ExperienceReport

Migrating from IBM WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.4

Michelle A. Meyer

September 2002

Table of Contents

Migrating from IBM WebSphere Commerce Suite V5.1 to IBM WebSphere Commerce V5.4 on IBM eServer for iSeries

Migration

Supplemental material

ExperienceReport

Migrating from IBM WebSphere Commerce Suite V5.1 to IBM WebSphere Commerce V5.4 on IBM eServer for iSeries

IBM WebSphere(R) Commerce V5.4 is a set of integrated software components used to build, maintain, and host commerce web sites. WebSphere Commerce V5.4 is scalable, dependable, and has new capabilities for merchandisers and marketing personnel to sell products. Like IBM WebSphere Commerce Suite V5.1 (WCS V5.1), WebSphere Commerce V5.4 is built on Java(TM) technology and utilizes enterprise beans and Java server pages. Many enhancements have been made to WebSphere Commerce V5.4 in areas such as Commerce Models, Inventory Management, and Security Enhancements.

Customers who use WCS V5.1 on the iSeries can migrate directly to WebSphere Commerce V5.4. Customers who have WCS versions prior to V5.1 must first migrate to WCS V5.1 before migrating to WebSphere Commerce V5.4. In addition, there is no supported migration path from WCS V5.1 on non-iSeries platforms to WCS V5.1 on the iSeries.

The migration from WCS V5.1 to WebSphere Commerce V5.4 was performed by a group of iSeries 400(R) developers in the IBM(R) Rochester laboratory, called the Platform Evaluation Test team. Platform Evaluation Test serves as a means of providing a prerelease readiness assessment of the iSeries 400 system's ability to be installed and operated in customer-like environments. These test environments focus on the total solution, including the current release of the operating system, new and existing hardware, and customer-like scenarios. The scenarios, designed and constructed by the team, demonstrate how iSeries 400 technology can be used to build e-business solutions.

This migration was performed on the iSeries 400 Version 5 Release 1. It is strongly recommend that customers who are on Version 5 Release 1 and are considering migrating to Version 5 Release 2 should perform the WebSphere Commerce migration before migrating to Version 5 Release 2. This is because iSeries 400 Version 5 Release 2 does not support WebSphere Application Server, version 3.5. Version 5 Release 1 and Version 5 Release 2 support WebSphere Application Server version 4.0 and above. Part of the WebSphere Commerce migration process involves migrating WebSphere Application Server. Because of this, migrating to Version 5 Release 2 after migrating to WebSphere Commerce V5.4 is a more logical transition.

For this migration, we used WebSphere Commerce V5.4, Professional Edition. Although there are differences between this version and WebSphere Commerce V5.4, Business Edition, many of the topics included in this report can apply to both versions.

This report reflects our experience in migrating two WCS 5.1 instances to WebSphere Commerce V5.4 on the same iSeries system. The first instance is a Business-to-Business website with one store and the second instance is a mall-configuration with three Business-to-Comsumer stores.

Since this scenario does not exist in a production environment, the entire migration was performed on the same iSeries system. Migration on the same system is not recommended if a site requires 24/7 availability, since the migration process would cause more downtime than a migration on a development machine. However, most of this report is still applicable if WebSphere Commerce V5.4 needs to be installed on a different system than where WCS V5.1 is currently. Coexistence of WCS V5.1 and WebSphere Commerce V5.4 on the same system is not supported.

This report contains information we discovered while migrating our commerce sites. We used the IBM WebSphere Commerce for IBM eserver iSeries 400 Migration Guide, Version 5.4(hereafter referred to as the *Migration Guide*), along with the WebSphere Commerce V5.4 online help and assistance from the IBM Toronto development group. The first half of this report is structured in the same fashion as the *Migration Guide,* and correlates with the first four chapters in order to help customers easily troubleshoot and find their answers quickly while migrating. The first half is not meant to replace the *Migration Guide*, but provides additional information we encountered during our migration process. The second half of the report contains supplemental information regarding product advisor, discounts, campaigns, enterprise bean migration, and catalog manager that should be performed after site migration is complete.

WebSphere Commerce Suite V5.1 Premigration

Before migrating from WebSphere Commerce Suite V5.1 (WCS V5.1) to WebSphere Commerce V5.4, there are certain preparation steps that need to be taken. Taking the proper steps now can lead to a smoother and perhaps faster migration. This section contains information we discovered regarding database cleanup, ATP inventory, master catalog, and contract ids.

Database Cleanup

Depending on the data in the WCS V5.1 site, administrators may want to consider database cleanup before migrating to WebSphere Commerce V5.4. In doing so, the time it takes to migrate from WCS V5.1 to WebSphere Commerce V5.4 can be reduced. There are a number of different tables that can be cleaned up, and this can be done on an individual basis, depending on the needs of the site. If rows need to be deleted from the WCS V5.1 database, it is important to use the Clean WCS Database command in order to properly handle constraints.

The Clean WCS Database command works in conjunction with the CLEANCONF table, and the table should be referenced when using the command to ensure proper results. Each entry in the CLEANCONF table refers to a specific table in the WCS database and the criteria in which to clean up that table. More information regarding the CLEANCONF table is provided in a following section.

Clean WCS Database command

Clean WCS Database (CLNWCSDB) is the command that can be used to clean up any WCS V5.1 table. The command uses these parameters:

- DATABASE: the name of the relational database on which the WCS V5.1 data exists (commonly the iSeries(TM) system name)
- COLLNAME: the name of the schema (which is normally the same name as the instance name)
- PASSWD: the password to the schema
- DBTABLE: the name of the table to clean up. The name must be in single quotes and should be typed in the same case as the column TABNAME in the CLEANCONF table (typically lower case)
- TYPE: the type of object that needs to be deleted. Object types can be found in the CLEANCONF table under the column TYPE. The name must be in single quotes and should be typed in the same case as the column TYPE in the CLEANCONF table (typically lower case)
- STATUS: the order status of the record that needs to be deleted, in single quotes. The STATUS parameter is only valid if DBTABLE = orders
- LOGLEVEL: the level of logging that takes place during the deletion process. A 0 indicates no logging. Level 1 records delete statements for specified table. Level 2 records delete statements for specified table and any child tables as well
- NAME: the name of the organization to be deleted, in single quotes. A value for NAME is required if MEMBER was indicated for the DBTABLE keyword and ORGANIZATION was indicated for the TYPE keyword
- FORCE: the deletion force option. *YES indicates child tables to be cleaned first, which *NO allows constraints to cascade delete from child tables.
- CHECKONLY: the check only option. *YES will check child tables and delete restricts only. *NO will perform the cleanup
- DAYS: the minimum age (in days) of the database entry to be deleted. The DAYS parameter is only valid if the DAYSARG column in the CLEANCONF table is set to yes
- LOG: the default location for the log file is /QIBM/UserData/CommerceSuite5/instances/DBCLEAN_*instance_name*_YYYY.MM.DD_HH.MM.SS.ZZZ.LOG

where *instance_name* is the name of the WCS instance. If the log produced should have a different name and location, this can be specified here

