail to the html/pdf document
# Change the path to your existing system.
WCDM.SERVERTHUMBPATH=http://tmws39/webapp/wcs/stores/DigitalMedia/en_US/webobjects
/images/thumbnails/

################# Client Parameters ###################

# Client Name
WCDM.CLIENTNAME=IBM

# Client Phone Number
WCDM.CLIENTPHONENO=1-800-IBM-4YOU

################# Text Parameters ###################

WCDM.MAILADDRESS=IBM<br>Department 6710A<br>Bethesda, MD 20850
WCDM.RETURNADDRESS=IBM<br>6710 A Rockledge Drive<br>Bethesda, MD 20850
<br><b>ATTN: IBM/DigitalMedia</b>

WCDM.FEDTAXID=12-345-6789

WCDM.TEXT1=The release of publication/use rights documented below is contingent on payment
of the full balance due to the accompanying invoice,

WCDM.TEXT2=<b>PLEASE REFER TO THE TERMS AND CONDITIONS AT THE END OF THIS
OF THIS RIGHTS RELEASE DOCUMENT</b>.

WCDM.TEXT3=The <b>Balance Due</b> listed below is the <b>NET</b> amount due. Any taxes, tariffs, bank charges,
 wire transfer fees, etc. are your responsibility.

WCDM.TEXT4=Please pay the <b>Balance Due</b> amount in <b>U.S. dollars</b> drawn from
a <b>U.S. bank</b>.

WCDM.TEXT5=<center><b>**</b> Please make your check payable to the <b>IBM</b>.</center>

WCDM.TEXT6=<center><b>**</b> Include the <b>IBM Job Number</b> with any <b>payment or return of
materials</b> to insure that your account will be properly credited.</center>

WCDM.TEXT8=DIGITALMEDIA - SALES

WCDM.TEXT9=On {0} reproduction rights to the following pictures were released as detailed below:

WCDM.TEXT10=On {0} the following activity occured on your account:

WCDM.RIGHTSRELEASE=Rights Release
WCDM.JOBNUMBER=Job Number:
WCDM.PHONE=Phone:
WCDM.CONSIGNEE=Consignee:
WCDM.REQUESTDATE=Request Date:
WCDM.COMPANY=Company:
WCDM.ADDRESS=Address:
WCDM.INVOICE=Invoice
WCDM.CLIENT=Client:
WCDM.BALANCEDDUE=Balance Due: $
WCDM.MAILPAYMENT=<U>Mail <b>PAYMENT ONLY</b> to:</U>
WCDM.RETURNMATERIALS=<U>Return <b>MATERIALS</b> to:</U><br>(including <b>ALL</b> duplicates)
WCDM.FEDERALTAXID=Federal Tax ID:
WCDM.ACCOUNTACTIVITY=Account activity for this job detailed on following pages(s)
WCDM.TRANSACTIONTYPE=Transaction Type
WCDM.IMAGEID=Image ID
WCDM.AMOUNT=Amount

################# Transaction types ###################

WCDM.SHIPPINGTT=Shipping Cost

WCDM.ITEMTT=repro fee

WCDM.TAXTT=Tax Cost

WCDM.ITEMTTDESC=This charge is for publication/use rights which are defined on
'Rights Release' document

WCDM.TAXTTDESC=
```

There will be a delivery daemon module that will perform fulfillment in batch mode, typically used to deliver to FTP sites and folders for CD burning. The customization for this module is to be determined.

Customizing the e-mail daemon and e-mail templates

This section describes how to customize your store’s e-mail functions.

E-mail will be sent under the following situations:

- The order is marked as shipped.
- The password is reset when the user indicates that he has forgotten his password.
- The user places a new order.
- The user shares a Media Folder with another user.
- The user sends a Media Folder to a user e-mail address.
- A new user is registered.

For credit card customers, it is considered a new order when the credit card is approved and the order is authorized and marked as **Completed**. This happens during checkout using a payment provider link.

A shipment notification e-mail is sent by WebSphere Commerce when an order is marked as shipped using the WebSphere Commerce Accelerator.

A password reset e-mail is sent by WebSphere Commerce when the user requests a new password.

The other e-mail messages are sent by the e-mail daemon provided with WebSphere Digital Media Enabler. This daemon monitors the e-mail table for new messages and then creates and sends e-mails based upon the sender, receiver, and message contents.

The areas of customization are as follows:

1. The configuration parameters for the e-mails are customized in the following file:

   WAS_installdir/installedApps/*hostname*/*WC_instance_Name*.ear/
   Stores.war/*store_directory*/email/mailprocess.properties
2. The message body of the e-mails for the orders is customized in the following file.

   WAS_installdir/installedApps/*hostname*/*WC_instance_name*.ear/
   Stores.war/*store_directory*/email/mailprocess_*locale*.properties
3. The message body for e-mailing media folders is customized in the following file.

   WAS_installdir/installedApps/*hostname*/*WC_instance_name*.ear/
   Stores.war/WEB-INF/classes/*store_directory*/dmstoretext_*locale*.properties

Here is an example of `mailprocess.properties`.

```
#----------------------------------------------------------------------------------#
#
# Change the below for Database properties

#----------------------------------------------------------------------------------#
#

# Database URL. Same as WebSphere Commerce Database URL
# DB2 example = jdbc:oracle:thin:@wcora:1521:orcl
# ORACLE example = jdbc:db2:MALL
WCDM.DATABASEURL=jdbc:db2:MALL

# Database username and password must be same as WebSphere Commerce
# Database Schema username and password
WCDM.DATABASEUSER=wcs
WCDM.DATABASEPASSWORD=wcs

# Database Driver
# DB2 example = COM.ibm.db2.jdbc.app.DB2Driver
# ORACLE example = oracle.jdbc.driver.OracleDriverWCDM.DATABASEDRIVER=COM.ibm.db2.jdbc.app.DB2Driver

#-------------------------------------------------------------------------------#
#
# Change the below for Locale properties file
#
# The Email subject,messages and the from email address is read from locale
#
# specific properties file.
#
#--------------------------------------------------------------------------------#

WCDM.LOCALE=en_US

#--------------------------------------------------------------------------------#

# Change the below for Email Server Parameters
#
#--------------------------------------------------------------------------------#
```

```
# SMTP host name
WCDM.SMTPHOST=mps07.ibmus2.ibm.com

#------------------------------------------------------------------------------#

# Change the below for Email Parameters
#
# Change to NO if you don't want to send the email.
#
#------------------------------------------------------------------------------#

# User Registration Message should be sent
WCDM.EMAILUSER=YES

# New Order Email should be sent
WCDM.EMAILNEWORDER=YES

# Folder Share Notification and Emailing Folder
WCDM.EMAILFOLDER=YES

# Order Shipped Email should be sent
WCDM.EMAILSHIPPEDORDER=YES

# New & Renew Subscription Email should be sent
WCDM.EMAILSUBSCRIPTION=YES

# Cancel Subscription Email should be sent
WCDM.EMAILCANCELSUBSCRIPTION=YES
#------------------------------------------------------------------------------#

# Change the below for Email Attachment Parameters
#
# Either HTML or PDF can be sent as an attachment
     #
# The Attachment is already created in the delivery daemon
#
#------------------------------------------------------------------------------#

# Attach html File to the new order email
WCDM.ATTACHHTML=YES

# Attach pdf File to the new order email
WCDM.ATTACHPDF=NO

# Path where the attachment html or pdf are stored.
# This is the same path where the delivery daemon created the attachment
# The attachment is sent with the order email.
WCDM.FILEATTACHMENTPATH=D:/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/wcstores.war/
DigitalMedia/delivery/orders//

#------------------------------------------------------------------------------#

# Change the below for Action File and daemon sleep time Parameters

#------------------------------------------------------------------------------#

# Action File. Change this File to ACTION=N0 if you want to stop the Email Daemon
# during the runtime.
WCDM.ACTIONFILE=D:/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/wcstores.war/
DigitalMedia/email/EmailAction.txt

# Daemon Sleep Time in minutes
WCDM.DAEMONSLEEPTIME=1

#------------------------------------------------------------------------------#
# Do not change the below Parameters
#
#------------------------------------------------------------------------------#


############### Order Table Status #######################

# Order Attachment Create Field2 Value
WCDM.ORDERATTACHMENTCREATED=0.00000

# Order Approved Field2 Value
WCDM.ORDERAPPROVED=1.00000

# Order Shipped Field2 Value
WCDM.ORDERSHIPPED=2.00000

################## Table Names ####################

# WCS Users table
WCDM.USERSTABLE=USERS

# WCS Orders table
WCDM.ORDERSTABLE=ORDERS
```

```
# WCS Order Comment table
WCDM.ORDERCOMMENTTABLE=ORCOMMENT

# WCDM Email table
WCDM.EMAILTABLE=WCDMEMAIL

# WCDM User Catalog View
WCDM.USERCATALOGVIEW=WCDMUSERCATALOGVIEW
```

Here is an example of `mailprocess_en_US.properties`.

