Updates to the Enhancements to IBM WebSphere Commerce Studio

Version 5.4

Migration Guide

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

First Edition (May 2003)

This edition applies to the following products:

- Enhancements to WebSphere Commerce Studio, Business Developer Edition for Windows NT and Windows 2000, Version 5.4 (Program 5724-A18)
- Enhancements to WebSphere Commerce Studio, Professional Developer Edition for Windows NT and Windows 2000, Version 5.4 (Program 5724-A18)

and to all subsequent releases and modifications of the above listed products, until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. You can send comments about this publication by one of the following methods:

1. Electronically to following web site:

   `http://www.ibm.com/software/commerce/rcf.html`

2. By mail to the following address:
   IBM Canada Ltd. Laboratory
   B3/KB7/8200/MKM
   8200 Warden Avenue
   Markham, Ontario, L6G 1C7
   Canada

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

Updates to the enhancements to IBM WebSphere Commerce Studio

Document description

This *Migration Guide* provides information regarding how to update an existing WebSphere Commerce Development environment within WebSphere Studio Applicaton Developer to the latest level. In particular, this document describes how to migrate a development environment that is using WebSphere Commerce 5.4.0.3 within WebSphere Studio Application Developer to the version 5.4.0.5 level of WebSphere Commerce .

Important: After migrating, the development environment will contain WebSphere Commerce code that is at either the FixPak 5.4.0.5 level, or the FixPak 5.4.0.5 level with the Commerce Enhancement Pack (October 2002). You must ensure that your target WebSphere Commerce Server is also running at this level of code before deploying any customized code from the development environment to the target WebSphere Commerce Server.

Conventions and terminology used in this book

This book uses the following highlighting conventions:

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

`Monospaced type` indicates examples of text you enter exactly as shown, as well as directory paths.

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

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

Windows indicates information that is specific to WebSphere Commerce for Windows NT(R) and Windows(R) 2000.

AIX indicates information that is specific to to WebSphere Commerce for AIX(R).

Solaris indicates information that is specific to to WebSphere Commerce for Solaris(TM) Operating Environment software.

400 indicates information specific to Commerce Suite, Pro Edition for the IBM@server(TM) iSeries(TM) 400 (R) (formerly called AS/400(R))

Professional indicates information specific to WebSphere Commerce Studio, Professional Developer Edition.

Business indicates information specific to WebSphere Commerce Studio, Business Developer Edition.

CEP indicates information that is specific to the Commerce Enhancement Pack.

DB2 indicates information specific to DB2 Universal Database(TM)

Oracle indicates information specific to Oracle.

This book uses the following terms:

development machine
The machine on which you are going to develop customized code for WebSphere Commerce. This is the machine onto which WebSphere Commerce Studio is installed.

target WebSphere Commerce Server
The machine to which you will deploy your customized WebSphere Commerce code.

Path variables

This guide uses the following variables to represent directory paths:

- *WAS_installdir*This is the installation directory for WebSphere Application Server. An example path is

  ```
  C:\WebSphere\AppServer
  ```

- *WCStudio_installdir*This is the installation directory for WebSphere Commerce Studio. An example path is

  ```
  C:\WebSphere\CommerceServerDev
  ```

- *enhancements_installdir*This is the installation directory for the enhancements to WebSphere Commerce Studio. An example path is

  ```
  C:\CommerceStudioEnhancments
  ```

- *WSAD_installdir*This is the installation directory for WebSphere Studio Application Developer. An example path is

  ```
  C:\Program Files\IBM\Application Developer
  ```

- *workspace_dir*This is the directory for the WebSphere Studio Application Developer workspace. An example path is

  ```
  C:\WCworkspace
  ```

  Important: Due to a limitation related to the number of characters in a path (it must be less than 256 characters), you should not accept the default workspace directory when opening WebSphere Studio Application Developer for the first time. If the default directory is accepted, errors may be encountered later when attempting to test WebSphere Commerce code in the WebSphere test environment.

Knowledge requirements

This book should be read by Store Developers that want to use the WebSphere Studio Application Developer, Version 5.0 product to perform programmatic customizations for WebSphere Commerce applications. Store Developers that are performing programmatic extensions should have knowledge in the following areas:

- Java
- Enterprise JavaBeans component architecture
- JavaServer Pages technology
- HTML
- Database technology
- WebSphere Studio Application Developer, Version 5.0

Where to find more information

For more information related to WebSphere Commerce, refer to the following Web sites:
Professional `http://www.ibm.com/software/webservers/commerce/wc_pe/ lit-tech-general.html`
Business `http://www.ibm.com/software/webservers/commerce/wc_be/ lit-tech-general.html`

Contents

Part 2. Code deployment

Part 3. Appendixes

Part 1. Installing and configuring the development

environment

This part of the document provides information related to setting up a WebSphere Commerce development environment that uses WebSphere Studio Application Developer Version 5.0 as the integrated development environment (IDE).

Chapter 1. Introduction

This document provides information on the following topics:
This document is provided to assist users of the

- Installing software and configuring the development environment on a new development machine.
- Migrating a development environment that is currently using WebSphere Commerce Version 5.4.0.3 to use version 5.4.0.5.
- Publishing stores within the WebSphere test environment component of WebSphere Studio Application Developer.
- Migrating customized code that was created using the WebSphere Commerce Version 5.4.0.3 workspace to the version 5.4.0.5 workspace.
- Deploying customized code from WebSphere Studio Application Developer to a target WebSphere Commerce Server.

This document refers to the *WebSphere Commerce Programmer's Guide* as a reference that provides details on the topic of code deployment. You should download the most recent copy of the *WebSphere Commerce Version 5.4 Programmer's Guide* from one of the following Web sites:
Professional `http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html`
Business `http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html`
On the preceding Web sites, click the link for user guides.

Note that once you have started to use WebSphere Studio Application Developer as your IDE, only certain parts of the current *WebSphere Commerce Programmer's Guide* remain applicable. In particular, the tutorials contained in the *WebSphere Commerce Programmer's Guide* no longer apply. They are written to introduce the WebSphere Commerce architecture and programming model within the context of VisualAge for Java, not within in the context of WebSphere Studio Application Developer.

In general, the first seven chapters of the *WebSphere Commerce Programmer's Guide* are still applicable, except for any VisualAge for Java specific step-by-step information. Also, most of the deployment information is still applicable, except for the steps about how to create JAR files. For more information about the new deployment process, refer to Chapter 10, "Deploying customized code" in this document.

This book may contain references to the WebSphere Commerce online help. If you do not have WebSphere Commerce installed, you can download the online help from the preceding listed Web sites.

Software and hardware prerequisites

This section describes the minimum software and hardware requirements that the development machine must meet. Additionally, the preferred configuration information provides a recommendation about the requirements for a more efficient development environment.

Minimum requirements

This section describes the minimum software and hardware requirements for your development machine.

- Pentium(R) III 700 MHz processor or higher
- 512 MB RAM minimum with at least 500 MB system swap
- Disk space requirements: 1 GB minimum
- Ensure that you have one of the following operating systems installed:
  - Windows 2000 Server Edition, or Advanced Server Edition with Service Pack 2 applied. You can obtain updates at the following Web site:

    ```
    http://www.microsoft.com
    ```

  - Windows NT Server Version 4.0 with Service Pack 6a installed on your development machine. You can obtain updates at the following Web site:

    ```
    http://www.microsoft.com
    ```

- Microsoft(R) Internet Explorer 5.5 or higher

Preferred configuration

This section describes software and hardware requirements to achieve a more efficient development environment.