For example, to clean up all obsolete addresses over six months (180 days) old, run the Clean WCS Database command with these parameters:

- CLNWCSDB DATABASE(relational database name) COLLNAME(schema name) PASSWD(schema password) DBTABLE('address') TYPE('obsolete') DAYS(180)

CLEANCONF table

As mentioned previously, the CLEANCONF table contains information necessary to run the Clean WCS Database command. Each row in the CLEANCONF table contains the following information:

- TABNAME: the name of the table where entries should be deleted
- TYPE: the identifier the type of entry to delete
- CONDITION: the condition under which all qualifying entries will be deleted
- NAMEARG: yes indicates a name parameter in CONDITION, and no indicates no such parameter
- DAYSARG: yes indicates a days parameter in CONDITION, and no indicates no such parameter

The CLEANCONF table comes with many default entries for database cleanup. However, site and store administrators can create their own entries in the CLEANCONF table to define new criteria for data to clean up. To define new types of data, enter rows in the CLEANCONF table that follow the previous guidelines.

Examples

These examples illustrate how a site administrator can create entries in the CLEANCONF table to specify data to delete. These two examples clean up pending orders and payment initiated order respectively.

Example 1, which will delete pending orders that have not been updated within a specified time period:

- TABNAME = orders
- TYPE = pending_order
- CONDITION = where status = 'P' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ?
- NAMEARG = no
- DAYSARG = yes

Example 2, which will delete all orders where payment has been initiated and authorization is pending:

- TABNAME = orders
- TYPE = payment_init
- CONDITION = where status = 'M'
- NAMEARG = no
- DAYSARG = no

In the first example, the column STATUS in the ORDERS table is used to specify the criteria to delete entries. Also, the second clause of the condition, (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ?, indicates that the age of the entry in days needs to be specified to delete the entry. Because of this, the DAYSARG is set to yes so a number of days can be specified in the Clean WCS Database command.

In the second example, again the column STATUS in the ORDERS table is used to specify the condition to delete entries. If compared to example 1, notice the days part of the condition is removed, and DAYSARG is set to no. This indicates that all entries in the orders tables where STATUS = 'M' will be removed, and the days parameter is not used in the Clean WCS Database command.

This is how to use the Clean WCS Database command with these two examples:

Example 1:

- CLNWCSDB DATABASE(relational database name) COLLNAME(schema name) PASSWD(schema password) DBTABLE('orders') TYPE('pending_order') DAYS(30)

Example 2:

- CLNWCSDB DATABASE(relational database name) COLLNAME(schema name) PASSWD(schema password) DBTABLE('orders') TYPE('payment_init')

Also refer to the *Migration Guide*, Chapter 1, under the section *Orders and order items* for additional database cleanup considerations.

Migrating to ATP inventory

A new optional capability of WebSphere Commerce V5.4 is an Available to Promise (ATP). This feature uses interfaces to check, allocate, and back order available or expected inventory items. The *Migration Guide* refers to premigration steps for those wanting to utilize ATP immediately after migrating. However, if you do not wish to migrate to ATP inventory right away, you can do so after migration. During our migration, we did not utilize ATP.

Master Catalog

In previous versions of WCS, catalogs could be created with less structure than WebSphere Commerce V5.4 requires. For example, with previous versions of WCS, a site did not need a master catalog. In addition, a store administrator could create items without having products associated with the items in previous versions of WCS. In WebSphere Commerce V5.4, each site must have a master catalog and every item must have an associated product.

During our migration, we had some concerns at this point regarding our catalog structure. Along with having stores that had items without associated products, one of our WCS instances had three stores, each having a separate catalog. We decided not to change anything regarding our catalog structure at this point. Later on in the migration process, we discovered WebSphere Commerce V5.4 not only creates a master catalog for stores that lack one, but that it creates products for items that do not have one associated with them.

It is important at this point, however, to ensure that the existing catalog structure is a proper tree. See the section entitled *Master Catalog* --> *Important structural considerations* in chapter 1 of the *Migration Guide.*

Validating contract ids

During the migration, we had some difficulties with stores having multiple contract ids. This caused problems later on in the migration when trying to purchase items. To see if a store has multiple contract ids, run this SQL statement:

- SELECT STORE_ID, TRADING_ID FROM *instance_name*.CATENCALCD WHERE TRADING_ID IS NOT NULL

  where *instance_name* is the name of the WCS instance.

If a STORE_ID has more than one TRADING_ID associated with it, problems may occur while migrating. It is recommend to change the column TRADING_ID before migration so that a particular STORE_ID only has one TRADING_ID associated with it.

Backing up Commerce Suite V5.1

As mentioned in the *Migration Guide*, it is extremely important to back up integrated file system directories, libraries, and HTTP servers. If a step should fail during the migration process, these backups may be used for recovery purposes. The *Migration Guide* mentions these three levels of back up:

- Backing up the WCS system: This ensures the entire migration process can be restarted if failure occurs.
- Backing up directories and files: As mentioned in the *Migration Guide*, back up the contents of the /QIBM/UserData/CommerceSuite5/instances/*instance_name* directory where *instance_name* is the name of the WCS instance, using a profile with *SECOFR authority,
- Backing up databases: Using a profile with *SECOFR authority, create a save file using the Create Save File (CRTSAVF) command. Then save the WCS instance library to that save file using the Save Library (SAVLIB) command.

To make saves more secure and useful:

- Place a copy of the directories and databases in a location other than the iSeries(TM) system where WCS V5.1 currently resides.
- When backing up directories and databases, it can be useful to put the date of when they were saved in the name of the directory or description field of the save file for future reference.

In addition to backing up directories and databases, it is strongly recommended to back up the current WebSphere Application Server, version 3.5 Administrative Configuration. To do so, follow these steps:

1. Ensure WebSphere Application Server, version 3.5 is running (STRSBS QEJBSBS/QEJB)
2. In a qsh window, enter the command

   /QIBM/ProdData/WebASAdv/bin myXMLConfig -export *backup_file_name.XML*
   -adminNodeName *admin_node_name*

   where *backup_file_name.XML* is the full pathname of the XML file that the command creates. The *admin_node_name* is name of the WebSphere node, which is generally the same as the system name.

Upgrading your software

The next step in the migration process is upgrading the WebSphere Commerce software. Before upgrading WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.4, WebSphere Application Server and WebSphere Payment Manager must be upgraded.