```
################# Locale Specific Email Parameters ###################

# From Email address for User Registration
WCDM.USERREGFROMMAIL=mkhilnan@us.ibm.com

# From Email address for New Order
WCDM.NEWORDERFROMEMAIL=mkhilnan@us.ibm.com

# From Email address for Shipped Order
WCDM.SHIPPEDORDERFROMEMAIL=mkhilnan@us.ibm.com

# From Email address for Subscription emails
WCDM.SUBSCRIPTIONFROMEMAIL=mkhilnan@us.ibm.com

# Admin Email address receiving the cancel subscription emails
WCDM.SUBSCRIPTIONCANCELADMINEMAIL=mkhilnan@us.ibm.com

# Subject for User Registration Email
WCDM.USERREGSUBJECT=WebSphere Digital Media Enabler Registration Confirmation

# Subject for New Order Email
WCDM.NEWORDERSUBJECT=WebSphere Digital Media Enabler New Order Confirmation

# Subject for Shipped Order Email
WCDM.SHIPPEDORDERSUBJECT=WebSphere Digital Media Enabler Order Shipped Confirmation


##################### Message Text #############################

# General message sent when a user registers
WCDM.USERREGISTERMESSAGETEXT = Thank you for registering with WebSphere Digital Media Enabler.
\n\nYou can now login to search, preview and order footage from our extensive\ncollection.
Since we will be updating our store frequently with new footage,\nwe hope that you will visit
us regularly to check out our new images and give\nus feedback that will help us meet your
creative needs.\n\nIf you have any questions about your account or feedback on www.ibm.com,
\nplease contact us immediately at our toll-free customer service number\n1-800-IBM-4YOU,
or via email at info@ibm.com

# General message sent when a new order is approved for a user - {0}=OrderNumber
WCDM.NEWORDERMESSAGETEXT = Thank you for your order.\nYour order number is {0}.
\n\nIf your order was download only then go to your Order History page at
\nhttp://www.ibm.com/ and click on the Order Number for this order and download your
clips.\n\nIf your order has to be delivered, we will send a shipping confirmation
email with a tracking \nnumber once they have been shipped.\n\nRights Release
Document is attached with this e-mail.\n\nThank you for choosing WebSphere
Digital Media Enabler Site as your Footage provider.\n

# Ship order message - {0}=OrderNumber {1}=FedExNumber
WCDM.SHIPPEDORDERMESSAGETEXT = Thank you for your order.\nYour order number {0} has
been shipped by Federal Express.\nYour Shipment tracking n
umber is: {1} \n\nAll packages can be tracked on the Federal Express website at
\n\nhttp://www.fedex.com/us/tracking/. \n\nThank you for choosing WebSphere
Commerce DigitalMedia store as your Footage provider.
```

Here is an example of `dmstoretext_en_US.properties` mail parameters.

```
#=================================================================
#DMFolderEmail.jsp
#=================================================================
DMFOLDEREMAIL_TITLE = IBM: Email Media Folder
DMFOLDEREMAIL_JS1 = Incorrect e-mail address.
DMFOLDEREMAIL_JS2 = Please specify one or more e-mail addresses to send
a URL of your Media Folder to.
DMFOLDEREMAIL_JS3 = Would you like to proceed with sending this email?
DMFOLDEREMAIL_TEXT1 = Media Folder -
DMFOLDEREMAIL_TEXT2 = A standard message with a URL for viewing the contents
of the Media Folder
DMFOLDEREMAIL_TEXT3 = will be sent with this email. You can specify a
personalized message to be included in this email by typing it into the Message Field
below. For more help using Media Folders, click on
DMFOLDEREMAIL_TEXT4 = To:
DMFOLDEREMAIL_TEXT5 = (separate addresses with a comma)
```

```
DMFOLDEREMAIL_TEXT6 = From:
DMFOLDEREMAIL_TEXT7 = Subject:
DMFOLDEREMAIL_TEXT8 = Message
DMFOLDEREMAIL_TEXT9 = (optional)
DMFOLDEREMAIL_TEXT10 = Send
DMFOLDEREMAIL_TEXT11 = Clear
#In emailmessage {0} = First Name, {1} = Last Name
DMFOLDEREMAIL_EMAILMESSAGE = A media folder has been sent to you by [{0}].
You can view this media folder by clicking on the following URL:\n\n{1}\n\n\n ---------------------------\n\n
```

Subscriptions

This section describes the setup for the three subscription models available for customization.

Setting up the system for subscription models

Some stores allow customers to subscribe for content usage, rather than to pay for each offering individually. WebSphere Digital Media Enabler supports three types of subscriptions:

- **Duration**: The user can buy the subscription for a particular duration. During this period all the purchased images are part of that subscription. For example, one can subscribe for 1 month, 3 months, 1 year, or 3 years.
- **Amount**: The user buys the subscription for a particular amount. As the user buys images, the price of the images is deducted from the subscription amount. For example, one can subscribe for $100.00, $200.00, $500.00, or $1000.00.
- **Number of items**: The user buys the subscription for a particular number of images. For example, one can subscribe for 1 item, 3 items, 5 items, or 10 items.

The WebSphere Digital Media Enabler sample store contains sample subscription products for all three types of subscriptions. The subscription product partnumbers are defined in `DMConfig.properties`:

```
static final String WCS_SUBSCRIPTION_DURATION_PARTNUMBER = "19998";

static final String WCS_SUBSCRIPTION_PACKAGE_PARTNUMBER = "19999";