- Pentium III 1 GHz processor or higher
- 1 GB RAM minimum with at least 1 GB system swap
- Disk space requirements: 5 GB minimum
- Ensure that you have one of the following operating systems installed:
  - Windows 2000 Server Edition, or Advanced Server Edition with Service Pack 2 applied. You can obtain updates at the following Web site:

    ```
    http://www.microsoft.com
    ```

  - Windows NT Server Version 4.0 with Service Pack 6a installed on your development machine. You can obtain updates at the following Web site:

    ```
    http://www.microsoft.com
    ```

- Microsoft Internet Explorer 5.5 or higher

Chapter 2. Installing prerequisite software

Before you can install the enhancements to WebSphere Commerce Studio, you must ensure that you have the following software installed:

- Database software:
  - DB2 DB2 Universal Database, Version 7.0 with FixPak 6. Ensure that you are using JDBC 2.

	To determine if you are using JDBC 2, do the following: 1. Navigate to the SQLLIB\java12 directory.

	2. If this directory contains an "inuse" file, open the file in a text editor to determine the level of JDBC that is currently in use. 3. If required, switch to JDBC 2, by doing the following: a. Ensure that no processes are currently using DB2$^{(R)}$ and that no DB2 process are running. b. Enter the following command: `usejdbc2`

  - Oracle Oracle 8.1.7.2.1
- WebSphere Studio Application Developer Version 5.0 general availability edition. Refer to the following sections for information about installing WebSphere Studio Application Developer.

Updating WebSphere Studio Application Developer

WebSphere Studio Application Developer early availability edition

If you are using WebSphere Studio Application Developer, Version 5.0, early availability edition, you must upgrade to WebSphere Studio Application Developer, Version 5.0 general availability.

After upgrading to WebSphere Studio Application Developer, Version 5.0 general availability, you must upgrade to the WebSphere Studio Application Developer PTF 001 - General Fixes.

For more information about how to apply these updates, refer to the following Web site:

```
http://www.ibm.com/support/docview.wss?rs=457&context=SSBRLP
   &q=&uid=swg24004419&loc =en_US&cs=utf-8&lang=en
```

Ensure that you read the install.html file for complete installation instructions.

WebSphere Studio Application Developer general availability edition

You must upgrade to the WebSphere Studio Application Developer PTF 001 - General Fixes.

For more information about how to apply these updates, refer to the following Web site:

```
http://www.ibm.com/support/docview.wss?rs=457&context=SSBRLP
   &q=&uid=swg24004419&loc =en_US&cs=utf-8&lang=en
```

Ensure that you read the install.html file for complete installation instructions.

Chapter 3. Updating the enhancements

You have two options for updating your develpment environment. These are as follows:

a. You can migrate your existing development environment inplace. This means that your existing workspace inside of WebSphere Studio Application Developer will be upgraded to the version 5.4.0.5 level. Also, your existing development database will be updated.   Code customizations that are in projects separate from the WebSphere Commerce projects will be untouched. If you have code customizations in the existing WebSphere Commerce projects (for example, if you have modified a WebSphere Commerce public entity bean), you must merge the changes into the updated workspace. For instructions on how to migrate an existing instance, refer to the "Migrating the WebSphere Commerce development instance" section. **Backup the existing workspace by making a copy of it in the file system before you start the process of migrating to the new WebSphere Commerce workspace.**

b. Create a new development environment at the version 5.4.05 level. In this case, a new development instance, development database and workspace are created at the version 5.4.0.5 level. The original development environment at the version 5.4.0.3 level remains untouched. If you take this approach, you must then manually migrate any database modifications or code modifications from the original development environment to the new environment. Refer to the "Creating WebSphere Commerce development instance" section for instructions on how to do this. **Backup any existing workspace by making a copy of it in the file system before you start the process of creating a new WebSphere Commerce workspace**

This migration process consists of the following high-level steps:

1. Installing the updates to the enhancements.
2. Creating a new WebSphere Commerce development instance or migrating an existing WebSphere Commerce development instance

Each of these steps is described in more detail in subsequent sections.

Depending on your business requirements, you should have downloaded one of the following updates to the enhancement packages and saved it on your development machine:

- CEP TLKT_BE_OCT02_CommerceEnhancementPack_EI_Migration.zip Use this package if you are migrationg code to WebSphere Commerce Business Edition, at the FixPak 5.4.0.5 level with the Commerce Enhancement Pack (October 2002)
- CEP TLKT_PRO_OCT02_CommerceEnhancementPack_EI_Migration.zipUse this package if you are migrating code to WebSphere Commerce Professional Edition, at the FixPak 5.4.0.5 level with the Commerce Enhancement Pack (October 2002)

For information about the download location for the preceding files, refer to one of the following Web sites:
Professional `http://www.ibm.com/software/webservers/commerce/wc_pe/support.html`
Business `http://www.ibm.com/software/webservers/commerce/wc_be/support.html`

Important: You can only install one version of the enhancements on your development machine. For information about determining which version of the enhancements is installed, refer to "Determining which version of the enhancements is installed".

Installing the updates to the enhancements

After you have downloaded the updates to the enhancements package to your development machine, do the following:

1. At a command prompt, navigate to the directory into which you downloaded the updates to the enhancements package.
2. Unzip the package into a temporary directory. For example, unzip the package into C:\temp.
3. Navigate to this temporary directory and launch the setup.exe program. An installation wizard opens to guide you through the install.
4. Select the language for your installation and click **OK** to continue.

5. A Welcome window opens. Click **Next** to continue.
6. The License Agreement window opens. Review the terms of the license agreement and either accept or decline the agreement. If you accept the terms of the license agreement, the installation program will continue. If you decline the license agreement, the installation program will exit.
7. You are prompted to specify the directory to which the enhancements will be installed. Either accept the default location or into the **Directory Name** field, enter a new directory. Click **Next** to continue.
8. Summary information displaying the selected installation directory and the required disk space is displayed. Click **Next** to install the enhancements.
9. Progress of the installation process is displayed. When the installation is complete, click **Finish**.

Migrating a WebSphere Commerce development instance

1. Close WebSphere Studio Application Developer.
2. At a command prompt, navigate to the following directory: `enhancements_installdir\bin`
3. Enter the following command: WorkspaceUpdate `workspace_dir` where `workspace_dir` is the directory for WebSphere Commerce workspace. The following is an example of this command with sample values provided:

   ```
   WorkspaceUpdate C:\WCworkspace
   ```

   The WorkspaceUpdate script will migrate your workspace to the WC 5.4.0.5 toolkit level. The script will take approximately 10 to 15 minutes to run. The process has finished when "End of script" is displayed in the Command prompt Window.

4. You must run the updateInstance.bat command for each instance that was created.
   Do the following:
   (1) At a command prompt,navigate to the following directory: enhancements_installdir\bin
   (2) Enter the following command: updateInstance instancename.
   The following is an example of this command with sample values provided:

   updateInstance wcdev

   The updateInstance script will migrate your instance xml to the WC 5.4.0.5 toolkit level. The process has finished when "End of script" is displayed in the command prompt window

   > If you receive an error indicating that the specified workspace is incorrect, do the following:
   >
   > 1. At a command prompt, navigate to the *WSAD_installdir* directory.
   > 2. Enter the following command:
   >
   >    ```
   >    wsappdev -setworkspace
   >    ```
   >
   > 3. Open WebSphere Studio Application Developer and select *workspace_dir* as the location of the workspace.
   > 4. Close WebSphere Studio Application Developer and rerun the WorkspaceUpdate script. command.

5. Open WebSphere Studio Application Developer
6. After migrating the instance, you will need to repair the WebSphereCommerceServer configuration. To

repair the WebSphereCommerceServer configuration, do the following:
   a. Switch to the Server perspective
   b. Double-click WebSphereCommerceServer
   c. When a dialog request confirmation to repair the server configuration, click yes
7. You must then update the Java build path for your EJB project, as follows:
   a. For each EJB project that you have created, you must right-click the project and select **Properties**
   b. In the left navigation pane, select **Java Build Path**.
   c. Click the **Projects** tab. Select every project that contains code that you reference. In particular, ensure that you select **wcs.ccore.ejbs**. Click **OK** then close the properties editor.

The next step is to start WebSphere Studio Application Developer and publish a store using Store Services. See Chapter 4, "Publishing stores using Store Services" for more information.