Before upgrading, ensure that the following products and corresponding fixes are installed on the iSeries(TM). system:

- IBM OS/400(R) V5R1 or higher, which includes DB2 Universal Database(TM) for iSeries
- IBM HTTP Server for iSeries (5722-DG1)
- Digital Certificate Manager (5722-SS1 Option 34)
- Qshell (5722-SS1 Option 30)
- Crypto Access Provider for iSeries (5722-AC3 (128 bit))
- IBM Developer Kit for Java(TM) (5722-JV1 Option 5)
- IBM Toolbox for Java(TM) (5722-JC1)
- TCP/IP Connectivity Utilities for OS/400 for the appropriate release (5722-TC1)
- DB2(R) Query Manager and SQL Development Kit (5722-ST1)

Upgrading to WebSphere Application Server 4.0.2

As mentioned in the *Migration Guide*, detailed instructions on migrating to WebSphere Application Server 4.0.2 are located at the following URL:

- www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/40Migration/40Migration3xToAE.html

When upgrading, it is strongly recommended to use the default port of 900. If a different port needs to be used, the following line must be edited in /QIBM/UserData/CommerceSuite5/instances/*instance_name*/XML/*instance_name*.XML, where *instance_name* is the name of the instance:

- <Websphere AdminServer="default" DatasourceName="WebSphere Commerce DB2 DataSource" JDBCDriverLocation="/Qibm/ProdData/Java400/ext/db2_classes.JAR" StoresWebApp="true" ToolsPort="8000" ToolsWebApp="true" WCMWebApp="true" name="WebSphere" port="900" />

It is recommended after upgrading to WebSphere Application Server 4.0.2 to back up the EJSADMIN4 library for recovery purposes.

Upgrading WebSphere Payment Manager 2.2 to WebSphere Payment Manager 3.1.2

The instructions in the *Migration Guide* regarding the migration from WebSphere Payment Manager 2.2 to 3.1.2 are simple and straight forward.

Upgrading WebSphere Commerce Suite V5.1 to WebSphere Commerce 5.4

The directions for upgrading WCS V5.1 to WebSphere Commerce V5.4 are simple and straightforward. After upgrading, visit the following link to find the latest fixes:

- www.ibm.com/software/webservers/commerce/support.html

It is recommended to have the latest fixes applied before migrating to WebSphere Commerce V5.4.

Migrating your instance

Migrating from WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.4 involves three main steps:

- Migrating your Commerce Suite V5.1 instance configuration
- Migrating your database
- Migrating store asset files (which is included under the heading *Post Data-migration*)

Migrating your Commerce Suite V5.1 instance configuration

Steps before migrating your instance configuration:

The instance's HTTP server configuration files need to be renamed. This is an important step, since the migration process will recreate a new *instance_name* HTTP and *instance_name***T** HTTP server, where *instance_name* is the name of the new WebSphere Commerce instance. This HTTP server will be created regardless if one already exists.

Migrating your instance configuration:

Detailed instructions for migrating an instance with the Migrate WCS Schema (MIGWCSSCH) command are located in the *Migration Guide*, in the section *Migrating your instance configuration.*

After the successful migration of the instance configuration (check /QIBM/UserData/WebCommerce/instances/*instance_name*_instMigrate.log to verify, where *instance_name* is the name of the WebSphere Commerce instance), these steps may need to be performed:

- Verify that an ear directory is created for the instance called /QIBM/UserData/WebASAdv4/*WebSphere_Application_Server_instance*/installedApps/WC_Enterprise_App_*instance_name*.ear, where *WebSphere_Application_Server_instance* is the name of the WebSphere Application Server instance and *instance_name* is the name of the WebSphere Commerce instance. When we performed the migration, we discovered after the database migration that this directory did not get created, and had to take steps to recover. It is easier to recover data at this point than after the database migration (which was when we discovered this error).
- Add these lines to the new HTTP configuration file to ensure Payment Manager will function properly:
    - Pass /PaymentManager* /QIBM/ProdData/HTTP/Protect/PymSvr/UI*
    - Pass /SampleCheckout* /QIBM/ProdData/HTTP/Protect/PymSvr/UI/SampleCheckout*
- Assign digital certificates: refer to the iSeries(TM) Information Center for instructions about enabling SSL and assigning digital certificates at:
    - http://publib.boulder.ibm.com/html/as400/infocenter.html
- If multiple WebSphere Commerce instances are being migrated on the same system, ensure that they are using different transport ports after the instance configuration. During our migration, the transport ports for both instances were automatically set to 22802. This caused problems when starting the WebSphere Application Server for the second instance.
    - To find out what transport WebSphere Application Server is using:
        1. Open the WebSphere Administrative Console
        2. Select the appropriate application server
        3. Click on the Services tab
        4. Click Web Container Service and click Edit Properties
        5. In the Web Container Service window, click on the Transport tab
        6. Under the port section is the transport port that will be used by the Application Server
    - To find which transport ports are already in use by the system:
        1. On an iSeries system run the Work with TCP/IP Network Status (NETSTAT) command
        2. Select option 3: Work with TCP/IP connection status
        3. Find a transport port that is not in use

Recovering from a failed instance configuration migration:

If the instance migration fails, take these steps to recover, as mentioned in the *Migration Guide*(in the following steps, *instance_name* refers to the name of the WebSphere Commerce instance):

1. Restore the instance's integrated file system directory
2. Remove any WebSphere Application Server components that the migration program has created. Alternatively, if the EJSADMIN4 folder was backed up ((as mentioned in the section Upgrading your Software) end the QEJBADV4 subsystem and restore the EJSADMIN4 library. Make sure it is restarted again before rerunning the Migrate WCS Configuration (MIGWCSCFG) command
3. Remove any HTTP Server configuration files, including *instance_name* and *instance_name***T**
4. Open /QIBM/UserData/WebCommerce/instances/wcs_instances. If there is a line referring to the particular instance similar to:

   instance_name;local=/Qibm/UserData/CommerceSuite5/instances/*instance_name*/XML/*instance_name*.XML

   remove this line

At this point the Migrate WCS Schema (MIGWCSSCH) command can be rerun.

Migrating your database

Running the database premigration analyzer:

The database premigration analyzer examines the database to determine if the data is ready for migration. The name "premigration analyzer" can be somewhat misleading, because the WebSphere Commerce instance needs to be migrated before the database premigration analyzer can be run.

Before running the premigration analyzer script (PREWCSMIG), first create a temporary empty library. This can be done with the command CRTLIB LIB(new library name). This library will be used as the SAVLIB parameter in the Premigration analyzer (PREWCSMIG) command.

After running PREWCSMIG, check the /QIBM/UserData/CommerceSuite5/instances/*instance_name*/logs/premigrate.log, where *instance_name* is the name of the WebSphere Commerce instance, to confirm no other database premigration actions need to be taken. If the premigration analyzer produces mandatory database premigration return codes, take the appropriate actions and run the premigration analyzer again until these return codes are not produced. Migration codes can be found in the *Migration Guide.* It is okay to proceed if optional database premigration return codes are produced, however, it is important to know the consequences that can occur. During the migration, we had optional database premigration return codes produced, and they had no ill effects to the site.