static final String WCS_SUBSCRIPTION_AMOUNT_PARTNUMBER = "19995";
```

You will need to define your subscription offerings and then create the appropriate subscription products. To set up your subscription offerings:

1. For each item, set ManufacturerName to:
   - P (package subscription)
   - D (duration subscription)
   - A (amount subscription)
2. For each item, set ManufacturerPartNumber to:
   - Number of items for package subscription
   - Number of months for duration subscription
   - Dollar amount for amount subscriptions

Refer to Appendix A. for an XML sample for defining a subscription product.

Use cases

The following use cases are used to illustrate the WebSphere Digital Media Enabler subscription policies.

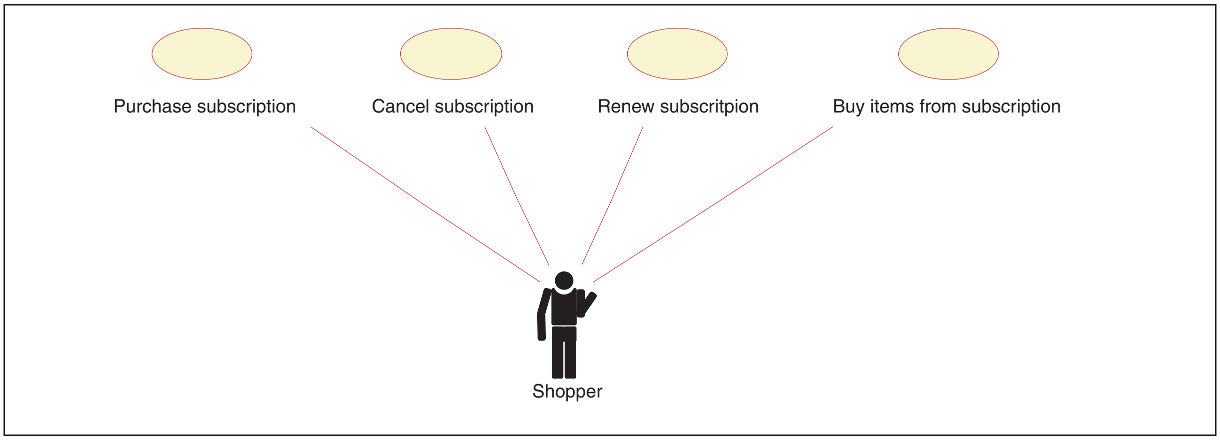

Figure 6. Subscription use cases

A sequence diagram for each use case follows:

Purchase subscription sequence

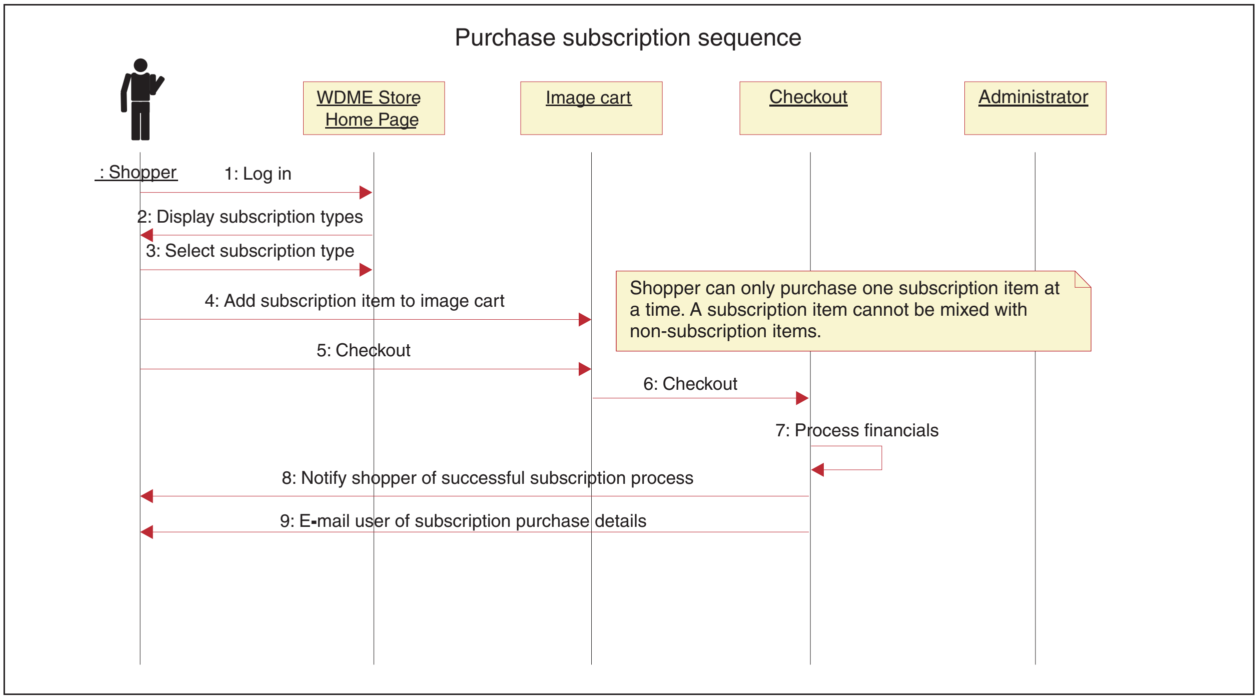
Purchase subscription sequence
WDME Store Home Page
Image cart
Checkout
Administrator
: Shopper
1: Log in
2: Display subscription types
3: Select subscription type
4: Add subscription item to image cart
5: Checkout
6: Checkout
7: Process financials
8: Notify shopper of successful subscription process
9: E-mail user of subscription purchase details
Shopper can only purchase one subscription item at a time. A subscription item cannot be mixed with non-subscription items.

Figure 7. Purchase subscription sequence

Cancel subscription sequence

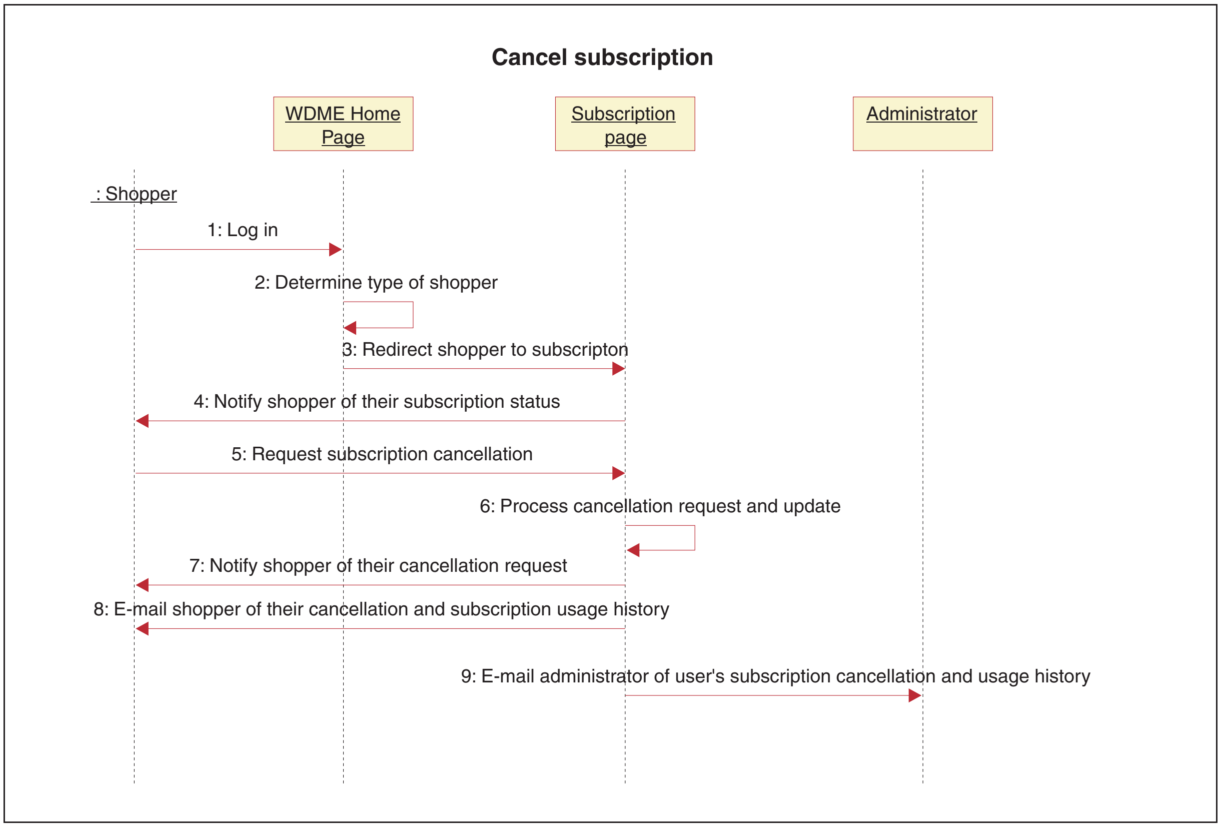
Cancel subscription
WDME Home Page
Subscription page
Administrator
: Shopper
1: Log in
2: Determine type of shopper
3: Redirect shopper to subscripton
4: Notify shopper of their subscription status
5: Request subscription cancellation
6: Process cancellation request and update
7: Notify shopper of their cancellation request
8: E-mail shopper of their cancellation and subscription usage history
9: E-mail administrator of user's subscription cancellation and usage history

Figure 8. Cancel subscription sequence

Renew subscription

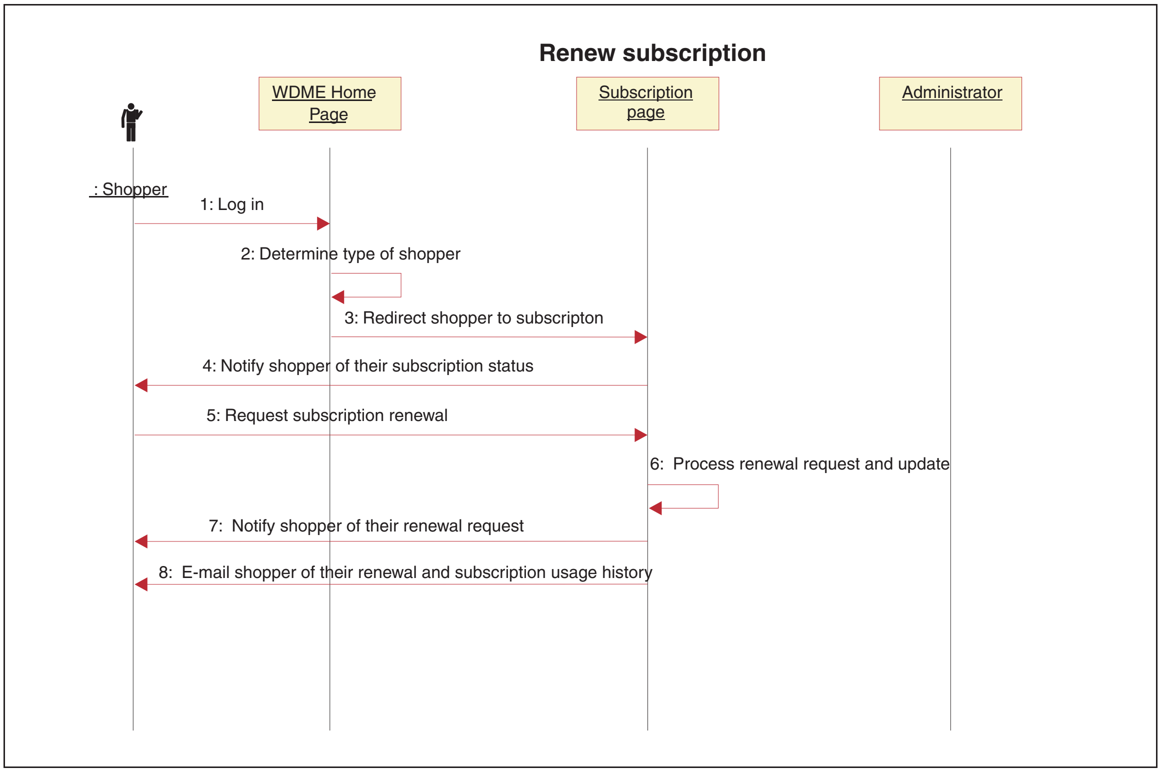
Renew subscription
WDME Home Page
Subscription page
Administrator
: Shopper
1: Log in
2: Determine type of shopper
3: Redirect shopper to subscripton
4: Notify shopper of their subscription status
5: Request subscription renewal
6: Process renewal request and update
7: Notify shopper of their renewal request
8: E-mail shopper of their renewal and subscription usage history

Figure 9. Renew subscription sequence

Buy items from subscription

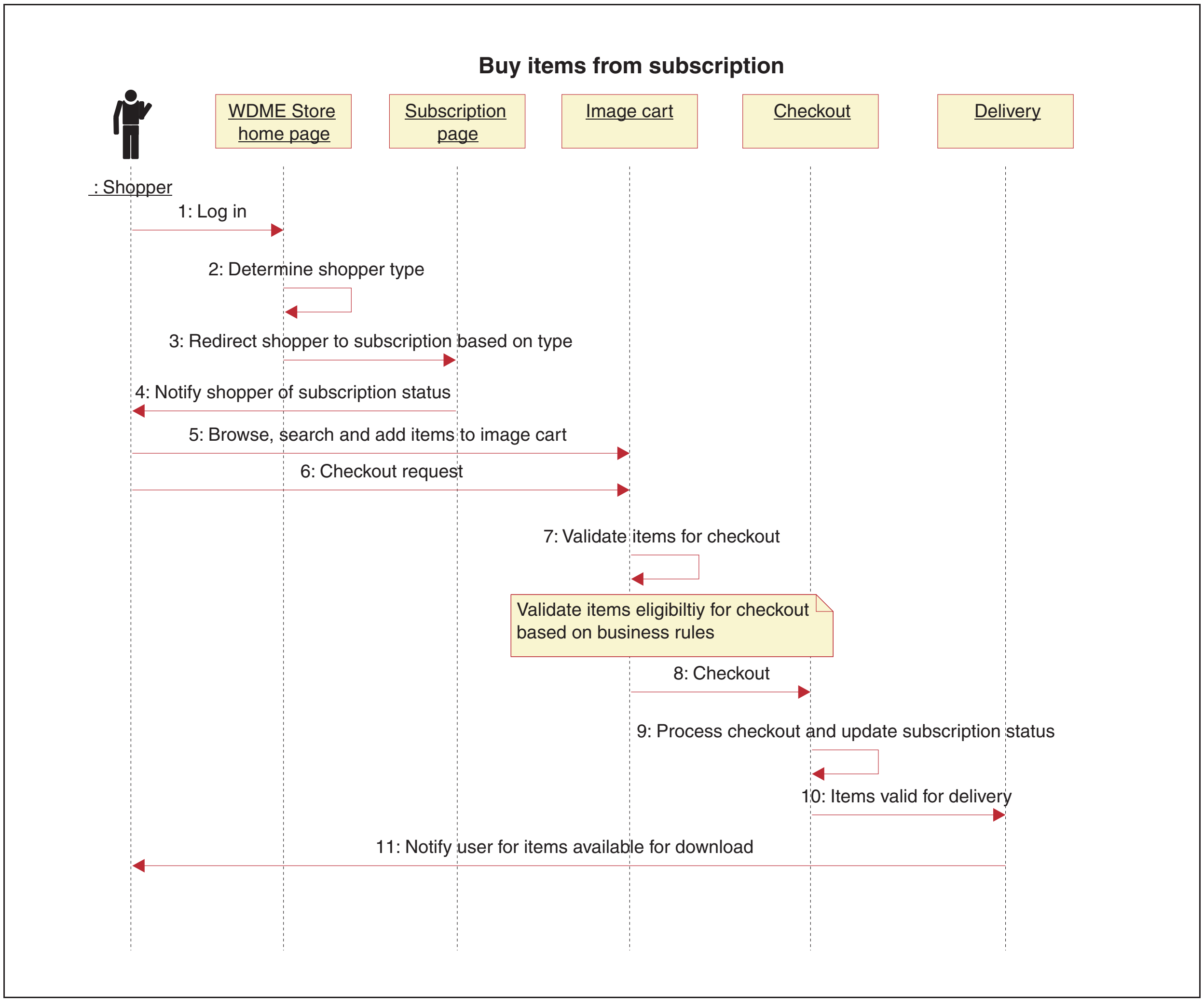
Buy items from subscription
: Shopper
WDME Store home page
Subscription page
Image cart
Checkout
Delivery
1: Log in
2: Determine shopper type
3: Redirect shopper to subscription based on type
4: Notify shopper of subscription status
5: Browse, search and add items to image cart
6: Checkout request
7: Validate items for checkout
Validate items eligibiltiy for checkout based on business rules
8: Checkout
9: Process checkout and update subscription status
10: Items valid for delivery
11: Notify user for items available for download

Figure 10. Buy items from subscription sequence

Setting Media Folder collaboration using a Sametime server

To allow users to collaborate interactively while viewing Media Folders, you must install a Sametime server and configure the Sametime parameters in `dmstoreconfig_`*`locale`*`.properties`. Use the following parameters.

- To enable Sametime collaboration: `ENABLE_SAMETIME_FEATURE = 1`
- To set base URL: `SAMETIME_LINKS_BASE_URL = /sametime/stlinks`
- If using Sametime 2.5: `SAMETIME_LINKS_VERSION = 2.5`
- If using Sametime 2.6: `SAMETIME_LINKS_VERSION = 2.6`
- If using Sametime 3.0: `SAMETIME_LINKS_VERSION = 3.0`
- `SAMETIME REG URL= http://{host_name}[:port_number]/ streg.nsf/NewAcc?CreateDocument`

As a result, users will be automatically registered in the Sametime server when they register in WebSphere Digital Media Enabler. At login, users will be automatically logged in to the Sametime server. To turn off autoconnect, use the Edit Account page.

If the Sametime Server is installed on the same machine as Websphere Commerce, the default port must be changed, so that it does not conflict with the IBM HTTP server or Websphere Commerce. If a new port is defined, it must be specified for SAMETIME_REG_URL and SAMETIME_LINKS_BASE_URL.

Chapter 6. Customizing content management

The digital content repository used with WebSphereDigital Media Enabler can be managed with either Content Manager or with a non-IBM content repository. This section describes how to customize Content Manager and how to implement WebSphere Digital Media Enabler's session bridge feature for use with a non-IBM content management system.

Creating the Content Manager WebSphere Digital Media Enabler item type

For Content Manager, you will need to use the Content Manager System Administration Client to create an item type and its attributes.

Using the content loading utility

Most of the tasks involved in content loading are documented in the *WebSphere Digital Media Enabler Administration Guide*. This section focuses on the tasks necessary to customize the utility program prior to exercising the steps described in the chapter titled *Loading content into WebSphere Digital Media Enabler* in the *WebSphere Digital Media Enabler Administration Guide*.

There are three areas in content loading that vary from customer to customer:

- The parameters specific to Content Manager, such as search parameters and location of Content Manager parts
- The mapping of field names in the metadata input file to specific Content Manager fields
- The metadata fields that need to be replicated over to the WebSphere Commerce catalog

Content Manager parameters

There is a file named `parms.txt` located in the following directory. Change the value for the fields in the file to reflect your customer environment.