Creating a WebSphere Commerce development instance

Note that you can have multiple workspaces to support various development projects. If you want to do this, you must create a separate instance to correspond to each workspace and use unique names for the both the instance and its database. If you are recreating an instance using an existing instance name, ensure that you perform the instance cleanup steps before recreating the instance (refer to "Removing an old instance" for more information).

Each development instance requires approximately 400MB of free space on your hard drive.

When you create your development instance, according to the instructions contained in this section the following tasks are accomplished:

1. Workspace preferences are set.
2. The WebSphere Commerce development instance is created.
3. The local WebSphere Commerce development database is created.
4. The WebSphere Commerce workspace is imported into WebSphere Studio Application Developer.

To create your development instance, do the following:

1. Start WebSphere Studio Application Developer.

   Important: You must change the default workspace directory. When WebSphere Studio Application Developer is started for the first time, you are prompted to select a directory for your workspace. By default, the default workspace directory is

   ```
   C:\Documents and Settings\win_user\My Documents\IBM\wsad\workspace
   ```

   where `win_user` is the Windows user ID with which you are currently logged on. You *must not* accept these settings. Change the directory path to a path that is much shorter, for example, set it to the following:

   ```
   C:\WCworkspace
   ```

   The preceding directory is referred to as the `workspace_dir` directory in this document.
   Note:
      If you are prompted to select your workspace, select *workspace_dir*. If you are not prompted, you

can start the application and use the correct workspace, by doing the following:

   a. At a command prompt, navigate to the *WSAD_installdir* directory.
   b. Enter the following command:

```
wsappdev -data workspace_dir -showlocation
```

   c. As another option, you can enter the following command so that you will be prompted for your workspace location:

```
wsappdev -setworkspace
```

2. Set your workspace preferences, by doing the following:
   a. From the **Window** menu, select **Preferences**.The Preferences window opens.
   b. Click **Import**. When prompted, select the preferences file that is found as follows:

```
enhancements_installdir\toolkit\workspace\wsad.epf
```

   Click **OK**.
   c. From the **Window** menu, select **Preferences** again.
   d. In the left pane, select **Workbench**, then in the right pane, clear the **Perform build automatically on resource modification** check box.
   e. In the left pane, select **Validation**, then in the right pane, clear the following check boxes
      - **HTML Syntax Validator**
      - **JSP Compilation Validator**
      - **WAR Validator**

   and click **OK**.
3. Close WebSphere Studio Application Developer.
4. At a command prompt, navigate to the following directory:

```
enhancements_installdir\bin
```

5. Enter the following command:

```
createInstance workspace_dir instanceName dbName dbUser dbPassword dbType
```

where
   - *workspace_dir* is the directory for WebSphere Commerce workspace.
   - *instanceName* is the name of your WebSphere Commerce development instance.
   - *dbName* is the name of your development database.
   - *dbUser* is the database user.
   - *dbPassword* is the password of your database user.
   - *dbType* is the type of database you are using for your development database. Valid options are `DB2` and `Oracle`.

The following is an example of this command with sample values provided:

```
createInstance C:\WCworkspace wcdev wcdev db2admin db2admin DB2
```

The createInstance script creates your instance and imports and configures your workspace. Expect this step to take at least 30 minutes to complete. You can follow its progress in the *enhancements_installdir*\logs\createInstance.log file. The process has finished when "Instance creation has completed." is displayed in the log file.

	If you receive an error indicating that the specified workspace is incorrect, do the

following:

1. At a command prompt, navigate to the *WSAD_installdir* directory.
2. Enter the following command:

   ```
   wsappdev -setworkspace
   ```

3. Open WebSphere Studio Application Developer and select *workspace_dir* as the location of the workspace.
4. Close WebSphere Studio Application Developer and rerun the createInstance command.

The next step is to start WebSphere Studio Application Developer and publish a store using Store Services. Refer to Chapter 4, "Publishing stores using Store Services" for more information.

Removing an old instance

To remove an old instance, do the following:

1. Drop the development database that was specified when you ran the createInstance command.
2. Delete the instance directory. This directory is called:

   ```
   enhancements_installdir\instances\instanceName
   ```

   where *instanceName* is the name of the instance you are removing.
3. Restore the *enhancements_installdir*\instances\wcs_instances.bak file to *enhancements_installdir*instances\wcs_instances. The result will be a file that has no entries after the [instance] tag.
4. Delete your workspace directory.

Chapter 4. Publishing stores using Store Services

This chapter provides information specific to publishing WebSphere Commerce stores, using Store Services running within the WebSphere test environment component of WebSphere Studio Application Developer.

Publishing a store

CEP **Important:** The recommended sample store for viewing business-to-consumer (B2C) functions is the new FashionFlow sample store. This sample store was introduced in the Commerce Enhancement Pack (October 2002). FashionFlow merges the features of all of the WebSphere Commerce B2C sample stores (WebFashion, NewFashion, InFashion, and WebAuction). It also incorporates some new features. For more information about the FashionFlow sample store, refer to the *FashionFlow Sample Store Guide*.

Store Services running within the WebSphere test environment is used to publish WebSphere Commerce stores that are used in development.

Publishing tips When publishing stores using Store Services running in the WebSphere

	test environment, keep the following points in mind: 1. Business If you want to publish both the FashionFlow store and the NewFashion store, they must be published under different store owners. For example, publish FashionFlow under the Seller organization and NewFashion under the Default organization. 2. Professional FashionFlow and NewFashion cannot coexist. Use FashionFlow, as it contains all of the functionality of NewFashion, as well as additional functionality.

The following steps describe how to publish a WebSphere Commerce sample store in the WebSphere test environment:

1. Open WebSphere Studio Application Developer (**Start > Programs > IBM WebSphere Studio > Application Developer 5.0**).
   Note:
   If you are prompted to select your workspace, select *workspace_dir*. If you are not prompted, refer to "Changing workspaces" for more information.
2. Open the Server perspective (**Window > Open Perspective >(Other >) Server**)
3. Right-click the **WebSphereCommerceServer** server and select **Start**.
4. Open a Web browser, and then open Store Services by entering the following URL:

   ```
   http://localhost/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=devtools.Logon
   ```

5. Business (Optional) Create an organization or organizational unit for the site, and then create users, as described in the WebSphere Commerce online help. Note, if you need to launch the Administration Console, use the following URL:

   ```
   http://localhost/webapp/wcs/tools/servlet/ToolsLogon?
      XMLFile=adminconsole.AdminConsoleLogon
   ```

6. Logon to Store Services. You can logon using the wcsadmin user ID (the initial password is wcsadmin, however it should be changed the first time it is used), the user ID of another site administrator, or the user ID of a store administrator that was created in step 5.
7. Create a new store archive, based upon one of the sample store archives, as described in the WebSphere Commerce online help.
8. From the **Store Archive** list in the Store Services pane, select the store archive you created in step 7 and click **Publish**.
9. Select all of the publishing options. Ensure that the path values are specified as follows:
   - Web application document root

     ```
     workspace_dir\wcstores\Web content
     ```

   - Application properties path

     ```
     workspace_dir\wcstores\Web content\WEB-INF\classes
     ```

   where `workspace_dir` is the location of your workspace directory.
10. When ready to publish, click **OK**.
11. While the store is being published, the browser returns to the Store Archive list page. On this page, the publishing state is reflected in the Publish status column. Click **Refresh** to update the status.
    Note:
    The logging for the Loader package has been configured so that trace information is not written, by default. This logging can be configured in the `WCALoggerConfig.xml` file, but turning the trace on

will negatively impact performance of the publishing process.