Migrating your database schema:

Before running the command Migrate WebSphere Commerce Schema (MIGWCSSCH), first create a temporary empty library. This can be done with the command CRTLIB LIB(new library name). This library will be used as the SAVLIB parameter in the MIGWCSSCH command.

If after running the migrate database schema command a fail.flag is produced, follow these recovery steps (in the following steps, *instance_name* refers to the name of the WebSphere Commerce instance):

1. Restore the instance's integrated file system directory
2. Remove any WebSphere Application Server components that the migration program has created. Alternatively, if the EJSADMIN4 folder was backed up (as mentioned in the section Upgrading your Software) end the QEJBADV4 subsystem and restore the EJSADMIN4 library. Make sure it is restarted again before rerunning the Migrate WCS Configuration (MIGWCSCFG) command
3. Remove any HTTP Server configuration files, including *instance_name* and *instance_name***T**
4. Drop the *instance_name* collection and restore the collection previously backed up in chapter 2 of the *Migration Guide*
5. Open /QIBM/UserData/WebCommerce/instances/wcs_instances. If there is a line referring to the particular instance similar to:

   instance_name;local=/Qibm/UserData/CommerceSuite5/instances/*instance_name*/XML/*instance_name*.XML

   remove this line
6. Go back to the beginning of chapter 4 of the *Migration Guide* and begin again with migrating the restored WCS V5.1 instance

configuration

Assigning a master catalog

A master catalog must be created if the script /QIBM/UserData/CommerceSuite5/instances/*instance_name*/temp/choosemc.sql exists at this point of the migration, where *instance_name* is the name of the WebSphere Commerce instance. This file may need to be edited prior to running, as described in the *Migration Guide*.

With both of our sites, this script was not generated. Therefore, no action was required.

Running the database consistency checker

Since WebSphere Commerce V5.4 requires a catalog with more structure than previous WCS catalogs, it is important to run the database consistency checker at this point to insure the migrated catalog fits the new structure. Like the database premigration analyzer, the database consistency checker produces mandatory and optional return codes, as indicated in the *Migration Guide*. If mandatory return codes are generated, take corrective actions and rerun the database consistency checker until these mandatory codes are not produced. If optional return codes are produced, corrective actions may need to be taken, but they are not mandatory.

With our sites we had optional return codes produced. We did not take any action towards them, and did not encounter later problems.

Post Data-migration

Migrating you security configuration

WebSphere Commerce V5.4 has stricter merchant key criteria than previous versions of WCS. If the merchant key used in WCS V5.1 does not meet the new criteria in WebSphere Commerce V5.4, the merchant key must be changed at this point of the migration process. Both of our sites had merchant keys that did not fit the new criteria, and consequently the merchant keys were changed. Refer to the *Migration Guide* for the new merchant key criteria and run instructions.

Migrating store file assets

In the *Migration Guide*, in the box before the section *Updating the Commerce Suite 5.1 JavaServer Pages*, one of the bullets mentions that you can launch a store with the Web address used in WCS V5.1, prior to migration. It is important to note that the migration being referred to here is the WebSphere Commerce V5.4 migration as a whole, not the store file asset migration (which is part of the WebSphere Commerce V5.4 migration). During the migration, we tried to use the web address at this point to access the site, but it failed because the store assets had not been migrated yet. After migrating the store file assets, the web address from WCS V5.1 should then work to access the site.

Changes to register.jsp and account.jsp

After running the migrateJSP command, the *Migration Guide* refers to Appendix F to update register.jsp and account.jsp. The *Migration Guide* indicates that in order to have stores return error codes when a customer fails to logon, the bolded lines in Appendix F must be added to these Java(TM) Server Pages (JSPs). While doing so, it is important to note that the bolded **flush="true"** lines have already been added to these JSPs via the migrateJSP command, and should not be added a second time.

Migrating your Payment Manager instance from 2.2 to 3.1.2

The directions in the *Migration Guide* referring to migrating the payment manager instance are straightforward.

Image path names in JSPs

After migration, as we started to view the sites, many of the images were not being displayed. We discovered that although images displayed in WCS V5.1, we did not have the JSPs pointing to the correct image files. One of the sites has been migrated many times, and when we performed the migration from WCS V4.1 to WCS V5.1, we copied the image files into the /QIBM/UserData/CommerceSuite5/instances/*instance_name*/stores/web/*store_name*/images folder, where *instance_name* is the name of the WebSphere Commerce instance and *store_name* is the name of the particular store. However, some of the JSPs referred to images with the old pathnames, and since we did not delete those images, they were displayed.

If images are not being displayed properly, check these items:

- Make sure the images are in the directory /QIBM/UserData/WebASAdv4/default/installedApps/WC_Enterprise_App_*instance_name*.ear/wcstores.war/*store_dir*/images, where *instance_name* is the name of the WebSphere Commerce instance and *store_dir* is the name of the particular store.
- The pathname that the JSPs need to return to access an image is *store_dir*/images/*the_image*, where *store_dir* is the name of the particular store and *the_image* is the name and extension of the desired images (for example, soap.jpg). This path name is calculated from both information in the JSP and information in tables such as CATENTDESC and CATGRPDESC. For example, both of these scenarios provide the correct path name:
    - The JSP refers to an image pathname as:

      <img src="<%=item.getDescription().getFullImage()%>"

      and the column FULLIMAGE or THUMBNAIL in either CATENTDESC or CATGRPDESC contains

      store_dir/images/*the_image*
    - The JSP refers to an image pathname as:

      <img src="**<%=storeDir%>**/<%=item.getDescription().getFullImage()%>"

      and the column FULLIMAGE or THUMBNAIL in either CATENTDESC or CATGRPDESC contains

      images/*the_image*
    - The JSP refers to an image pathname as:

      <img src="**<%=storeDir%>/images/**<%=item.getDescription().getFullImage()%>"

      and the column FULLIMAGE or THUMBNAIL in either CATENTDESC or CATGRPDESC contains

      the_image

  Note the following combinations will not work:

    - The JSP refers to an image pathname as:

      <img src="<%=item.getDescription().getFullImage()%>"

      and the column FULLIMAGE or THUMBNAIL in either CATENTDESC or CATGRPDESC contains

      images/*the_image*

      This will not work because the pathname produced is images/*the_image*, and the store directory is not included.
    - The JSP refers to an image pathname as:

      <img src="**<%=storeDir%>**/<%=item.getDescription().getFullImage()%>"

      and the column FULLIMAGE or THUMBNAIL in either CATENTDESC or CATGRPDESC contains

      store_dir**/**images/*the_image*

      This will not work because the pathname produced is ***store_dir*****/*****store_dir***/images/*the_image*, with the store name added twice.