```
WAS_installdir/installedApps/hostname/WC_instance_name.ear/
Stores.war/store_directory>/Custom_mass_loader/
```

Here is an example:

```
!index mode off text search entries not put in table
TRACE_MODE=ON
INDEX_MODE=ON
INPUTFILE=Sample.txt
USER=wcdmp
PASSWORD=wcdmp
SERVER_TYPE=ICM
SERVER=LIBSRVRN
INDEX_CLASS_NAME=DMDEMO
SEARCH_ENGINE=SM
SEARCH_INDEX=TM-DMDEMO
SEARCH_LANG=ENU
COMMERCETHUMBNAILPATH=WAS_installdir/installedApps/hostname/WC_instance_name.ear
    /Stores.war/DigitalMedia/webobjects//thumbnails
STORAGE_GROUP_MEDIUM=cbr.cllct001
STORAGE_GROUP_HIGH=cbr.cllct001
```

This file must be placed in the same directory as the load program for it to be picked up by the loading utility; for example,

WAS_installdir/installedApps/*hostname*/*WC_instance_name*/
Stores.war/*store_directory*/custom_mass_loader/

Defining your field attributes for loading into Content Manager

Modify the file dmstoreload.properties to reflect your specific data model. This file is located in the following directory:

WAS_*installdir*/installedApps/*hostname*/WC_instance_name.ear/Stores.war/WEB-INF/
classes/store_directory

This file defines how your fields are to be handled. You must specify: (1) whether the field is single- or multi-valued; (2) whether the field should be indexed as free text; (3) whether the field is to be indexed as a keyword that can be used as a search filter; (4) whether the field is character such that it needs single quotes in any generated XML; and, (5) whether the field is a column in the item type, or in the upload library table, or both.

Open the file with an editor and go to the Attribute Definition section to do the field mapping. This is done in the constructor method of the AttributeDefinition class.

- The first parameter to the constructor is the attribute name. The attribute name is the name of the field attribute to be created in the item type. Each attribute name MUST match the name in the name-value pair values used in the metadata input file; for example, the name-value pair "imageDesc = Man skiing down hill" in the metadata input file will be put into the Content Manager field attribute named imageDesc.
- The second parameter is a Boolean: TRUE if the field has multiple values, FALSE if it has a single value; for example, imageKeywords is TRUE.
- The third parameter is a Boolean: TRUE if the field is to be text-indexed, FALSE if it is not; for example, imageDesc is TRUE, duration is FALSE.
- The fourth parameter is a Boolean: TRUE if the field is to be included into the text part as a keyword name-value pair, FALSE if it is not; for example, clipNumber is TRUE in the third parameter to be text-indexed, but is FALSE in the fourth parameter because it is not included in the text part as a name-value pair. Being a name-value pair in the text part allows this value to be used as a filter to an overall full text search without doing a parametric search.
- The fifth parameter is used to indicate if the field is a character field which will need to be surrounded by single quotes in any XML that is generated for this field.
- The sixth parameter indicates whether the column is in the upload library table or in the Content Manager item type, or both.

Here is an example:

```
#-------------------------------------------------------------
#
# WCDM FIELDS
#--------------------------------------------------------------
#
# This properties file makes customization of the field names.
# Each input field maps to a CM field in an index class.
#--------------------------------------------------------------
#
# Example:
# field#= value          ,SM       ,FT       ,NV        ,SQ       ,WH       SM=Single or Multi;
#                                                                           FT=Free Text;
#                                                                           NV=Name-Value;
#                                                                           ST=Single Quote;
#                                   WH=0 for Library, WH=1 for index Class, WH=2 for Both
#---------------------------------------------------------------------------------------------------
#

#Required fields (cannot change the field name also)
field1= imageID                  ,false,false,true ,false ,2
field2= imageDesc                ,false,true,true,true ,2
field3= cameraElement            ,false,true ,true,true ,2
field4= imageLDeleteFlg          ,false,false,true,true ,2
field5= imageKeywords            ,true,true,true,true ,2
#this field goes in to indexClass as a part
field6= thumbnailFile            ,false,false,false,true ,2
#this field goes in to indexClass as a part
field7= lowResFile               ,false,false,false,true ,2
#this field goes in to indexClass as a part
field8= highResFile              ,false,false,false,true ,2
field9= imageStatus              ,false,false,true ,true ,2
field10= libraryID               ,false,false,false,false,0
field11= createDate              ,false,false,false,true ,0
field12= modifyDate              ,false,false,false,true ,0
field13= batchID                 ,false,false,false,true ,0
field14= batchDate               ,false,false,false,true ,0
field15= imageSequence           ,false,false,false,false,0
#additional fields (customizable)
field16= imagePhotClass         ,false,false,true ,true ,2
field17= TCIn                   ,false,false,false,true ,2
field18= TCOut                  ,false,false,false,true ,2
field19= duration               ,false,false,false,true ,2
field20= FBDTapeNo              ,false,true,true ,true ,2
field21= clipNumber             ,false,true,false,true ,2
field22= clipCatalogType        ,false,true,false,true ,2
field23= clipFootageDate        ,true,true,false,true ,2
field24= age                    ,true,true,true,true ,2
field25= era                    ,true ,true,true ,true ,2
field26= imageRestricts         ,false,false,true ,true ,2
field27= timePeriod             ,true ,true,true ,true ,2
field28= imageRelease           ,true ,false,true ,true ,2
field29= imageHighRAvail        ,false,false,true ,true ,2
field30= imageCaption           ,false,true,true ,true ,2
field31= imageColor             ,true,true,true ,true ,2
field32= audio                  ,true,true,true ,true ,2
field33= cameraSpeed            ,true,true,true ,true ,2
field34= composition            ,true,true,true ,true ,2
field35= category               ,true,true,true ,true ,2
field36= subcategory            ,true,true,true ,true ,2
field37= intExt                 ,true,true,true ,true ,2
field38= timeOfDay              ,true,true,true ,true ,2
field39= productType            ,false,false,false,true ,2
field40= imageSize              ,false,false,false,true ,2
field41= fileSize               ,false,false,false,true ,2
field42= compression            ,false,false,false,true ,2
field43= CDSKU                  ,false,false,true ,true ,1
field44= clipPrice              ,false,false,false,true ,1
field45= Reserved1              ,false,false,true ,true ,1
field46= Reserved2              ,false,false,true ,true ,1
field47= Reserved3              ,false,false,true ,true ,1
field48= Reserved4              ,false,false,true ,true ,1
field49= clippricehi            ,false,false,true ,true ,2
field50= clippricelo            ,false,false,true ,true ,2
```

Defining your upload user responsibilities

If you are using the upload feature, you must define which users are authorized to perform the functions of uploading, indexing, and approving new content. This is done in the `dmstoreload.properties` file by specifying the user actions and user authorizations as shown below. You will then use the Accelerator to set up the authorizations for the appropriate users.

An example follows:

```
#############################################################
# DigitalMedia Library Flow
# If the value is YES then the approval logic will be used.
# If it is NO then the normal logic, where a single user
# can upload and index the metadata, will be used.
#############################################################
WCDMLIBRARYAPPROVALPROCESS = YES
```

```
############################################################
# Library User Roles
# The field is no of household members in the userdemo table
#
# 0 = Normal User (Upload Low res)
# 1 = Cataloger (Enter MetaData)
# 2 = Supervisor (Upload Low res and enter metadata) (0 and 1)
# 3 = Admin (Upload Low res, enter metadata and index it)

# ####################################################
WCDMLIBRARYUPLOAD = 0|2|3
WCDMLIBRARYINDEX = 1|2|3
WCDMLIBRARYAPPROVE = 3

#-------------------------------------------------------------------------#
# Upload daemon properties. start
# d:/WebSphere/WAS_installdir/installedApps/hostname/WC_Enterprise_App_demo.ear/
Stores.war/DigitalMedia/custom_mass_loader/upload.bat
#-------------------------------------------------------------------------#

# upload daemon sleep time in minutes
WCDM.DAEMONSLEEPTIME=10

# The custom mass loader directory is located in
WCDM.MASSLOADERDIR =
d:WAS_installdir/installedApps/hostname/WC_Enterprise_App_demo.ear/
Stores.war/SVTStore2/custom_mass_loader


# Library Uploaded Images Path
# This is the location where all the images are uploaded during Library Upload #
# Same as LIBRARY_UPLOAD_PATH variable in the
/Stores.war/DigitalMedia/common/lightbox/jsp/DMConfig.inc file

WCDM.LIBRARYIMAGEUPLOADPATH =
d:WAS_installdir/installedApps/hostname/WC_Enterprise_App_demo.ear/
Stores.war/SVTStore2/common/webobjects/uploadedimages

# Load the data in WCS
WCDM.RUNWCSLOADER = YES

# Load the data in Content Manager
WCDM.RUNCMLOADER = YES

# Number of days files(logs,errors and processed) in the upload directory should be kept.
# If the number is -1 then the files are never deleted
WCDM.DELETEDAYS=-1

# Clean the database and related files which are in the status X
WCDM.CLEANFILES = YES
```

Here is an example from CM8repository.properties:

```
#===============================================================================
# Properties for Repository Implementation CM8RepositoryImpl:
#
# [Class: com.ibm.commerce.digitalmedia.repository.impl.CM8RepositoryImpl]
#===============================================================================
#--- Library Server Name
CM8.LIBRARYSERVERNAME=ICMNLSDB

#--- Content Manager User
CM8.CMUSER=wcdmp

#--- Content Manager Password
CM8.CMPASSWORD=wcdmp

#--- Item Type Name
CM8.WCDMCMINDEXCLASS=DMDEMO

#--- Part Numbers (Item Type)
CM8.MEDIA_PART_ID_TEXT           = 1
CM8.MEDIA_PART_ID_TOC            = 2
CM8.MEDIA_PART_ID_THUMBNAIL      = 3
```

```
CM8.MEDIA_PART_ID_LOW_RES        = 5
CM8.MEDIA_PART_ID_HIGH_RES       = 7
CM8.MEDIA_PART_ID_STORYBOARD     = 6
CM8.MEDIA_PART_ID_DESCRIPTION    = 9
CM8.MEDIA_PART_ID_KEYWORDS       = 11
CM8.MEDIA_PART_ID_STORYBOARD_TOC = 12

#------------------------------------------------
# Content Manager Search Locale Values
# The values are in cmbroot/include/DKConstant2.h
# Not to be changed for US English
#------------------------------------------------
#--- The WCDMCMCCSID value is same as DK_CCSID_00819
CM8.WCDMCMCCSID=819

#--- The WCDMCMCCSID value is same as DK_LANG_ENU
CM8.WCDMCMLANDID=6011
```

Replicating metadata into the WebSphere Commerce catalog

You will define your mapping of input fields to WebSphere Commerce tables.

Once content is loaded in IBM Content Manager, you should convert the metadata input file used by the content loading utility to a WebSphere Commerce XML file for passage to the WebSphere Commerce loading utility. The latter will put the selected metadata into the WebSphere Commerce catalog. The conversion utility is a Java program called `CMtoWCS.class`. The `dmstoreload.properties` file contains the externalized values for the conversion.

Here is an example that maps the input `imageDesc` to the long description, etc.

```
# ---------------------------------------------------------------------------------#
# FIELD Mappings to WCS
# ---------------------------------------------------------------------------------#
# This properties file makes customization of the
# Each CM field maps to a WCS field as shown
# ---------------------------------------------------------------------------------#
# Example:
# wcsfield1=>catentry_id
# wcsfield1=imageID
# ---------------------------------------------------------------------------------#
#wcsfield1=>catentry_id
wcsfield1=imageID
#wcsfield2=>long description
wcsfield2=imageDesc
#wcsfield3=imageID
wcsfield3=>mfpartnumber
#wcsfield4=>field4
wcsfield4=duration
#wcsfield5=>auxdescription1
wcsfield5=Compression
#wcsfield6=>compareprice
wcsfield6=clippricelo
#wcsfield7=>price
wcsfield7=clippricehi
#wcsfield8=>fullimage
wcsfield8=lowResFile
```

Defining your fields for loading ACSII delimited input text data

If the metadata is extracted from an existing legacy database into an ASCII delimited file, the following steps can be used to import into IBM Content Manager.

The loading process allows you to supply your input in an ASCII delimited format, similar to a database export. A sample entry for an image would look like this:

```
Man and Woman playing Basketball.|Image||IBM; man; woman; Sports; fun; excitement;
basketball; doubles; Leisure|01:06:56:08|01:07:15:2|19:06|11196HSH|11196HSH-001|
New|03/01/2001|Adult|Contemporary||N|1990s|60400|Y|Y|Man and Woman playing Basketball|
C|N|S|NS|Sports|Summer Sports; Basketball|Exterior|Day|video clip|400x800|640KB|
MPEG1; MPEG2|19997|199.00|100|120|||120.00|200.00|"images\5k\60400.jpg"|"images\
85k\11196HSH-001_300k.jpg"|"images\85k\11196HSH-001_300k.jpg"
```

Each column is separated by a special character: `|` . You must define what is in each column in the `dmstoreload.properties` file. Once you define your data model, you must customize this properties file for your input data fields as shown previously.

Defining your fields for loading via XML input

You may also load your input data by defining the content using XML. A sample XML file for loading an Adobe Acrobat PDF file with an image ID of 68010 is listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
<wcdmRecords>
<wcdmRecord>
<action>purge</action>
<imageID>68010</imageID>
<imageDesc>Test the Library Data</imageDesc>
<cameraElement>Acrobat</cameraElement>
<cameraElement>Acrobat</cameraElement>
<imageLDeleteFlg></imageLDeleteFlg>
<imageKeywords>Library,Upload</imageKeywords>
<thumbnailFile>upload/incoming/images/Thumbnail\68010.jpg</thumbnailFile>
<lowResFile>upload/incoming/images/Lowres\68010.pdf</lowResFile>
<highResFile></highResFile>
<imageStatus>W</imageStatus>
<imagePhotClass></imagePhotClass>
<TCIn></TCIn>
<TCOut></TCOut>
<duration></duration>
<FBDTapeNo></FBDTapeNo>
<clipNumber>68010</clipNumber>
<clipCatalogType></clipCatalogType>
<clipFootageDate></clipFootageDate>
<age></age>
<era></era>
<imageRestricts>Y</imageRestricts>
<timePeriod></timePeriod>
<imageRelease>Y</imageRelease>
<imageHighRAvail></imageHighRAvail>
<imageCaption></imageCaption>
<imageColor>C</imageColor>
<audio></audio>
<cameraSpeed>R</cameraSpeed>
<composition></composition>
<category>Library</category>
<subcategory>Upload</subcategory>
<intExt>Exterior</intExt>
<timeOfDay>Night</timeOfDay>
<productType></productType>
<imageSize></imageSize>
<fileSize>200</fileSize>
<compression></compression>
<clippricehi>100</clippricehi>
<clippricelo>50</clippricelo>
</wcdmRecord>
</wcdmRecords>
```

Integrating with external Content Management systems

If you already have a content repository other than Content Manager, you have a customization option to easily integrate your current repository with the rest of the WebSphere Digital Media Enabler solution. This feature is called the WebSphere Digital Media Enabler session bridge. When using the session bridge, you create a bridge between the WebSphere Digital Media Enabler application and your external content management system. Essentially, the Digital Media storefront operates as designed, except the external repository interface is used for search and retrieval of content. There are two fundamental integration scenarios — one where WebSphere Digital Media Enabler is the front-end system and one where the external content repository is the front-end system. By front-end system we mean that the user enters and registers through that system.