12. When publishing has completed, select your store archive from the list and click **Publish Summary** to view the results of the publishing process. If an error occurred during the publish, a window displays more information about the error. For more information, refer to the "Troubleshooting publishing" topic in the WebSphere Commerce online help.
13. After a successful store publish, use the Navigator View in WebSphere Studio Application Developer to browse the **wcstores** Web project. Expand the **wcstores** Web project, then right-click the **Web Content** subfolder and select **Refresh**. The JSP templates, HTML files and other Web assets should now be available in the following folder:

    ```
    wcstores\Web Content\storeDir
    ```

    where *storeDir* is the name of your new store.The properties files for the store should now be available in the following directory:

    ```
    wcstores\Web Content\WEB-INF\classes\storeDir
    ```

    If rebuilding the `wcstores` Web module, the `classes` directory is emptied and created from the `source` folder. As a result, you should copy *`storeDir`* to the `source` folder.

14. Launch your store using one of the following methods:
    a. Using the **Launch store** button in Store Services.
    b. By entering the following URL in a Web browser:

    ```
    http://localhost/webapp/wcs/stores/servlet/storeDir/index.jsp
    ```

    where *storeDir* is the directory name for your store.

Chapter 5. Understanding payment options

When running stores within the WebSphere test environment component of WebSphere Studio Application Developer, you have different options for processing payments. If you are using a business-to-consumer store, you can either use a test payment method, or call out to a remote Payment Manager. The test payment method is only to be used for testing purposes, and was only designed to allow you to complete a purchase. By default, the test payment method is configured when you create your WebSphere Commerce development instance.

The test payment method cannot confirm that payment was accepted, cancel payments, deposit funds, or perform refunds. If you wish to perform these actions you must install WebSphere Payment Manager on a remote machine.

If you are using a business-to-business store, you can either use the Credit Line payment option, or call out to a remote payment manager.

For business-to-consumer stores, this chapter provides information about how to switch from using the test payment method to using a remote Payment Manager.

For business-to-business stores, this chapter provides information about how to enable the Credit Line payment option for use in the development environment.

Using Credit Line in a business-to-business store

This section provides instructions to use the Credit Line payment option in your business-to-business store for use in your development environment. Note that these instructions are provided so that you can easily complete a purchase within the development environment. For information on using this payment option outside of the development environment, ensure that you refer to the WebSphere Commerce online help.

To use the Credit Line payment method within your development environment, do the following:

1. Publish a store based upon the ToolTech sample store.
2. Create a new registered user. Be sure to specify the following options when creating this user:
    a. In the **Buyer organization** field, enter `Buyer organization b`.
    b. Fill in the other mandatory fields with appropriate values for your testing scenario.
    c. Click **Submit**.
3. Launch the Adminstration Console and logon as a site administrator. (Refer to "Links to WebSphere Commerce tools" for more information about accessing this tool.)
4. From the **Approvals** menu, select **Approve requests**.
5. Select the user you created in step 2. Click **Approve**, then click **OK**.
6. Logon to your business-to-business store as the user created in step 2. Browse the catalog and select a product. When selecting the product, select one that uses the "4567" contract.
7. When completing the purchase, select **Credit Line** as the payment method.

Using a remote Payment Manager with the WebSphere test environment

If the test payment method is insufficient for your development needs, you can configure your development environment to work with a remote Payment Manager. In doing so, orders placed in your stores running in the WebSphere test environment component of WebSphere Studio Application Developer can be processed.

Note that once you configure your development environment to use a remote Payment Manager, all stores that you have will use the remote Payment Manager. If you have previously configured your development environment to use the test payment method, the test payment method will be ignored once you have configured the development environment to use the remote Payment Manager.

Configuring your development environment to use a remote Payment Manager involves the following steps:

1. Completing your WebSphere Studio Application Developer configuration steps. You must be able to publish (as described in Chapter 4, Publishing stores using Store Services) and launch your store within the WebSphere test environment before continuing to the next step.
2. Configuring Payment Manager to use the WCSRealm.
3. Restarting the WebSphere test environment.

Configuring your development environment to use a remote Payment Manager

This section describes the step-by-step details for configuring your development environment to use a remote Payment Manager.

Preparation

It is assumed that you have successfully installed and configured Payment Manager on a machine separate from your development machine. For complete details on installing and configuring Payment Manager, refer to the *IBM WebSphere Payment Manager for Multiplatforms Installation Guide*.

Configuring your WebSphere Commerce Studio instance to use a remote Payment Manager

To configure your instance to use a remote Payment Manager, do the following on your development machine:

1. Open the *enhancements_installdir*`\instances\`*`instance_name`*`\xml\` *`instance_name`*`.xml` file in a text editor, and do the following:
    a. Search for the `PaymentManager` element within the file, and edit the following attributes:
        - UseNonSSLPMClient="1"
        - Hostname="*host_name*"
        - WebServerPort="*port_number*"
        - UseExternalPM="false"
        - ProfilePath="*profile_path*"
        - If you are using a SOCKS server, do the following:
          Note:
          If you do not know if you are using a SOCKS server, please consult your network administrator.
            - Set UseSocksServer="true"
            - Add SocksHostname="*socks_server_hostname*"
            - Add SocksPort="*socks_port*"

    where
        - *host_name* is the fully qualified host name of the Payment Manager machine.
        - *profile_path* is the full path name of the directory where the standard WebSphere Commerce Payment Manager Cashier Profiles are located. By default, this should be:

          ```
          enhancements_installdir\instances\instance_name\
          xml\payment
          ```

        - *port_number* is the Web server port that Payment Manager uses for non-SSL transactions. The default port number is 80.
        - *socks_server_hostname* is the fully qualified host name of your SOCKS server.
        - *socks_port* is your SOCKS port number.

Restart the servlet instance.

Configuring Payment Manager to use the WCSRealm

To use the WCSRealm on a remote machine where Payment Manager is installed, you must manually configure Payment Manager through the Administration Console. To manually configure your system to use WCSRealm, do the following:

1. Open the WebSphere Application Server Administration Console on the Payment Manager machine.
2. Navigate to and access the WebSphere Payment Manager application server by doing the following:
    a. Expand the WebSphere Administrative Domain.
    b. Expand Nodes.
    c. Expand the *host_name*.
    d. Expand **Application Servers**.
    e. Select WebSphere Payment Manager. Click the **Stop** button to stop the WebSphere Payment Manager application server.
3. After the WebSphere Commerce Payments application server has stopped, select the **JVM Settings** tab

page of the application server. Scroll down to the **System Properties** box, select the `wpm.RealmClass` system property, and do the following:

   - Replace the `com.ibm.etill.framework.payserverapi.PSDefaultRealm` value with `com.ibm.commerce.payment.realm.WCSRealm`

   Click **Apply**.

4. If Payment Manager is enabled for SSL, you must also use the WebSphere Application Server administrative console to add *:443 entries to the default host window under Virtual Hosts
   a. Under the WebSphere Administrative Domain, click Virtual Hosts
   b. Select the General tab for the default host
   c. Add *:443 as a Host Alias.
5. In the Payment Manager install directory, open the `WCSRealm.properties` file to change the value of the `WCSHostName` property to the fully-qualified host name of the WebSphere Commerce Studio machine with the WebSphere test environment instance. In addition, while editing the `WCSRealm.properties` file, add the following lines to the bottom of the file:

   ```
   WCSWebServerPort=webserverPortNumber
   UseNonSSLWCSClient=1
   ```

   where *webserverPortNumber* is the HTTP transport that is used by the WebSphere Commerce development instance running in WebSphere Studio Application Developer. To verify this value, do the following:
   a. Open WebSphere Studio Application Developer and switch to the Server perspective.
   b. Double-click the **WebSphereCommerceServer** server.
   c. Click the Ports tab and verify the HTTP transport number..

   Save the `WCSRealm.properties` file.
6. Start the Payment Manager Application Server, within the WebSphere Application Server Administration Console.
   If you added *:443 entries to the default host window, you also need to stop and restart WebSphere Application Server.
7. Start Payment Manager:
   a. Open a command prompt.
   b. Change to the Payment Manager install directory.
   c. Run the following command: `IBMPayServer`.

   You will be prompted to enter your Payment Manager password, which is the password of the user you specified when connecting to the Payment Manager database.
   Note:
      Once these updates to your Payment Manager machine are complete, you may only use Payment Manager when WebSphere Commerce is running under the WebSphere test environment. Payment Manager relies on WebSphere Commerce to authenticate the Payment Manager administrator.