Published and unpublished items

Another problem encountered after migrating was that unpublished items were not being displayed. We later found out we needed to make changes to ItemDisplay.jsp in order to display unpublished items. In WCS V5.1, the .getItems() method on the CategoryDataBean returned all items, published and unpublished. In WebSphere Commerce V5.4, the getItems() method only returns published items.

To access published and unpublished items in WebSphere Commerce V5.4, use the .getCatalogEntries(Long catalog_id) method on the CatalogEntryAccessBean as the following example illustrates:

- CatalogEntryAccessBean allItems[] = parentCategory.getCatalogEntries(new Long(parentCategory.getCatalogId()));

Product Advisor

Product Advisor enables shoppers to quickly browse items based on one or more criteria. If Product Advisor was set up on the WCS V5.1 site, only a few minor changes are required to enable Product Advisor to work in WebSphere Commerce V5.4 after the database schema and store assets are successfully migrated:

1. As mentioned in the *Migration Guide*, make sure the pe51.jsp, pc51.jsp, and sa51.jsp files are in the directory

   /QIBM/UserData/WebASAdv4/*WebSphere_Application_Server_instance_name*/installedApps/WC_Enterprise_App_*instance*.ear/wcstores.war,

   where *WebSphere_Application_Server_instance* is the name of the WebSphere Application Server instance, and *instance* is the name of the WebSphere Commerce instance. Only the Java(TM) server pages that were used in WCS V5.1 need to be there. For example, For example, if sa51.jsp was not used in WCS V5.1, it does not need to be used in WebSphere Commerce V5.4 if the same metaphor is being used.
2. In pe51.jsp, pc51.jsp, and sa51.jsp, references for com.ibm.commerce.datatypes need to be changed to com.ibm.commerce.pa.datatypes:

WebSphere Commerce Suite V5.1	**WebSphere Commerce V5.4**
com.ibm.commerce.datatype.DsString	com.ibm.commerce.**pa**.datatype.DsString
com.ibm.commerce.datatype.DsInteger	com.ibm.commerce.**pa**.datatype.DsInteger
com.ibm.commerce.datatype.DsDouble	com.ibm.commerce.**pa**.datatype.DsDouble
com.ibm.commerce.datatype.DsCurrency	com.ibm.commerce.**pa**.datatype.DsCurrency
com.ibm.commerce.datatype.DsDecimal	com.ibm.commerce.**pa**.datatype.DsDecimal
com.ibm.commerce.datatype.DsURLLink	com.ibm.commerce.**pa**.datatype.DsURLLink
com.ibm.commerce.datatype.DsImage	com.ibm.commerce.**pa**.datatype.DsImage
com.ibm.commerce.datatype.DsDate	com.ibm.commerce.**pa**.datatype.DsDate

3. In the WCS V5.1 searchspace.XML file, all references for com.ibm.commerce.datatypes need to be changed to com.ibm.commerce.pa.datatypes.
4. Run the Create WCS Product Advisor (CRTWCSPA) command with these parameters:
   - CONF = The name of the configuration file, as located in /QIBM/UserData/CommerceSuite5/instances/*instance_name*/XML/*instance_name*.XML, where *instance_name* is the name of the WebSphere Commerce instance. The name must be in single quotes
   - XML = The full pathname to searchspace.XML in single quotes
   - LOG = The full pathname of where the resulting log file should be created, in single quotes
   - HOST = The fully qualified instance host name
   - PORT = The WebSphere Application Server port number. The default is 900
5. After the CRTWCSPA command is successful, run the Create Metaphor(R) (IMPWCSPA) command with these parameters:
   - CONF = The name of the configuration file, as located in /QIBM/UserData/CommerceSuite5/instances/*instance_name*/XML/*instance_name*.XML, where *instance_name* is the name of the WebSphere Commerce instance. The name must be in single quotes
   - XML = The full pathname to metaphor.XML that was used in WCS V5.1, in single quotes
   - LOG = The full pathname of where the resulting log file should be created, in single quotes
   - HOST = The fully qualified instance host name
   - PORT = The WebSphere Application Server port number. The default is 900
6. At this point Product Advisor should be ready to use at the following URL:

   http://*instance_host_name*/webapp/wcs/stores/servlet/XX51.jsp?categoryId=*category_id*&storeId=*store_id*&langId=*language_id*
   - *instance_host_name* is hostname of the server where WebSphere Commerce is running.
   - *category_id* is the *category CID* used in the searchspace.XML file.
   - *store_id* is the identification number for the store. Open the STOREENT table to find the appropriate *store_id* for the store.
   - *language_id* is the language you wish to have displayed. For a list of available languages and their *language_id*s, refer to the LANGUAGE table.
   - XX51.jsp refers to the metaphor you wish to open.

Customer Profiles

WebSphere Commerce V5.4 provides customer profiles that allow store administrators to group customers together that meet certain criteria. These customers can have a wide range of traits that they commonly share. Examples of customer profiles that can be created include:

- Customers from a particular state
- Customers over 60 years of age
- Unregistered customers
- Customers from a particular state who are over 60 years of age

These customer groups can be used in conjunction with discounts and campaigns to target a specific audience.

To create a customer profile, follow these steps:

1. Log on to Commerce Accelerator as the store administrator.
2. Chose the desired store and click OK
3. Using the menu, click Marketing --> Customer Profiles
4. On the Customer Profiles page, click New
5. Enter information regarding the profile. Click on items from the menu on the left and customize the profile to fit the customers that should be targeted.
6. When finished customizing the profile, click OK. A window should appear indicating successful creation of the profile

If administrators create customer profiles that base their criteria on information that is not already included in the site's customer account pages, they would need to modify these pages so that customers can include this information when they register or update their profile. For example, in our sites we do not ask customers for their age. If we wanted to implement a customer profile that had an age requirement, we would need to alter the site's customer profile pages so that they can include their age when they registered or altered their profile.

Once the customer profile is created, it can be applied to discounts and campaigns. See the sections discounts and campaigns regarding these instructions.

Discounts

No additional steps need to be taken to enable general discounts created in WCS V5.1 to work in WebSphere Commerce V5.4. (General discount referring to those not bound to a user group). For discounts created in WCS V5.1 that target particular customer profiles, see the following section regarding discounts for specific customer profiles.

These next sections highlight some of the lesser known features of WebSphere Commerce V5.4 discounts:

Creating discounts: per product vs. per order

In WebSphere Commerce V5.4, there are two types of discounts that can be created. One type of discount is a per product discount, in which the discount will only apply to the product(s) it is assigned to. The other type is a per order discount, where the discount is applied to the entire order. Both types of discounts can have specific criteria such as minimum qualifications and amount of discount as a dollar amount or percentage.