Integrating using WebSphere Digital Media Enabler as the front-end system

In this scenario, users access WebSphere Digital Media Enabler as the primary Web site and register there. All WebSphere Digital Media Enabler functions are used except for the search and retrieval of content. When a user invokes the search functionality, the WebSphere Digital Media Enabler search link URL will point to the external content manager rather than the WebSphere Digital Media Enabler search URL.

The external content manager will provide the search, search results, and item detail display functionality through its own interface. In addition, the external search HTML pages will be modified to contain links back to WebSphere Digital Media Enabler to:

- Add an item to the product cart.
- View the product cart.
- Add an item to a media folder.
- View media folders.
- View order history.

The WebSphere Digital Media Enabler session is maintained during these linkages via the WebSphere Digital Media Enabler session bridge. The WebSphere Digital Media Enabler download function will also have to be modified to access the external repository for delivery to the end user.

Integrating using the external repository as the front-end system

In this scenario, users access the external content repository as the primary Web site and register there. The repository provides for searching and retrieval of the content. When an item of interest is found by the user, the linkages to add the item to the cart or media folder are provided. As in the first scenario above, the user can link to the Media Folder and order functions.

When the user attempts to access the WebSphere Digital Media Enabler functions, the user will be transparently logged on to WebSphere Digital Media Enabler. If this is the first time and the user is not yet registered in WebSphereDigital Media Enabler, the user will be automatically registered into WebSphere Digital Media Enabler before proceeding with the user request. The user’s existing Content Manager ID will be used for the behind-the-scenes WebSphere Digital Media Enabler registration.

Integrating an external content management system

To integrate an external content management system with WebSphere Digital Media Enabler, you will need to perform the following

- Implement the session bridge
- Modify download and delivery functions
- Extract metadata and load product tables

:

Implement the session bridge

To implement the session bridge:

1. Modify the content manager search results page by adding the WebSphere Digital Media Enabler form with links to the WebSphere Digital Media Enabler pages.
2. Modify the WebSphere Digital Media Enabler `Sidebar.inc` file to contain links to the external content manager search page.
3. Modify the session bridge JSP file (`DMSearchSessionBridge.jsp`) to reflect the external content manager search URL.

Modify download and delivery functions

The download servlet and delivery daemon are implemented with Content Manager APIs to query and retrieve content. You will need to modify these to access your content management system or deliver content via another mechanism.

Extract content metadata and load WebSphere Commerce product tables

Any digital content that can be processed in WebSphere Digital Media Enabler must have entries in the WebSphere Commerce product tables. For each digital product, you will need to:

1. Extract the necessary product metadata from your repository.
2. Format it into XML suitable for loading into WebSphere Commerce.
3. Run it through the WebSphere Commerce ID resolver.
4. Load the tables using the WebSphere Commerce loader.

Note that if you want to utilize the WebSphere Digital Media Enabler Upload Manager feature, you will need to customize the process to ingest the upload content into your system.

Example of implementing the WebSphere Digital Media Enabler session bridge with Virage on Windows NT

1. Copy the WebSphere Digital Media Enabler bridge JavaScript file to the `virage` directory.

   Copy `wcdm_sessionbridge.js` from

   ```
   drive: WAS_installdir\installedApps\WC_instance_name.ear\
   Stores.war\store_directory\SessionBridge
   ```

   to

   ```
   drive:\<Virage_directory>\VS\vssdocs\accounts\sample\search
   ```

2. Save the Virage Search Results template.

   Save the original `results.tmpl` in the directory `drive:\<Virage_directory>\VS\conf\accounts\sample\search`

   `drive` is the location where Virage is installed.

3. Copy the form that implements the links to WebSphere Digital Media Enabler into the Virage template.

   Copy and paste the portion (including the tags) between

   ```
   <form name="extdmform" action="">
   ```

   and `</form>` to `results.tmp` from the file with the same name in the directory

   ```
   drive: WAS_installdir\installedApps\WC_instance_name.ear\
   Stores.war\store_directory\SessionBridge\samples
   ```

   This portion of the file provides sample code showing how to implement a link to WebSphere Digital Media Enabler from the Virage search result page. It has links for View Product Cart, View Media Folder, View Order History, Add to Product Cart, and Add to Media Folder.
4. Update the WebSphere Digital Media Enabler sidebar with a link to the Virage Search page.

   The sample code embedded between `<%//virage.search.sn%>` and `<%//virage.search.en%>` in `DMSidebar.inc` shows how to link to the Virage search page. It is contained in the directory:

   ```
   drive: WAS_installdir\installedApps\WC_instance_name.ear\
   Stores.war\store_directory\common\lightbox\jsp
   ```

   The sample code provides a text entry field for **Virage Search** along with a **Go** button on the sidebar of WebSphere Digital Media Enabler pages to allow the user to provide a key word for returning to Virage side to perform search capability.
5. Update the session bridge JSP file to reflect the correct hostname for Virage.

   Modify the `DMSearchSessionBridge.jsp` in the directory.

   ```
   drive: WAS_installdir\installedApps\WC_instance_name.ear\
   Stores.war\store_directory\common\lightbox\jsp
   ```

   Change the tag

   ```
   <form name="viragesearch" action="http://<host-name>/vss-bin/vss_SR.exe/sample/search">
   ```

   to reflect the correct hostname of the machine which is running Virage.

   This JSP file serves as the interface to link the user back to Virage when the search task is kicked off from the sidebar of WebSphere Digital Media Enabler pages.

Customizing the ZIP utility for delivery to Macintosh customers

If your customers are Macintosh users, be aware that you must use a ZIP function that is usable by both Macintosh and PC users.

By default, WebSphere Digital Media Enabler uses the Java ZIP utility to package content for downloading to the end user's PC. If content is to be delivered to Macintosh users, you must use a ZIP utility that creates a `.zip` format that can be processed by both PC and Macintosh systems; for example, Winzip or gnuzip. The ZIP utility must be capable of being invoked as a command and the command must be defined in the `delivery.properties` and `DMConfig.inc` files.

▶ Windows

- `DMConfig.inc`

`static final String MEDIAZIP_COMMAND =` (to point to the path of whatever ZIP tool utility you have installed)

If you will be using Winzip, set the value as follows:

```
static final String MEDIAZIP_COMMAND="drive:\\progra~1\\winzip\\wzzip -rp -u -x*.zip"
```

It may be necessary for you to install the Command Line Support Add-On from WinZip's site at `http://www.winzip.com`. The Winzip must be a registered version of the utility in order to work properly.

- `delivery.properties`

  `WCDM.ZIPCMD=C:/progra~1/winzip/wzzip -rp -u -x*.zip`

▶ UNIX

- `DMConfig.inc`

  `static final String MEDIAZIP_COMMAND =` (to point to the path of whatever ZIP tool utility you have installed)
- `delivery.properties`

  `WCDM.ZIPCMD=` (to point to the path of whatever ZIP tool utility you have installed)