Creating a merchant, brand, and account in Payment Manager

Before using the remote Payment Manager with a WebSphere Commerce store running in the WebSphere test environment, you must create a merchant, brand, and account on the Payment Manager server. To configure your Payment Manager in this manner, do the following:

1. Ensure that WebSphere Commerce is running under the WebSphere test environment.
2. Log in to Payment Manager using the `wcsadmin` user ID and the password. You can access the Payment Manager log in page by entering the following URL in a browser:

   ```
   http://host_name/webapp/PaymentManager
   ```

3. From the left hand panel, click **Merchant Settings**. In the right hand window, click **Add a Merchant**.

4. Enter a Merchant Name for your sample store. The Merchant Number must be the same as the storeId for your store (e.g. 10001). Check the **OfflineCard** check box and click **Create Merchant**. A "Merchant created successfully" message should appear.
5. In the left hand panel, click Merchant Settings again. In the right hand window, beside the name of your new merchant, the green cassette status icon (under the OfflineCard column) indicates that the OfflineCard cassette is enabled and running for that merchant. Click on the green icon. You must now create an account for each currency supported by your store, by clicking on **Accounts**.
6. Click **Add an Account** Enter an Account Name and an Account Number. The account number must be unique within that store. Select the appropriate currency from the drop down list. Click **Create Account**.
7. In the Accounts window, click on the name of your new account. Then click on **Brands**
8. Click on **Add a Brand** Enter a Brand Name (e.g. VISA) and click **Create Brand**.
9. Repeat for all other Brands to be used with this account. Repeat the above three steps to add accounts for all other currencies to be supported by your store.

Removing orders that used the TestPaymentMethod

To use a remote Payment Manager, you must remove any orders that were placed using the TestPaymentMethod. To remove these orders, do the following:

- Open a DB2 or Oracle session and connect to your instance database. Enter the following statements (Oracle users must include a semicolon at the end of each command):

  ```
  delete from orders where orders_id in ( select orders_id
  from ordpaymthd where policy_id in (select policy_id from
  policy where policyname='TestPaymentMethod')) and storeent_id = your_storeent_id

  delete from policy where policyname='TestPaymentMethod'
  ```

  where *your_storeent_id* is the unique identifier for the store in which the orders were placed.Oracle users must enter the following command after running these SQL statements:

  ```
  commit;
  ```

Verifying your remote Payment Manager

To verify that your store can successfully use the remote Payment Manager, you must launch your store and attempt to complete a purchase in that store. As such, you should do the following:

1. Ensure that your server instance for WebSphere Commerce is running.
2. Launch your store in a browser. For example, you might enter a Web address similar to the following:

   ```
   http://localhost/webapp/wcs/stores/servlet/storeDir/index.jsp
   ```

   where *storeDir* is the name of your store.
3. To place an order using Payment Manager, do the following when providing credit card information:
   a. From the **Credit card type** drop-down list, select the brand name that you have entered on step number 8. For example, **VISA**.
   b. In the **Card number** field, enter a 16 digit number. For example, 4111111111111111 (4 followed by 15 ones).
   c. Set the expiration month and year to a future date.

Chapter 6. Developing code for iSeries

If your target deployment WebSphere Commerce Server is running on iSeries, specific steps need to be taken in order to be able to deploy your customized code to that platform. This chapter provides information about these steps.

Development environment for iSeries

In short, a special development environment is not required in order to create customized code for iSeries. The development workstation is setup following the same procedure that is outlined in Chapter 2, "Installing prerequisite software" and in Chapter 3, "Installing the enhancements". The local development database used should be DB2. Using this configuration, customized code can be created and tested using the local DB2 database and local WebSphereCommerceServer test server.

After testing determines that the customized code functions to your satisfaction within the context of the test server, you must then deploy it to a target WebSphere Commerce Server running on the iSeries platform. In order to account for differences between the database on the Windows and iSeries platforms, a WebSphere Commerce enterprise bean conversion tool is provided. Steps on how to use this tool are provided in the next section.

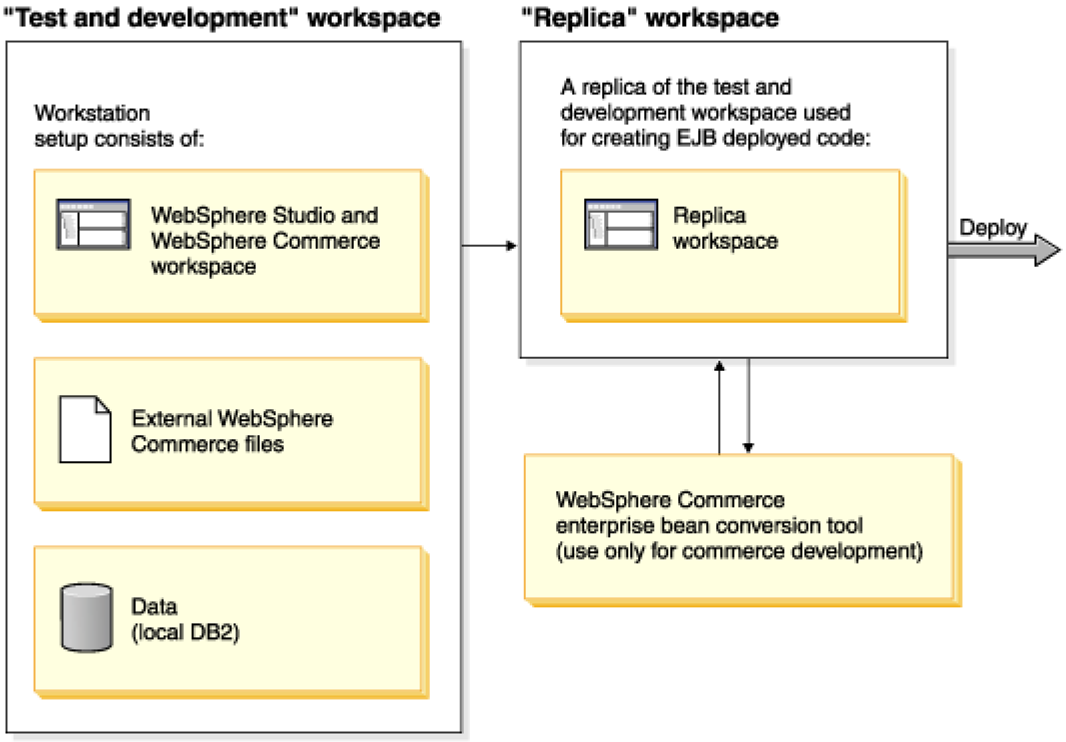
Overview of the WebSphere Commerce enterprise bean conversion tool

At a high-level, using the conversion tool involves the following steps:

1. On the development machine, create code in the test and development workspace until the code functions as desired.
2. Using the file system, create a replica of the test and development workspace.
3. Run the WebSphere Commerce enterprise bean conversion tool.
4. Import the output from the conversion tool into the replica workspace.
5. Open WebSphere Studio Application Developer using the replica workspace and generate deployed code for any EJB groups that you have modified.
6. Deploy the customized code to your target WebSphere Commerce Server running on iSeries.

These steps are shown in the following diagram:

Figure 1.