Here are detailed steps for creating discounts. Creation of both per product and per order discounts are very similar. However, creating a per product discount requires a few additional steps at the end:

1. Log on to Commerce Accelerator as the store administrator
2. Chose the desired store and click OK
3. Using the menu, click Merchandise --> Discounts
4. On the discounts page, click New
5. Fill out general discount information (name, currency, and duration) and click Next
6. On the customer profiles page, select a customer profile (if desired) and click Next. If no customer profile is selected, the discount will be assigned to all customers
7. On the discount type page, select the type of discount to create (per product, per order, or multiple ranges discount), and click Next
8. Depending on the type of discount selected, Accelerator prompts for additional information regarding minimum qualifications and amount to discount. Click Finish when completed. If successful, Commerce Accelerator will indicate that the discount was created successfully

These are the extra steps needed to create a per product discount (which entails assigning the discount to one or more products):

1. Using the menu click Merchandise --> Products
2. Select a product to assign the per product discount to and click SKU
3. On the SKU page, choose one or more SKUs to add the per product discount to and click Discounts
4. On the Discount page select the discount and click OK. At this point Commerce Accelerator should indicate that the discount was successfully applied

At this point, the new discount should appear on qualifying orders. It is recommended that all browser windows to the site be closed and reopened before testing the discount.

Discounts for specific customer profiles

In WCS V5.1, tools did not exist to easily create a discount for a specific group of customers. If a store administrator wished to create such a discount, a customer group needed to be created by manually editing the database. However, in WebSphere Commerce V5.4, Commerce Accelerator provides tools to easily create customer profiles and apply those profiles to discounts, so that only customers who fit the corresponding profiles will receive the appropriate discount. In WCS V5.1, a customer profile was a separate entity than a customer group. It was possible to create customer profiles in WCS V5.1 using Commerce Suite Accelerator, but discounts could not be assigned to customer profiles in WCS V5.1. They could only be assigned to customer groups. In WebSphere Commerce V5.4, it is recommended to create customer profiles to replace the customer groups created in WCS V5.1.

During the migration process, it was discovered that customer groups that were manually created in WCS V5.1 no longer functioned properly. The discounts were still being applied to qualifying orders, but instead of being limited to only customers that fit the profile, the discount was being applied to all customers. We fixed this by creating customer profiles to replace the WCS V5.1 customer groups.

To create a customer profile, see the section entitled customer profiles.

Once a customer profile is created, a discount using the newly created profile can be created by following these steps:

1. Follow the directions under the previous heading creating discounts, steps 1 - 5
2. On step 6, select Assign this discount to specific customer profiles. A list of available customer profiles will appear. Select the profile and click the <<Add button to have the discount applied to the profile
3. Continue with step 7 and follow the rest of the discount creation steps

After the discount is successfully created, the new discount should appear on qualifying orders. Again, it is recommended that all browser windows to the site be closed and reopened before testing the discount.

Activating and Deactivating Discounts

In WebSphere Commerce V5.4, discounts can be easily turned off (deactivated) and turned back on (activated). This allows reuse of frequently used discounts. For example, if a company has a monthly sale on a particular product, a discount can be created for the sale. The discount only needs to be created once, and once the sale is over, the discount can be deactivated. When it comes time to implement the sale again, the discount can be activated again.

To activate or deactivate a discount, follow these steps:

1. Log on to Commerce Accelerator as the store administrator
2. Chose the desired store and click OK
3. Using the menu, click Merchandise -> Discounts

4. Click the desired discount and click either Activate or Deactivate

If a per product discount is deactivated and reactivated, it must be reassigned to the appropriate product(s). To do so, see the instructions under the previous heading creating discounts regarding assigning a discount to products.

When a discount is activated or deactivated, customers may not immediately see the changes on the site. This is because changes made to discounts will not be picked up until the CalculationRegistry has been refreshed. Since the refresh action has to be broadcasted to all clones, it has to be done through the scheduler. This process is done automatically by WebSphere Commerce, but depending on the other tasks the scheduler has, it can take a while for the refresh to complete.

Campaigns

If campaigns were successfully created in WCS 5.1 using Commerce Suite Accelerator, follow these steps to migrate the campaign to WebSphere Commerce V5.4:

1. In the Java(TM) server page (JSP) that displayed the campaign in WCS 5.1, add this import statement, as mentioned in the *Migration Guide*:

   ```
   <%@ page import="com.ibm.commerce.marketing.beans.EMarketingSpot" %>
   ```

2. Include code similar to the following in the JSP:

   ```
   <!-- Start of campaign code -->

   <tr>

   <td align="left" valign="middle" bgcolor="#FF3366" colspan=3>

   <h3>Featured Product</h3>

   <%
   //create the e-Marketing Spot
   EMarketingSpot eMarketingSpot = new EMarketingSpot();

   //IMPORTANT -set the correct name here
   eMarketingSpot.setName("ShoppingCartPage");

   //instantiate the bean
   DataBeanManager.activate(eMarketingSpot,request);

   EMarketingSpot.CatalogEntry []productResults = eMarketingSpot.getCatalogEntries();
   if (productResults != null && productResults.length > 0)
   {
   for(int i =0; i < productResults.length ;i++){
   EMarketingSpot.CatalogEntry catalogBean = productResults [i];
   CatalogEntryDescriptionAccessBean catalogDescriptionBean =
   catalogBean.getDescription();
   %>
   <TR>
   <TD>
   <A HREF="ProductDisplay?storeId=<%=catalogBean.getCommandContext().getStoreId()%>&langId=<%=
   catalogBean.getCommandContext().getLanguageId()
   %>&productId=<%=catalogBean.getCatalogEntryID()%>&catalogId=<%=catalogId%>">
   <img src="<%=storeDir%>/<%= catalogDescriptionBean.getFullImage() %>" alt="" width="35" height="50"
   hspace="8" border="0">
   </A>
   </TD>
   <TD>
   <A HREF="ProductDisplay?storeId=<%=catalogBean.getCommandContext().getStoreId()%>&langId=<%=
   catalogBean.getCommandContext().getLanguageId()
   %>&productId=<%=catalogBean.getCatalogEntryID()%>&catalogId=<%=catalogId%>">
   <%= catalogDescriptionBean.getShortDescription() %>
   </A>
   </TD>
   ```

```
</TR>
<%
} // end for loop
} // end if productResults != null
%>

</td>

<td width="5"><font class="text">&nbsp;</font></td>
<td width="5"><font class="text">&nbsp;</font></td>
</tr>
</TABLE>

<!-- End of campaign code -->
```

3. After implementing this code, the campaign should work.

Differences in campaign creation between WCS 5.1 and WebSphere Commerce V5.4

Creating a campaign in WebSphere Commerce V5.4 is similar to campaign creation in WCS 5.1. However, there is a difference between the two versions. In WCS 5.1, a store administrator needed to configure a rule service for each campaign based on the .adv file created by the campaign. In WebSphere Commerce V5.4, no rule service needs to be configured.