Appendix A. Sample subscription product XML

The following is a sample of the XML to define a subscription by duration:

```
<!-- DigitalMedia Sample subscription by Duration Data Start -->
<!-- Subscription by Duration -->

<!-- baseitem table-->
<!-- BaseItems represent a general family of goods with a common name and description.
BaseItems are used exclusively for fulfillment. Each CatalogEntry that represents a
Product in the catalog has a corresponding BaseItem for fulfillment purposes.  -->
<baseitem
baseitem_id="@baseitem_id_19998"
member_id="&MEMBER_ID;"
markfordelete="0"
partnumber="19998"
itemtype_id="ITEM"
quantitymeasure="C62"
quantitymultiple="1.0"
/>
<!--itemversn table -->
<!-- Each row of this table represents an ItemVersion for a BaseItem. An ItemVersion
expires when its expiration date is in the past. Each BaseItem must only have a single
ItemVersion defined. -->
<itemversn
itemversn_id="@itemversn_id_19998"
baseitem_id="@baseitem_id_19998"
expirationdate="2010-01-01 00:00:00.000000"
versionname="version"
/>
<!--itemspc table -->
<!-- Information about specified items.
A specified item is a product with values for all
its attributes. A specified item is the customer view
of what the merchant sells. -->
<itemspc
itemspc_id="@itemspc_id_10019998"
baseitem_id="@baseitem_id_19998"
markfordelete="0"
partnumber="10019998"
member_id="&MEMBER_ID;"
discontinued="N"
/>
<itemspc
itemspc_id="@itemspc_id_20019998"
baseitem_id="@baseitem_id_19998"
markfordelete="0"
partnumber="20019998"
member_id="&MEMBER_ID;"
discontinued="N"
/>
<itemspc
itemspc_id="@itemspc_id_30019998"
baseitem_id="@baseitem_id_19998"
markfordelete="0"
partnumber="30019998"
member_id="&MEMBER_ID;"
discontinued="N"
/>
<itemspc
itemspc_id="@itemspc_id_40019998"
baseitem_id="@baseitem_id_19998"
markfordelete="0"
partnumber="40019998"
member_id="&MEMBER_ID;"
discontinued="N"
/>
<!-- versionspc -->
<!– Each row defines the relationship
between a product version and a specified item. –>
<versionspc
versionspc_id="@versionspc_id_10019998"
itemspc_id="@itemspc_id_10019998"
itemversn_id="@itemversn_id_19998"
/>
```

```
<versionspc
versionspc_id="@versionspc_id_20019998"
itemspc_id="@itemspc_id_20019998"
itemversn_id="@itemversn_id_19998"
/>
<versionspc
versionspc_id="@versionspc_id_30019998"
itemspc_id="@itemspc_id_30019998"
itemversn_id="@itemversn_id_19998"
/>
<versionspc
versionspc_id="@versionspc_id_40019998"
itemspc_id="@itemspc_id_40019998"
itemversn_id="@itemversn_id_19998"
/>
<! --distarrang -->
<! -- Each row of this table represents
a DistributionArrangement, enabling a Store to sell
its own inventory. -->
<distarrang
distarrang_id="@distarrang_id_19998"
wholesalestore_id="@storeent_id_1"
merchantstore_id="@storeent_id_1"
baseitem_id="@baseitem_id_19998"
pickingmethod="F"
startdate="2000-12-25 00:00:00.000000"
enddate="2010-01-01 00:00:00.000000"
/>
<!-- receipt -->
<!--Each row contains information about
each receipt of an item at a FulfillmentCenter.  –>
<receipt
receipt_id="@receipt_id_10019998"
store_id="@storeent_id_1"
ffmcenter_id="@ffmcenter_id_1"
versionspc_id="@versionspc_id_10019998"
vendor_id="@vendor_id_1"
receiptdate="2001-01-01 00:00:00.000000"
qtyreceived="100"
qtyinprocess="0"
qtyonhand="100"
qtyinkits="0"
receipttype="ADHC"
createtime="2000-12-25 00:00:00.000000"
/>
<receipt
receipt_id="@receipt_id_20019998"
store_id="@storeent_id_1"
ffmcenter_id="@ffmcenter_id_1"
versionspc_id="@versionspc_id_20019998"
vendor_id="@vendor_id_1"
receiptdate="2001-01-01 00:00:00.000000"
qtyreceived="100"
qtyinprocess="0"
qtyonhand="100"
qtyinkits="0"
receipttype="ADHC"
createtime="2000-12-25 00:00:00.000000"
/>
<receipt
receipt_id="@receipt_id_30019998"
store_id="@storeent_id_1"
ffmcenter_id="@ffmcenter_id_1"
versionspc_id="@versionspc_id_30019998"
vendor_id="@vendor_id_1"
receiptdate="2001-01-01 00:00:00.000000"
qtyreceived="100"
qtyinprocess="0"
qtyonhand="100"
qtyinkits="0"
receipttype="ADHC"
createtime="2000-12-25 00:00:00.000000"
/>
<receipt
receipt_id="@receipt_id_40019998"
store_id="@storeent_id_1"
ffmcenter_id="@ffmcenter_id_1"
versionspc_id="@versionspc_id_40019998"
vendor_id="@vendor_id_1"
receiptdate="2001-01-01 00:00:00.000000"
```

```
qtyreceived="100"
qtyinprocess="0"
qtyonhand="100"
qtyinkits="0"
receipttype="ADHC"
createtime="2000-12-25 00:00:00.000000"
/>
<!--rcptavail -->
<!-- Defines which DistributionArrangements
have access to received inventory. -->
<rcptavail
rcptavail_id="@rcptavail_id_10019998"
distarrang_id="@distarrang_id_19998"
receipt_id="@receipt_id_10019998"
/>
<rcptavail
rcptavail_id="@rcptavail_id_20019998"
distarrang_id="@distarrang_id_19998"
receipt_id="@receipt_id_20019998"
/>
<rcptavail
rcptavail_id="@rcptavail_id_30019998"
distarrang_id="@distarrang_id_19998"
receipt_id="@receipt_id_30019998"
/>
<rcptavail
rcptavail_id="@rcptavail_id_40019998"
distarrang_id="@distarrang_id_19998"
receipt_id="@receipt_id_40019998"
/>
<!--storeitem -->
<!– Each row of this table contains attributes
that affect how a particular Store allocates inventory
for the specified items of a particular BaseItem. –>
<storeitem
baseitem_id="@baseitem_id_19998"
storeent_id="@storeent_id_1"
trackinventory="Y"
forcebackorder="N"
releaseseparately="N"
returnnotdesired="N"
backorderable="N"
creditable="Y"
minqtyforsplit="0"
/>
<!--catentry table -->
<!-- This table holds the information related
to a CatalogEntry. Examples of CatalogEntries include
Products, Items, Packages, and Bundles. -->
<catentry
catentry_id="@catentry_id_19998"
baseitem_id="@baseitem_id_19998"
member_id="&MEMBER_ID;"
catenttype_id="ProductBean"
partnumber="19998"
mfpartnumber="SKU-19998"
mfname="DigitalMedia"
markfordelete="0"
buyable="1"
/>
<catentry
catentry_id="@catentry_id_10019998"
itemspc_id="@itemspc_id_10019998"
member_id="&MEMBER_ID;"
catenttype_id="ItemBean"
partnumber="10019998"
mfpartnumber="SKU-19998"
mfname="DigitalMedia"
markfordelete="0"
buyable="1"
/>
<catentry
catentry_id="@catentry_id_20019998"
itemspc_id="@itemspc_id_20019998"
member_id="&MEMBER_ID;"
catenttype_id="ItemBean"
partnumber="20019998"
mfpartnumber="SKU-19998"
mfname="DigitalMedia"
markfordelete="0"
```

```
buyable="1"
/>
<catentry
catentry_id="@catentry_id_30019998"
itemspc_id="@itemspc_id_30019998"
member_id="&MEMBER_ID;"
catenttype_id="ItemBean"
partnumber="30019998"
mfpartnumber="SKU-19998"
mfname="DigitalMedia"
markfordelete="0"
buyable="1"
/>
<catentry
catentry_id="@catentry_id_40019998"
itemspc_id="@itemspc_id_40019998"
member_id="&MEMBER_ID;"
catenttype_id="ItemBean"
partnumber="40019998"
mfpartnumber="SKU-19998"
mfname="DigitalMedia"
markfordelete="0"
buyable="1"
/>
<!--catentrel-->
<!-- This table holds containment relationships
between CatalogEntries. Examples of these relationships
are Product-Item. -->
<catentrel
catentry_id_parent="@catentry_id_19998"
catreltype_id="PRODUCT_ITEM"
catentry_id_child="@catentry_id_10019998"
sequence="1"
quantity="1"
/>
<catentrelcatentry_id_parent="@catentry_id_19998"
catreltype_id="PRODUCT_ITEM"
catentry_id_child="@catentry_id_20019998"
sequence="2"
quantity="1"
/>
<catentrel
catentry_id_parent="@catentry_id_19998"
catreltype_id="PRODUCT_ITEM"
catentry_id_child="@catentry_id_30019998"
sequence="3"
quantity="1"
/>
<catentrel
catentry_id_parent="@catentry_id_19998"
catreltype_id="PRODUCT_ITEM"
catentry_id_child="@catentry_id_40019998"
sequence="4"
quantity="1"
/>
<!--storitmffc -->
<!-- Each row of this table contains information
about a BaseItem for a particular Store (or all the
Stores in a StoreGroup) and FulfillmentCenter. -->
<storitmffc
baseitem_id="@baseitem_id_19998"
storeent_id="@storeent_id_1"
ffmcenter_id="@ffmcenter_id_1"
shippingoffset="86400"
/>
<!--itemffmctr-->
<!– Each row contains information about
reserved quantities, amount on backorder, and amount
allocated to backorders for items owned by a store
at a fulfillment center. –>
<itemffmctr
itemspc_id="@itemspc_id_10019998"
qtyallocbackorder="0"
qtybackordered="0"
qtyreserved="0"
ffmcenter_id="@ffmcenter_id_1"
store_id="@storeent_id_1"
/>
<itemffmctr
itemspc_id="@itemspc_id_20019998"
```

```
qtyallocbackorder="0"
qtybackordered="0"
qtyreserved="0"
ffmcenter_id="@ffmcenter_id_1"
store_id="@storeent_id_1"
/>
<itemffmctr
itemspc_id="@itemspc_id_30019998"
qtyallocbackorder="0"
qtybackordered="0"
qtyreserved="0"
ffmcenter_id="@ffmcenter_id_1"
store_id="@storeent_id_1"
/>
<itemffmctr
itemspc_id="@itemspc_id_40019998"
qtyallocbackorder="0"
qtybackordered="0"
qtyreserved="0"
ffmcenter_id="@ffmcenter_id_1"
store_id="@storeent_id_1"
/>
<!-- DigitalMedia Sample subscription by
Duration Data End -->
```

Appendix B. The following is a sample Virage HTML template

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="${template_lang}">
<head>
    <title>Search Results</title>

<VTL_INCLUDE NAME="head.tmpl">

<script type="text/javascript" src="${javascript_uri}/playlist.js"></script>
<script type="text/javascript">
<!--//
var del_msg = "Are you sure you want to delete this asset, completely\n" +
           "removing it from the asset index and affecting all the\n" +
           "views in this account?";
// -->
</script>

</head>

<body bgcolor="${color_view_body_background}" link="${color_view_body_link}"
 alink="${color_view_body_alink}" vlink="${color_view_body_vlink}"
onload="window.focus();<VTL_IF name="playlist_ids"><VTL_ELSE_IF
NAME="bin_id"><VTL_ELSE>set_hidden_search_fields();</VTL_IF>">
<font face="Verdana, Arial, sans-serif" size="2">

<VTL_IF name="playlist_ids">
<!-- purpose of calling this module here is to arrange the result
     order for the clips displaying under player in that long
     pop-up window -->
<vtl_script name="Virage::VTL::Playlist">
<vtl_call name="Virage::VTL::Playlist::create_playlist">
<hr>