"Test and development" workspace
"Replica" workspace
Workstation setup consists of:
WebSphere Studio and WebSphere Commerce workspace
External WebSphere Commerce files
Data (local DB2)
A replica of the test and development workspace used for creating EJB deployed code:
Replica workspace
Deploy
WebSphere Commerce enterprise bean conversion tool (use only for commerce development)

Using the WebSphere Commerce enterprise bean conversion tool

To use this conversion tool, do the following:

1. Create a copy of your test and development workspace directory. For example, make a copy of the entire C:\WCworkspace directory, and call it C:\tempWorkspace.
2. Using a command prompt, navigate to your test and development workspace directory (for example, navigate to C:\WCworkspace).
3. Temporarily set the class path to include the wcsconvert.jar JAR file, by entering the following command:

   ```
   set CLASSPATH=%CLASSPATH%;enhancements_installdir\toolkit\
      tools\wcsconvert.jar;
   ```

   Note, the line break in the preceding class path is for presentation purposes only.
4. Run the meta data conversion program, by entering the following command:

   ```
   WSAD_installdir\eclipse\jre\bin\java
   com.ibm.commerce.metadata.conversion.ConvertMetaData  "workspace_dir"
   "iSeriesPropMapFile" "outputDir"
   ```

   where
    - *workspace_dir* is the directory of your test and development workspace
    - `iSeriesPropMapFile` is the iSeries property map file
    - `outputDir` is the directory where the converted meta data is placed.

   The following is an example of the command with values filled in (line breaks are for presentation purposes only):

   ```
   C:\Program Files\IBM\WebSphere Studio\eclipse\jre\bin\java
   com.ibm.commerce.metadata.conversion.ConvertMetaData "C:\WCworkspace"
   "C:\CommerceStudioEnhancements\toolkit\tools\as400.mapping"
   ```

```
"C:\mappingOut"
```

5. Once the conversion tool has finished, navigate to the directory you specified where the converted meta data should be placed. For example, navigate to C:\mappingOut.
6. From this output directory, copy the files corresponding the EJB groups that you modified into the replica workspace (for example, C:\tempWorkspace). Overwrite the existing meta data.
7. Open WebSphere Studio Application Developer using the replica workspace.
   Note:
   > If you are prompted to select your workspace, select the directory for your replica workspace (for example, C:\tempWorkspace). If you are not prompted, refer to Changing workspaces for more information.
8. For each EJB group that contains modified code, you must generate deployed code, as follows:
   a. From the **Window** menu, select **Show View > Navigator**.
   b. Navigate to the first EJB group that contains modified code. Right-click on this group and select **Generate Deployed and RMIC code**.
   c. In the Generate Deployed and RMIC Code window, select all of the enterprise beans and click **Finish** to generate the code.
   d. Repeat these steps for each EJB group that contains modified code.

- Once the deployed code is generated, you can follow the regular instructions (refer to Chapter 10, "Deploying customized code") to deploy the customized code to your target WebSphere Commerce Server. Additionally, after the code has been deployed, you can remove the replica workspace from your workstation, if desired.

Part 2. Code deployment

This part describes how to deploy customized WebSphere Commerce code from the WebSphere Studio Application Developer development environment to a target WebSphere Commerce Server.

Chapter 7. Deploying customized code

Overview of code deployment

When customizing your e-commerce application, you may do any of the following:

- Create new commands, data beans or entity beans
- Extend existing WebSphere Commerce entity beans
- Modify the logic of existing commands or data beans

When developing code within WebSphere Studio Application Developer, you can test your code within the WebSphere test environment. At some point, you must deploy your code to a WebSphere Commerce Server outside of the development environment. Note that the level of code on your *target WebSphere Commerce Server* must be the same as the level of code on the development machine.

In the following sections, *target WebSphere Commerce Server* refers to the WebSphere Commerce Server to which you are deploying the customized code. In some testing scenarios, you may deploy to a WebSphere Commerce Server that is running on the same machine as WebSphere Studio Application Developer. In other situations, the target WebSphere Commerce Server is on another machine, and may even be running on a different platform.

The following sections describe the high-level steps for deploying the various types of customized code. Use them to understand the steps involved in the deployment process but refer to "Appendix B Deployment details" of the *WebSphere Commerce Programmer's Guide* for step-by-step deployment instructions.

In addition to the following sections that describe deployment of customized code, if you have created new access control policies in your development environment, the same access control policies must be created on the target WebSphere Commerce Server. Refer to the "Loading the access control policies for the new resources" topic starting on page 272 in the tutorials contained in the *Programmer's Guide* for an example of this procedure.

All references to particular pages in the *Programmer's Guide* refer to the version of the document with the following version information in the Edition notices (inside the front cover):Second Edition (June 2002, Revision 1 released in September 2002)

Alternative approach to deployment

In the future, an alternative and more automated approach to deployment of customized code will be provided. Check the WebSphere Commerce Web site in the future for more information.

Deployment of new commands and data beans

When creating new commands and data beans, you should place them in a package that is named appropriately for your application. For example, you could create a new package by the name of `com.mycompany.mycommands` in which you keep your new commands. This package must be stored within the stores web project, `wcstores`. Careful organization of code is required, to ensure that deployment is successful.

Note that the major difference when deploying new commands and data beans from WebSphere Studio Application Developer instead of from VisualAge for Java is the fact that a `.class` file is now used instead of a JAR file. As such, the steps that were previously described about repackaging JAR files are no longer applicable. Additionally, when you copy the `.class` file to the appropriate directory on the target machine, you must also maintain the correct directory structure to represent the package naming convention for the `.class` file.

With all of your new commands and data beans stored in the Java source directory of the wcstores Web project, deployment consists of the following high-level steps:

1. From the development machine, copy the `.class` file that contains the new command and data bean code into the following directory on the target WebSphere Commerce Server.

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

   As an example, if you had `mycommand.class` class file, you might do the following:
   a. Locate the `mycommand.class` class file on your development machine:

      ```
      workspace_dir\myproject\bin\com\mycompany\mycomponent\mycommand.class
      ```

   b. Copy this class file into the following directory on the target WebSphere Commerce Server:

      ```
      drive:\WebSphere\AppServer\installedApps\
      ```

```
WC_Enterprise_App_instanceName.ear\wcstores.war\
WEB-INF\classes\com\mycompany\mycomponent\mycommand.class
```

2. Register the new command in the command registry on the target WebSphere Commerce Server. Refer to the "Command registration framework"section on page 26 of the *Programmer's Guide* for more information.
3. Stop and restart the WebSphere Commerce enterprise application, using the WebSphere Application Server Administrator's Console. Refer to the appropriate *WebSphere Commerce Installation Guide* for more information about starting and stopping this application.

Deployment of new entity beans

When creating new entity beans, you must create them in your own EJB project that is separate from the EJB project that contains the WebSphere Commerce entity beans. You must also ensure that this project does not contain the code for any commands or data beans. For example, you could create a new EJB project called `MyEntityBeansProject`. If the project contains code other than the code for new entity beans, deployment may not be successful.

Once you are satisfied with the way your new entity beans function within the WebSphere test environment, you must deploy them to a target WebSphere Commerce Server.

In general, the deployment steps are very similar to those described in the *Programmer's Guide*, except for the following:

1. Instead of creating the EJB 1.1 JAR file, you now create a JAR file that includes the EJB deployed code. As such, you must follow the steps contained in this document to create the EJB JAR file, instead of the ones in the *Programmer's Guide* to create the EJB 1.1 JAR file.
2. Since the new EJB JAR file contains deployed code, you no longer have to perform the "Generating deployed code" section as described in the *Programmer's Guide*.
3. Creating the implementation JAR file for your own new entity beans is no longer required. As such, when you refer to the *Programmer's Guide*, ignore any steps related to the implementation JAR file for your own new entity beans.
4. Repackaging the JAR file, as described in the *Programmer's Guide* is no longer required.

The following information provides an overview of the deployment steps:

1. On the development machine, use WebSphere Studio Application Developer to create a new EJB JAR file for your new EJB project. To create this JAR file, do the following:
   a. In the J2EE perspective, highlight the EJB project that contains your new entity beans.
   b. From the **File** menu, select **Export**.
   c. Select **EJB JAR File** as the export destination and click **Next**.
   d. From the **What resources do you want to export?** drop-down list, select the EJB project that you want to export.
   e. Click **Browse** to select the directory into which the JAR file should be placed. In the **File name** field, enter a name for the JAR file and click **Open**. All of the resources in the Java output folder `bin` are included in the JAR file.