Creating a campaign

To create and display a campaign in WebSphere Commerce V5.4, follow these steps:

1. Log on to Commerce Suite Accelerator as the store administrator
2. Chose the desired store and click OK
3. Using the menu, click Marketing -> Campaigns
4. Click New to create a new campaign
5. Enter the name and description (the description is optional) of the campaign and click Next
6. Enter the sponsor and objectives for the campaign (both are optional) and click Finish. A box will appear indicating successful creation of the campaign.
7. A campaign initiative needs to be created for the campaign:
    1. Click on Initiatives
    2. Choose an existing initiative or click New to create a new one
    3. Fill in the general definition regarding the initiative. Under campaign, assign the initiative to the newly created campaign. Click Next
    4. Fill in the information regarding the content of the initiative. If you want to suggest specific products, use the Find and Browse buttons to view a list of available products. Click Next to continue
    5. On the defining target customers screen, select either to target all customers or specific customer profiles. If targeting customer profiles, choose the appropriate profile. Click Next to continue
    6. On the defining dates screen, select either to display the campaign everyday or only on

    specified days. Click Next to continue.

    7. On the defining target customer behaviors screen, choose a customer behavior. If no customer behaviors exist, create one by doing the following:
        1. Click Add to create a new behavior
        2. Choose a customer behavior from the list and complete the requested information for that behavior
        3. Click OK when finished
    8. After the customer behavior is created, click Finish to complete the initiative. A box will appear indicating successful creation

8. Next, an e-marketing spot needs to be created for the campaign
    1. Using the menu, click Marketing --> E-Marketing Spots
    2. Select the name of the page you wish to display the campaign on, and click Scheduler
    3. Click New to create a new Campaign Initiative Scheduling
    4. Fill in information regarding Campaign Initiative Scheduling:
        1. Choose the appropriate campaign from the drop down box
        2. Use the <<Add button to move the appropriate campaign initiative to the selected campaign initiatives box
        3. Enter a start date and time
        4. Chose whether the campaign should run indefinitely
        5. Click OK to finish

       If the campaign initiative schedule was created successfully, it will appear in the campaign initiative schedule list
9. Insert the code mentioned previously in the this section into the JSP that will display the campaign. Change this line to include the e-marketing spot where the campaign should be displayed (the same e-Marketing spot chosen in the previous step):

    ```
    eMarketingSpot.setName("ShoppingCartPage");
    ```

    For a list of valid eMarketingSpots, see Commerce Suite Accelerator

The campaign should be successfully created and ready to view.

Migrating custom enterprise beans

Before migrating to WebSphere Commerce V5.4, we implemented a search function for one of the sites. The search function consisted of a command which called an enterprise bean. The search function was based on the product search function code provided in the *WebSphere Commerce Suite V5.1 Customization and Transition Guide*, (http://w3.itso.ibm.com/itsoapps/Redbooks.nsf/9445fa5b416f6e32852569ae006bb65f/9a3be685f587d407852569fa00582819?OpenDocument), chapter 12. Using IBM(R) Visual Age for Java(TM) 3.5, we customized the command and enterprise bean to fit the needs of the site. We then implemented the search function using WebSphere Application Server.

When we migrated to WebSphere Commerce V5.4, there were many steps we needed to take to enable the custom enterprise bean to function properly. The following section describes the high level steps that were taken. It is important to note that because of many different variables present when working with commands and enterprise beans, these steps may not work with different combinations of commands and enterprise beans. We are simply describing the steps we took in our scenario. More information regarding enterprise beans in WebSphere Commerce V5.4 can be found in the *IBM WebSphere Commerce Programmers Guide, Version 5.4*, (http://www-3.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html).

It is also important to mention that before any changes were made, we imported the existing code into IBM's WebSphere Studio Application Developer (WSAD). WSAD combines functionality offered by IBM Visual Age for Java and WebSphere Studio. WSAD provides administrators functionality to create and test components such as commands and enterprise beans in a development environment. After we imported the code into WSAD, we had a Enterprise Application project (EAR project) called ProductSearchEAR and an Enterprise Bean Project (EJB Project) called ProductSearch. Additional information regarding WSAD can be found at http://www-3.ibm.com/software/ad/studioappdev/library/.

These are the high level steps we took to migrate the product search function:

1. We imported the class BaseJDBCHelper into the EAR project. We found that if we did not import this class, we would get a class not found error when starting up the WebSphere Application Server. The BaseJDBCHelper class can be found in the jar file wcsejbimpl.JAR.
2. WebSphere Commerce V5.4 has the ability to contain multiple instances on the same system. Consequently, the possibility lies to create an enterprise bean with the same name for each instance. Since multiple beans can be created with the same name, the name of the instance must be added to the beginning of the package name of the bean. This way the WebSphere Application Server knows which bean to access if there are multiple beans with the same name.

   For example, the instance we were using is called cruises. In WCS V5.1, the package for the bean was com/ibm/netfashion/objects/ProductSearch. In WebSphere Commerce V5.4, the bean needed to be in the package **cruises**com/ibm/netfashion/objects/ProductSearch.

   We changed the package location of the commands from com/ibm/netfashion/commands to **cruises**com/ibm/netfashion/commands and the package location of the results bean from com/ibm/netfashion/beans to **cruises**com/ibm/netfashion/beans. This way all packages originate from cruisescom/ibm/netfashion.

3. The Java Naming and Director Interface (JNDI) name of the enterprise bean needed to be com/ibm/netfashion/objects/ProductSearch, even though we changed the package name. This is because when WebSphere Commerce is looking for an enterprise bean's JNDI name, it automatically adds the instance name to the beginning of the JNDI name. Since the instance name was added to the package name, WebSphere Commerce will be able to find the enterprise bean. Here is an example to clarify:
   - The package name: cruisescom/ibm/netfashion/objects/ProductSearch
   - The JNDI name: com/ibm/netfashion/objects/ProductSearch
   - How WebSphere Commerce V5.4 interprets the JNDI name: cruisescom/ibm/netfashion/objects/ProductSearch (which is the same as the package name)

   Note: The JNDI name can be changed using WSAD's EJB Extension Editor.

   The methods defaultJNDIName() and setJNDIName() in the access bean also needed to return the JNDI name com/ibm/netfashion/objects/ProductSearch.
4. We needed to make sure that the information pertaining to the datasource was correct by using the same datasource that the cruises instance used.
   - Before making a connection to the database, we added the following line: debugInitConnection("WCInstanceDatasourceName", "WCInstanceName", "WCInstancePassword")
   - We verified the datasource information in WSAD's EJB Extension Editor matched the WebSphere Commerce instance's

datasource information

Note: Datasource information can also be changed in WSAD's EJB Extension Editor.