<VTL_ELSE_IF NAME="bin_id">
<hr>

<VTL_ELSE>

<VTL_INCLUDE NAME="results_header.tmpl">

</VTL_IF>

<VTL_IF NAME="result_start">

<script type="text/javascript" language="javascript">
<!-- //
function get_pl_asset_ids()
{
  var a_type = 'asset_type=clips;';
  var asset_ids = "playlist_ids=" + new_asset_ids;
  var playlist_query = a_type + asset_ids;
  return playlist_query;

}
// -->
</SCRIPT>

<FORM name="playlist">

<VTL_LOOP NAME="results">

<VTL_INCLUDE NAME="results_result.tmpl">

</VTL_LOOP>

</FORM>

<!-- <br clear="all"> –>

<form name="extdmform" action="">
<pre>
<h5>
IBM WCDM server: <input type="text" name="wcdmServer"  <VTL_IF
NAME="wcdmServer">   value="${wcdmServer}" <VTL_ELSE>
```

```
value="tmws02.mpsnet.ibm.com"</VTL_IF>>
User logon ID:   <input type="text" name="logonId"
    <VTL_IF NAME="logonId">      value="${logonId}"
    <VTL_ELSE> value=""</VTL_IF>>
Store ID:        <input type="text" name="storeId"
    <VTL_IF NAME="storeId">      value="${storeId}"
    <VTL_ELSE> value="10151"</VTL_IF>>
Catalog ID:      <input type="text" name="catalogId"
  <VTL_IF NAME="catalogId">    value="${catalogId}"
  <VTL_ELSE> value="10151"</VTL_IF>>
E-mail address:  <input type="text" name="emailAddress" <VTL_IF
NAME="email1">       value="${email1}"
     <VTL_ELSE> value="test@us.ibm.com"</VTL_IF>>
Language ID:     <input type="text" name="langId"
     <VTL_IF NAME="langId">       value="${langId}"
     <VTL_ELSE> value="-1"</VTL_IF>>
</h5>
</pre>
<script type="text/javascript">
function get_asset_ids()
{
 var selected_asset_ids = new Array();
 if (count == 0) {
  alert("Please check (or select) one or more image items before being able to perform the task.");
return null;
}
 if (count == 1) {
  selected_asset_ids = new Array(1);
  selected_asset_ids[0] = playlist_asset_ids;
} else {
  selected_asset_ids = playlist_asset_ids.split(":");
 }
return selected_asset_ids
}
function adjust_asset_ids(asset_ids)
{
 var assetIdArray = null;
 if ((asset_ids
!= null) && (asset_ids.length > 0)) {

  assetIdArray = new Array(asset_ids.length);
  for (var i=0; i<asset_ids.length; i++) {
   assetIdArray[i] = "100603" + asset_ids[i];

 }
 }
 return assetIdArray;
}
function adjust_asset_ids_thumbnail(asset_ids)
{
 var assetIdArray = null;

 if
((asset_ids != null) && (asset_ids.length > 0))
{
  assetIdArray = new Array(asset_ids.length);
  for (var i=0; i<asset_ids.length; i++)
{
   assetIdArray[i] = "603" + asset_ids[i];

 }
 }
 return assetIdArray;
}
</script>
<script type="text/javascript"
src="${javascript_uri}/wcdm_sessionbridge.js"></script>
<!-- //
<a href="javascript:view_media_folder_manager(document.extdmform.wcdmServer.value,
document.extdmform.logonId.value, document.extdmform.langId.value,
document.extdmform.storeId.value, document.extdmform.catalogId.value,
document.extdmform.emailAddress.value)">Media Folders</a>
<br>
<a href="javascript:go_to_cart(document.extdmform.wcdmServer.value,
document.extdmform.logonId.value, document.extdmform.langId.value,
document.extdmform.storeId.value, document.extdmform.catalogId.value,
document.extdmform.emailAddress.value)">Product Cart</a>
<br>
<a href="javascript:view_order_history(document.extdmform.wcdmServer.value,
document.extdmform.logonId.value, document.extdmform.langId.value,
document.extdmform.storeId.value, document.extdmform.catalogId.value,
document.extdmform.emailAddress.value)">Order History</a>
<br>
<a href="javascript:add_to_cart(document.extdmform.wcdmServer.value,
document.extdmform.logonId.value, document.extdmform.langId.value,
document.extdmform.storeId.value, document.extdmform.catalogId.value,
adjust_asset_ids(get_asset_ids()), document.extdmform.emailAddress.value)">Add
To Cart</a>
<br>
<a href="javascript:add_to_media_folder(document.extdmform.wcdmServer.value,
document.extdmform.logonId.value, document.extdmform.langId.value,
document.extdmform.storeId.value, document.extdmform.catalogId.value,
adjust_asset_ids(get_asset_ids()), document.extdmform.emailAddress.value)">Add
To Media Folder</a>
<br>
// -->
<input type="button" value="View Product Cart"
 onClick="go_to_cart(document.extdmform.wcdmServer.value,
document.extdmform.logonId.value, document.extdmform.langId.value,
document.extdmform.storeId.value, document.extdmform.catalogId.value,
document.extdmform.emailAddress.value)">
<input type="button" value="View Media Folders"
onClick="view_media_folder_manager(document.extdmform.wcdmServer.value,
document.extdmform.logonId.value, document.extdmform.langId.value,
document.extdmform.storeId.value, document.extdmform.catalogId.value,
document.extdmform.emailAddress.value)">
<input type="button" value="View Order History"
onClick="view_order_history(document.extdmform.wcdmServer.value,
document.extdmform.logonId.value, document.extdmform.langId.value,
document.extdmform.storeId.value, document.extdmform.catalogId.value,
document.extdmform.emailAddress.value)">
<br>
<input type="button" value="Add To Cart"
onClick="add_to_cart(document.extdmform.wcdmServer.value,
document.extdmform.logonId.value, document.extdmform.langId.value,
document.extdmform.storeId.value, document.extdmform.catalogId.value,
adjust_asset_ids(get_asset_ids()), document.extdmform.emailAddress.value)">
<input type="button" value="Add To Media Folder"
onClick="add_to_media_folder(document.extdmform.wcdmServer.value,
document.extdmform.logonId.value, document.extdmform.langId.value,
document.extdmform.storeId.value, document.extdmform.catalogId.value,
adjust_asset_ids_thumbnail(get_asset_ids()), document.extdmform.emailAddress.value)">
</form>

<VTL_INCLUDE NAME="results_footer.tmpl">

<VTL_ELSE>

<p>
<img src="${images_uri_part}/spacer.gif" width="1" align="left">
No matching <VTL_IF NAME="search_type"
VALUE="clips">clips<VTL_ELSE>assets</VTL_IF> found.
</p>

</VTL_IF>

</font>
</body>
</html>
```

Appendix C. 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 publication 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 obligations 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 Corporation
Department TL3B/Bldg 503
PO Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195
USA

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

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

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Credit card images, trademarks, and trade names provided in this product should be used only by merchants authorized by the credit card mark's owner to accept payment via that credit card.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

- AIX
- DB2®
- DB2 Universal Database™
- IBM
- WebSphere

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

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

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

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

Glossary

document. An *item* that can be stored, retrieved, and exchanged among Content Manager systems and users as a separate unit. It can be any multimedia digital object. A single document can include varied types of content including, for example, text, images, and spreadsheets.

folder. A container used to organize *objects* which can be other folders or *documents*.

include (INC) files. A text file that contains declarations that are used by a group of functions, programs, or users.

item. In Content Manager, generic term for an instance of an *item type*. For example, an item might be a *folder*, document, video, or image. Generic term for the smallest unit of information that Enterprise Information Portal administers.

item type. A template for defining and later locating like items, consisting of a root component, zero or more child components, and a classification.

JavaServer Pages (JSP). A server-side scripting technology that enables you to embed Java code within static Web pages (HTML files) and to execute the Java code when the page is served.

library server. The component of a Content Manager system that stores, manages, and handles queries on *items*.

object. Any digital content that a user can store, retrieve, and manipulate as a single unit; for example, JPEG images, MP3 audio, AVI audio, and a text block from a book.

part. See *object*.

resource manager. The component of a Content Manager Version system that manages *objects*. These objects are referred to by *items* stored on the *library server*.

session bridge. A customizable feature in WebSphere Digital Media Enabler that enables the use of a content management system other than IBM Content Manager.

SAR file format. In WebSphere Commerce, SAR (store archive) is a platform-independent file format that aggregates many files into one. A store archive is a compressed file that contains all the assets (including file assets and database information) necessary to create a store. Publishing the store archive to a WebSphere Commerce server creates an operational store.

Index

A

apostrophes 8

C

content management 7, 43
contract pricing
  process 30
  setup 30
conventions
  boldface 2
  italic 2
  monospace 2
customization areas
  content management 5
  e-mail 5
  store data 5
  store front appearance 5

D

directory structures
  JSP 11
  proxies 11
  web objects 11

E

e-mail 7, 31
  configuring the delivery daemon 31
  customizing the daemon and templates 34

H

header, sidebar, and footer web page areas 19

I

include files 21
item type 7, 8, 45

J

JSP directory structure 11
jsp files 12, 20

L

locale 3

M

main body web page area
  customer-provided static pages 20
  inc files 20
main body web page area *(continued)*
  jsp files 20
media folder collaboration 7, 42

N

Notices 65
notification 31

P

pricing 6
  description 26
  tables 27
pricing, contract
  process 30
  setup 30
proxies directory structure 17
publications 1

S

sametime 42
session bridge 8, 45, 51, 52
setup
  e-mail 6, 23
  media folder collaboration 6
  pricing 6
  store data 23
  subscriptions 6
  user data 6
static pages 21
store front appearance 5, 17
subscriptions 7, 38
  amount 38
  buy items 42
  cancel sequence 40
  duration 38
  number 38
  purchase sequence 39
  renew sequence 41
  use cases 38

T

trademarks 66

U

user data 6, 23

W

web objects directory structure 15

Printed in USA