      Note:

      If this EJB project had been previously imported and the imported JAR file is specified on the class path, the contents of the specified JAR file will be merged back into the exported JAR. If a compiled class exists in the project, it will not be merged in from the imported JAR. As a result, if a Java class was deleted from the current project, during the export the class will be merged back into the new JAR file because it is contained in the `imported_classes` folder.

   f. Specify additional options as required, using the **Options** check boxes.
   g. Click **Finish**.

2. Copy the ***`EJBJarFile`***`.jar` JAR file into the following directory on the target WebSphere Commerce Server:

   `drive``:\WebSphere\CommerceServer\temp`

3. On the target WebSphere Commerce Server, modify the transaction isolation level of the enterprise beans that are contained in the JAR file. Use the `modifyIsolationLevel` command line utility provided with WebSphere Commerce to set the transaction isolation level to the appropriate level for your database type. Refer to the "Modifying transaction isolation level of entity beans" section on page 349 of the *Programmer's Guide* for step-by-step details.
4. On the target WebSphere Commerce Server, load the access control policies for any new resources that you have created. Use the `acpload` and `acpnlsload` commands to load the policy information. For an example of how to load access control policy information, refer to the "Loading the access control policies for the new resources" section of the *Programmer's Guide* for step-by-step details.
5. On the target WebSphere Commerce Server, export the current WebSphere Commerce enterprise application from the WebSphere Application Server. Once the export has completed, an `.ear` file is created that contains the entire application. Refer to the "Exporting the current WebSphere Commerce enterprise application"section on page 350 of the *Programmer's Guide* for step-by-step details.
6. On the target WebSphere Commerce Server, export the configuration information for the enterprise beans that are contained in the current WebSphere Commerce enterprise application that is running within WebSphere Application Server. The information is exported to an XML file by using the `-export` option of the XMLConfig command line utility that is provided by WebSphere Application Server. You must add one new stanza to the generated XML file for each new entity bean that you have created. Refer to the "Exporting configuration information for enterprise beans"section on page 351 of the *Programmer's Guide* for step-by-step details.
7. Assemble your new enterprise bean or beans into the enterprise application using the Application Assembly Tool provided by WebSphere Application Server. Using this tool, you can open the `WC_Enterprise_App_`*`instanceName`*`.ear` file for the current application and then import any new enterprise beans. You can also specify configuration information such as class path information and security information using this tool. The end result of using this tool is that a new `.ear` file for the modified enterprise application is created. Refer to the "Assembling new enterprise beans into an enterprise application "section on page 357 of the *Programmer's Guide* for step-by-step details.
8. Stop and remove the current enterprise application from WebSphere Application Server. Refer to the "Stopping and removing an enterprise application"section on page 366 of the *Programmer's Guide* for step-by-step details.
9. Import your modified enterprise application (created in step 7) into WebSphere Application Server. Refer to the "Importing an enterprise application"section on page 366 of the *Programmer's Guide* for step-by-step details.

   While attempting to run the XMLConfig -import command, you may receive the following err message: "Cannot expand the ear file under Windows *`WAS_installdir`*`\installedApps\WC_Enterprise_App_`***`instanceName`***`.ear` AIX Solaris *`WAS_installdir`*`/installedApps/WC_Enterprise_App_`***`instanceName`***`.ear` 400 `/QIBM/UserData/WebAsAdv4/`*`wasInstance`*`installedApps/WC_Enterprise_App_`*`instanceName`*`.ear`If you receive this message, remov rename the preceding directory and run the command again.

10. Restart your WebSphere Commerce enterprise application. Refer to the "Starting an enterprise application"section on page 368 of the *Programmer's Guide* for step-by-step details.

Deploying extended commands and data beans

The method for extending existing commands depends upon the type of modification required. The methods for extension are explained in the "Customization of existing commands" section on page 139 of the *Programmer's Guide*.

In general, modification of existing WebSphere Commerce commands involves creating new classes that inherit from the classes that require customization. Override methods of the superclasses as required, to replace or modify logic. When customizing data beans, you also create a new class that extends an existing data bean. Within the new class, make the required modifications.

When creating these new classes, ensure that they are stored within one of your own packages, which is itself stored within the stores web project, `wcstores`.

Since extensions are effectively handled by subclassing, deployment for extensions to commands and data beans is the same as deployment for new commands and data beans. For more information about this type of deployment scenario, refer to "Deployment of new commands and data beans".

Note that the major difference when deploying new commands and data beans from WebSphere Studio Application Developer instead of from VisualAge for Java is the fact that a `.class` file is now used instead of a JAR file. As such, the steps that were previously described about repackaging JAR files are no longer applicable.

Deploying modified WebSphere Commerce public entity beans

When you modify a WebSphere Commerce public enterprise bean you modify WebSphere Commerce code. The deployment technique for customized entity beans is, therefore, slightly different than that used for new entity beans.

In general, the deployment steps for modified WebSphere Commerce public enterprise beans are very similar to those described in the *Programmer's Guide*, except for the following:

1. Instead of creating the EJB 1.1 JAR file, you now create a JAR file that includes the EJB deployed code. As such, you must follow the steps contained in this document to create the EJB JAR file, instead of the ones in the *Programmer's Guide* to create the EJB 1.1 JAR file.
2. Since the new EJB JAR file contains deployed code, you no longer have to perform the "Generating deployed code" section as described in the *Programmer's Guide*.
3. Repackaging the JAR file, as described in the *Programmer's Guide* is no longer required.

The following provides an overview of the deployment steps:

1. On the development machine, use WebSphere Studio Application Developer to create a new EJB JAR file for your modified EJB project. To create this JAR file, do the following:
   a. In the J2EE perspective, highlight the EJB project that contains your modified entity beans. For example, if you had modified the User entity bean, highlight the **WCSUser** EJB project
   b. From the **File** menu, select **Export**.
   c. Select **EJB JAR File** as the export destination and click **Next**.
   d. From the **What resources do you want to export?** drop-down list, select the EJB project that you want to export.
   e. Click **Browse** to select the directory into which the JAR file should be placed. In the **File name** field, enter a name (for example, `Cust_WCSUser-ejb.jar`) for the JAR file and click **Open**. All of

    the resources in the Java output folder `bin` are included in the JAR file.

    Note:

    > If this EJB project had been previously imported and the imported JAR file is specified on the class path, the contents of the specified JAR file will be merged back into the exported JAR. If a compiled class exists in the project, it will not be merged in from the imported JAR. As a result, if a Java class was deleted from the current project, during the export the class will be merged back into the new JAR file because it is contained in the `imported_classes` folder.

    f. Specify additional options as required, using the **Options** check boxes.
    g. Click **Finish**.

2. Copy the ***`Cust_EJBJarFile`***`.jar` JAR file into the following directory on the target WebSphere Commerce Server:

   ```
   drive:\WebSphere\CommerceServer\temp
   ```

3. On the target WebSphere Commerce Server, modify the transaction isolation level of the enterprise beans that are contained in the JAR file. Use the `modifyIsolationLevel` command line utility provided with WebSphere Commerce to set the transaction isolation level to the appropriate level for your database type. Refer to the "Modifying transaction isolation level of entity beans" section on page 349 of the *Programmer's Guide* for step-by-step details.
4. On the target WebSphere Commerce Server, export the current WebSphere Commerce enterprise application from the WebSphere Application Server. Once the export has completed, an `.ear` file is created that contains the entire application. Refer to the "Exporting the current WebSphere Commerce enterprise application"section on page 350 of the *Programmer's Guide* for step-by-step details.
5. On the target WebSphere Commerce Server, export the configuration information for the enterprise beans that are contained in the current WebSphere Commerce enterprise application that is running within WebSphere Application Server. The information is exported to an XML file by using the `-export` option of the XMLConfig command line utility that is provided by WebSphere Application Server. You must modify the stanza for each WebSphere Commerce public entity bean that you have modified to reflect the new JAR file that contains the modified code. Refer to the "Exporting configuration information for enterprise beans"section on page 351 of the *Programmer's Guide* for step-by-step details.
6. Assemble the modified WebSphere Commerce public entity bean or beans in the enterprise application using the Application Assembly Tool provided by WebSphere Application Server. Using this tool, you can open the `WC_Enterprise_App_`*`instanceName`*`.ear` file for the current application and perform a number of tasks. These include:
    a. Determining the class path information for the current version of the WebSphere Commerce public entity beans that you are replacing.
    b. Removing the current version of these entity beans.
    c. Importing the new version of these entity beans.
    d. Applying the previous class path information to the beans.
    e. Configuring WebSphere Application Server security for the methods contained in all of the enterprise beans in the modified EJB module.
    f. Saving the modification and creating a new `.ear` file for the modified enterprise application.