5. We exported the application to /QIBM/UserData/WebAsAdv4/*WebSphere_Application_Server_instance*/installableApps, where *WebSphere_Application_Server_instance* is the name of the WebSphere Application Server instance. (We used the default instance for all applications).
6. We imported the enterprise application into WebSphere Application Server using the WebSphere Application Administrative Console. During the installation process, most of the values were left as default. The following values were changed:
   - Since the WebSphere Application Administrative Console interprets the JNDI name as is, the JNDI name was changed from com/ibm/netfashion/objects/ProductSearch to **cruises**com/ibm/netfashion/objects/ProductSearch.
   - When asked to assign a server to the enterprise application, we selected the same application server that the cruises instance used.
   - After going through the installation process and clicking OK on the last step, a screen appeared asking if the code should be deployed. We clicked Yes and filled in the following:
     - For database type, we selected DB2UDBAS400_V4R5
     - For database name, we typed in the system name
     - For schema name, we typed cruises
     - All other values were left as default

   After completing these steps, the enterprise application was installed.
7. We verified in the WebSphere Application Administrative Console that the datasource assigned to the enterprise application's EJB Module was the same datasource that was specified in step 4, and that the user name and password information was correct.
8. We placed a copy of the ProductSearch.JAR (which was found in the expanded ProductSearchEAR.ear file) in the instance's wcsstores.war/WEB-INF/lib directory. There needed to be a copy in this directory because of the Java Server Pages (JSPs) that called product search commands and functions.
9. We needed to make changes to the JSPs that called the product search command and enterprise bean to reflect the package name change, since we changed the name of the package from com/ibm/netfashion to cruisescom/ibm/netfashion.
10. We changed entries in the cruises database tables to reflect the new package name. After the changes were made, we had the following entries in the cruises database tables:

TABLE	**COLUMN**	**VALUE**
CMDREG	STOREENT_ID	0
	INTERFACENAME	cruisescom.ibm.netfashion.commands.ProductSearchCmd
	DESCRIPTION	Command for Product Search
	CLASSNAME	cruisescom.ibm.netfashion.commands.ProductSearchCmdImpl
	PROPERTIES	NULL
	LASTUPDATE	NULL
	TARGET	Local
URLREG	URL	ProductSearch
	STOREENT_ID	0
	INTERFACENAME	cruisescom.ibm.netfashion.commands.ProductSearchCmd
	HTTPS	0
	DESCRIPTION	Command to carry out a search on the product catalog
	AUTHENTICATED	0
	INTERNAL	0
VIEWREG	VIEWNAME	ProductSearchViewTask
	DEVICEFMT_ID	-1
	STOREENT_ID	0
	INTERFACENAME	com.ibm.commerce.commands.ForwardViewCommand
	CLASSNAME	com.ibm.commerce.commands.HttpForwardViewCommandImpl
	PROPERTIES	docname=ProductSearch.jsp
	DESCRIPTION	View for Product Search
	HTTPS	0
	LASTUPDATE	NULL
	INTERNAL	NULL
ACCCMDGRP (entry 1)	ACCCMDGRP_ID	*unique_id* (see note below)
	MBRGRP_ID	-2
	ACCCMDTYPE_ID	Controller

	INTERFACENAME	cruisescom.ibm.netfashion.commands.ProductSearchCmd
	VIEWNAME	NULL
ACCCMDGRP (entry 2)	ACCCMDGRP_ID	*unique_id* (see note below)
	MBRGRP_ID	-2
	ACCCMDTYPE_ID	View
	INTERFACENAME	NULL
	VIEWNAME	ProductSearchViewTask

When we edited the ACCCMDGRP table, the entries listed were already present. We changed entry 1's INTERFACENAME from com.ibm.netfashion.commands.ProductSearchCmd to cruisescom.ibm.netfashion.commands.ProductSearchCmd. If an entry needed to be added to the ACCCMDGRP table, it is recommended to use the tools in the Administrative Console to do so because of the unique_id used for the column ACCCMDGRP_ID.

11. We verified that all servers and enterprise applications were started.
12. We went to the site and were able to use the ProductSearch command.

Configuration Manager

After a successful migration from WCS 5.1 to WebSphere Commerce V5.4, follow these steps to enable the use of the WebSphere Commerce configuration manager:

1. If configuration manager is going to be installed on the same PC as it was previously installed, back up the WCS400 directory. If configuration manager is going to be installed on a different PC, no action needs to be taken.
2. Follow the directions in the *IBM(R) WebSphere Commerce for IBM eServer iSeries(TM) 400(R) Installation Guide* under the section entitled Launch the Configuration Manager. The directions are identical for first time installation of configuration manager and installation over a previous version.

Helpful hints while installing configuration manager:

1. When running commands on Windows(R) by clicking Start --> Run, if either the java path name or the path name to RAWTGui.jar contains a space, use quotes around the path to signify that spaces should be included:

   "c:\Program Files\IBM\Java1.3\bin\java*.exe" -JAR c:\WCS400\RAWTGui.JAR
2. When running the Start WCS Config Command (STRWCSCFG), both the Client IP address and the server port number need single quotes around them, as mentioned in the *IBM WebSphere Commerce for IBM eServer iSeries 400 Installation Guide.*
3. If the following command fails:

   config_client.bat iSeries_Host_name Server_port_number

   try changing the case of the iSeries_Host_name. It is case sensitive.

Accessing WebSphere Commerce V5.4 Tools

WebSphere Commerce provides a variety of tools to manage sites and stores. These tools include the following:

- **Store Services:** Store Services allows administrators to manage their store archive files, and can be accessed at the following URL:

  https://*instance_host_name*:8000/storeservices

- **Commerce Accelerator:** Commerce Accelerator helps administrators manage customer and marketing information, and can be accessed at the following URL:

  https://*instance_host_name*:8000/accelerator

- **Administrative Console:** The Administrative Console allows administrators to manage groups and security, as well as WebSphere Payment Manager. It also enables administrators to view performance statistics. The Administrative Console can be found that the following URL:

  https://*instance_host_name*:8000/adminconsole

- **Web Editor:** The Web Editor allows administrators to manage their catalog, and can be accessed at the following URL:

  https://*instance_host_name*:8000/webeditor

- **WebSphere Commerce V5.4 Online Help:** the online helps provides general information on a wide range of topics. To access the online help, open the file

  \QIBM\ProdData\WebCommerce\web\doc\en_US\index.html

  on the system where WebSphere Commerce V5.4 is installed

On all of the previously mentioned URLs, *instance_host_name* is the hostname of the server where WebSphere Commerce is running.

Disclaimer

Information is provided "AS IS" without warranty of any kind. Mention or reference to non-IBM products is for informational purposes only and does not constitute an endorsement of such products by IBM.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.