   Refer to the "Assembling modified enterprise beans into an enterprise application "section on page 361 of the *Programmer's Guide* for step-by-step details.
7. Stop and remove the current enterprise application from WebSphere Application Server. Refer to the "Stopping and removing an enterprise application"section on page 366 of the *Programmer's Guide* for step-by-step details.
8. Import your modified enterprise application (created in step 6) into WebSphere Application Server. Refer to the "Importing an enterprise application"section on page 366 of the *Programmer's Guide* for step-by-step details.

	While attempting to run the XMLConfig -import command, you may receive the following err expand the ear file under Windows *`WAS_installdir`*`\installedApps\WC_Enterprise_App_`***`ins`***

AIX Solaris `WAS_installdir/installedApps/WC_Enterprise_App_instanceName.ear`
`/QIBM/UserData/WebAsAdv4/wasInstanceName/installedApps/WC_Enterprise_App_ins`
If you receive this message, remove or rename the preceding directory and run the command a

9. Restart your WebSphere Commerce enterprise application. Refer to the "Starting an enterprise application"section on page 368 of the *Programmer's Guide* for step-by-step details.

Part 3. Appendixes

Appendix A. Hints and tips

Links to WebSphere Commerce tools

Once you start the WebSphereCommerceServer in WebSphere Studio Application Developer, you can use the following links to access WebSphere Commerce tools:

- For WebSphere Commerce Accelerator http://localhost/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=common.mcLogon
- For the Administration Console http://localhost/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogon
- For Store Services http://localhost/webapp/wcs/tools/servlet/ToolsLogon? XMLFile=devtools.Logon

The initial user ID and password for the site administrator is "wcsadmin" (for both values).

Changing workspaces

When WebSphere Studio Application Developer is started for the first time, you are prompted to specify the workspace directory that you would like to use. At that time, you can specify that you would not like to be prompted for this information anymore. If you make this selection, but then later want to switch workspace directories, you can do the following:

1. At a command prompt, navigate to the *WSAD_installdir* directory.
2. If you would like to specify the workspace diretory that should be used for this particular time that the application is opened, enter the following command:

   ```
   wsappdev -data workspace_dir -showlocation
   ```

3. If you would like to be prompted for the workspace location, enter the following command:

```
    wsappdev -setworkspace
```

If you are not prompted, you can start the application and use the correct workspace, by doing the following:

Browser usage

Using Microsoft Internet Explorer Version 5.5

If you are using Microsoft Internet Explorer Version 5.5 to access WebSphere Commerce code that is running within the context of WebSphere Studio Application Developer, the browser may redirect to an error page, rather than returning the page that was requested. In order to prevent this behavior, you should apply the "324929: December, 2002 Cumulative Patch for Internet Explorer 5.5 Service Pack 2".

To download the patch, visit the following Web site:

```
http://www.microsoft.com
```

Using Microsoft Internet Explorer Version 6.0

If you are using Microsoft Internet Explorer Version 6.0 to access WebSphere Commerce code that is running within the context of WebSphere Studio Application Developer, the browser may redirect to an error page when making SSL connections, rather than returning the page that was requested. In order to prevent this behavior, you should apply the "Internet Explorer 6 Service Pack 1".

To download this service pack, visit the following Web site:

```
http://www.microsoft.com
```

Incorrect PATH environment variable value

If a NoClassDefFoundError occurs in the "main" thread when you start the WebSphereCommerceServer, there may be an incorrect forward slash or back slash at the end of your PATH environment variable.

If this occurs, do the following:

1. Right-click **My Computer** and select **Properties**.
2. Click the **Advanced** tab.
3. Click **Environment Variables**.
4. From the **User variables for *userName*** list, select **PATH** and click **Edit**.Ensure that there are no extra forward or back slashes at the end of the value.
5. From the **System variables** list, select **PATH** and click **Edit**.Ensure that there are no extra forward or back slashes at the end of the value.
6. If you made changes to the values, click **OK** until you close the window.

Disabling caching to view JSP template changes

You must perform two configuration steps in order to be able to view changes to JSP templates without restarting the WebSphereCommerceServer test server.

First, you must disable the WebSphere Commerce caching, by doing the following:

1. Outside of WebSphere Studio Application Developer, navigate to the following directory:*enhancements_installdir*\instances*instanceName*\xml
2. Open the *instanceName*.xml file in a text editor.
3. Locate the following stanza and modify the `Enabled="true"` value to `Enabled="false"`, as shown:

```
<Cache AutoPageInvalidation="True"
         CacheCleanupAgentHostname="piggy"
         CacheCleanupAgentPort="8080"
         CacheCleanupPollingInterval="600"
         CacheConnectionTimeout="120000"
         CacheDaemonBindAddress="localhost"
         CacheDaemonMaxThreads="64"
         CacheDaemonPort="16999"
         CacheDirsPerMember="100"
         CacheFilePath="D:\COMMER~1\instances\WCDEV\xml\cache"
         CacheStoreClassName="com.ibm.commerce.cache.FileSystemCacheStore"
         Enabled="false"
         MaxAllowedRefreshPeriod="3600"
         MaxObjectsPerMember="0"
         WCSAppPath="/webapp/wcs/stores/servlet"
         name="Caching SubSystem"
         useCacheCleanupTriggers="false">
```

   Note that other values from the preceding snippet are specific to the particular environment.

The second step you must perform is to enable reloading of JSP code within the wcstores Web project, as follows:

1. Open WebSphere Studio Application Developer.
2. In the J2EE Navigator, expand the **wcstores** project.
3. Double-click **Web Deployment Descriptor**.
4. Select the **Extensions** tab and enable **Reloading enabled**.
5. Set an appropriate value for the reload interval and save your changes.
6. Restart your WebSphereCommerceServer test server.

Determining which version of the enhancements is installed

To determine which version of the enhancements is installed on your development machine, do the following:

1. From the **Start** menu, select **Run**.
2. In the **Open** field, enter `regedt32` and click **OK**.
3. In the HKEY_CURRENT_USER on Local Machine window, expand the following folders: **SOFTWARE** > **IBM** > **WebSphere Commerce Studio Enhancements**.
4. The pane on the right displays information about the version of the enhancements that is installed. You can determine the following information:
   - The value for `Edition` indicates if you have installed the enhancements for WebSphere Commerce Studio, Business Developer Edition, or for WebSphere Commerce Studio, Professional Developer Edition. The values are "Business" and "Professional" respectively.

- The value for `Modification` indicates if you have the FixPak 5.4.0.5 level of code, or the FixPak 5.4.0.5 *with* the Commerce Enhancement Pack (October 2002) level of code installed. The values are "Fix Pack" and "October Commerce Enhancement Pack" respectively.

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 LicensingIBM Corporation500 Columbus AvenueThornwood, NY 10594U.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 CorporationLicensing2-31 Roppongi 3-chome, Minato-kuTokyo 106, Japan

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

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

IBM Canada Ltd.Office of the Lab Director8200 Warden Avenue, Markham, Ontario L6G 1C7Canada

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

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

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

Trademarks and service marks

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

AIX
AS/400
DB2
DB2 Universal Database
@server
IBM
iSeries
VisualAge
WebSphere
400

Pentium is a trademark or registered trademark of Intel Corporation in the United States, other countries or both.

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

Solaris, Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc.

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